

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

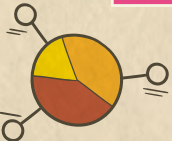
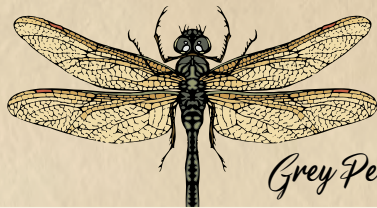
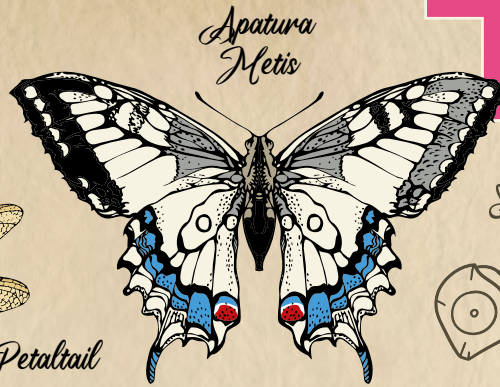
August 2020

Issue #33

OSCILLOSCOPE: FIND AND FIX FAULTS

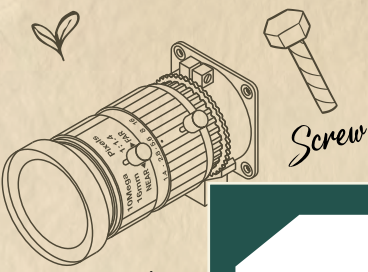
LEGO

3D-print extra parts to build useful stuff

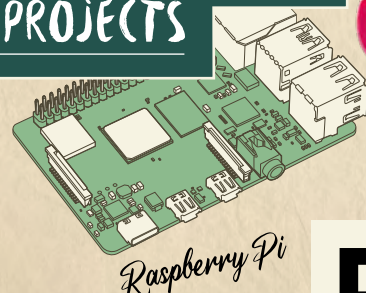
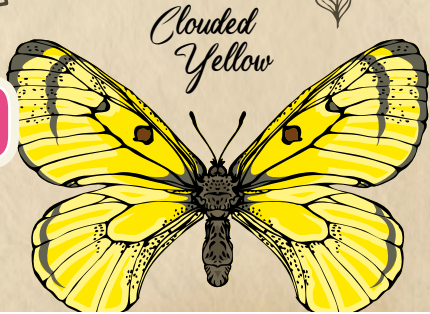


14 WILD HACKS

MONITOR, PHOTOGRAPH, AND RECORD NATURE WITH THESE SUMMER PROJECTS



Aug. 2020 Issue #33 £6



Alicia Gibb

Why Open-Source Hardware makes sense for makers

HACK A FURBY

Use this terrifying toy as an output device

KNITTING FLEX FILAMENT DIY WATCH KEYBOARD

Delivering Excellence in 48 hours to most areas in Europe

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



0800 587 0991
DIGIKEY.CO.UK



9.2 MILLION+ PRODUCTS ONLINE | 1,100+ 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. © 2020 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER
Supporting The Authorized Channel



Welcome to HackSpace magazine

Nature is amazing almost any way you look at it. The physical characteristics of, say, a bird are phenomenal. If you were to try and make something similar you'd need – actually, I've no idea what you'd need. It goes far beyond the strength-to-weight ratios of anything I've worked with. Maybe some fancy composites could do it. Yet nature makes this out of an egg and some insects with a little bird-seed sprinkled in.

Amazing! While I may not be able to compete with the making skills of Mother Nature, I can watch in awe at her creations. This month, we're looking at the point where tech and wildlife

meet. There are some great makers working with the natural world, whether that's watching nature for fun or doing serious ecological research. Let's get stuck in and work with our feathered, furred, and flying friends!

BEN EVERARD

Editor ben.everard@raspberrypi.org

While I may not be able to compete with the making skills of Mother Nature, I can watch in awe at her creations

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

get in touch at hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.org

[hackspacemag](#)

[hackspacemag](#)

ONLINE

[hsmag.cc](#)



PAGE **44**
SUBSCRIBE TODAY

EDITORIAL

Editor

Ben Everard

ben.everard@raspberrypi.org

Features Editor

Andrew Gregory

andrew.gregory@raspberrypi.org

Sub-Editors

David Higgs, Nicola King

DESIGN

Critical Media

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Sam Ribbits, Harriet Knight, Ty Logan

Photography

Brian O'Halloran

CONTRIBUTORS

Lucy Rogers, Rosie Hattersley, Jo Hinchliffe, Mayank Sharma, Andrew Lewis, Rob Miles, Andrew Robinson, Matt Bradshaw, Marc de Vinck, Rys Sommefeldt

PUBLISHING

Publishing Director

Russell Barnes

russell@raspberrypi.org

Advertising

Charlie Milligan

charlotte.milligan@raspberrypi.org

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](tel:01293312189)

hsmag.cc/subscribe

Subscription queries

hackspace@subscriptionhelpline.co.uk



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

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

Contents



68

06

SPARK

- 06 Top Projects**
Inspire yourself with shiny projects
- 18 Objet 3d'art**
Form and function made out of filament
- 20 Meet the Maker: 8 Bits and a Byte**
Internet-controlled surrealism
- 26 Columns**
A message from a time traveller
- 28 Letters**
Praise for OctoPrint, nozzles, and robots
- 30 Kickstarting**
A monthly box of maker projects

33

LENS

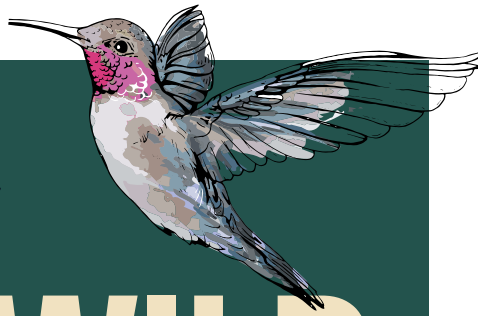
- 34 Nature**
Bring the natural world into your workshop
- 46 How I Rebuilt: A dual-head 3D printer**
A modified printer improved in every way
- 50 In the workshop**
Cut down on water use by washing like a plant
- 52 Interview: Alicia Gibb**
How open-source hardware benefits us all
- 60 Improviser's Toolbox Soap**
A very on-trend maker medium

Cover Feature

14 WILD HACKS

MONITOR, PHOTOGRAPH, AND RECORD NATURE WITH THESE SUMMER PROJECTS

34



Tutorial

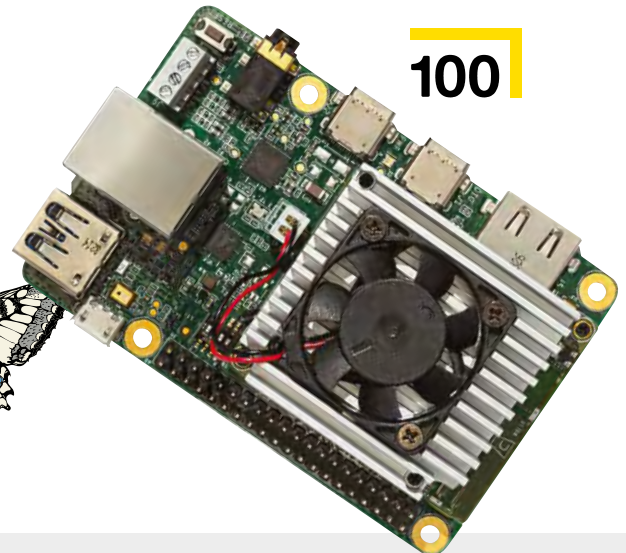
Hack a Furby



82

Turn a reanimated Furby into a personal assistant

100



106

Tutorial

Build with Lego



92

Integrate Lego into real-world builds

65

FORGE

66

SoM Mbed threading

Make the computer do two things at once

68

SoM Make a keyboard

Customise your primary input device

74

Tutorial Flex filament

Because bendy is trendy

76

Tutorial Knitting

Create a pot plant holder with this traditional craft

80

Tutorial Radio control

Add remote control to an Arduino robot

82

Tutorial Hack a Furby

Turn a toy into an internet display

86

Tutorial Oscilloscope

Advanced tips on debugging hardware

92

Tutorial Lego

Integrate real-world parts with building bricks

76

Interview

Alicia Gibb



52

We speak with the executive director of the Open Source Hardware Association

110



06



99

FIELD TEST

100

Best of Breed

The best development boards for AI

106

Review T-Watch 2020

An Arduino wearable smartwatch

108

Review Dial callipers

Accurate measurement the analogue way

110

Review Dremel 8220

Battery power without the downsides

112

Review Inkscape

Open-source vector editing for the masses

113

Book Review The Story of Tools

The best things since opposable thumbs

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.

Book Nook

By Jen Schachter

JenSchachter.com/booknookinfo

We've never embraced e-books or Kindles, probably because there's something so nice about seeing a row of book spines all turned out on display on a shelf. If you share this aesthetic appreciation, and would like to enhance your bookshelf with a portal to Narnia, here's a cool bit of decoration made by Jen Schachter.

It's all laser-cut plywood – the kits are all sold out at the moment, but you can buy the plans and wiring diagrams from Jen's Etsy shop. ▣

Right ▣
Narnia, on a bookshelf near you (Mr Tumnus just out of shot)





MAKING THINGS MOVE
DIY Mechanisms for Inventors,
Hobbyists, and Artists
Roberts
McGraw Hill

The World of Edward Gorey
Abrams

LESSONS OF DECEPTION
STERLING

Small add-on holder

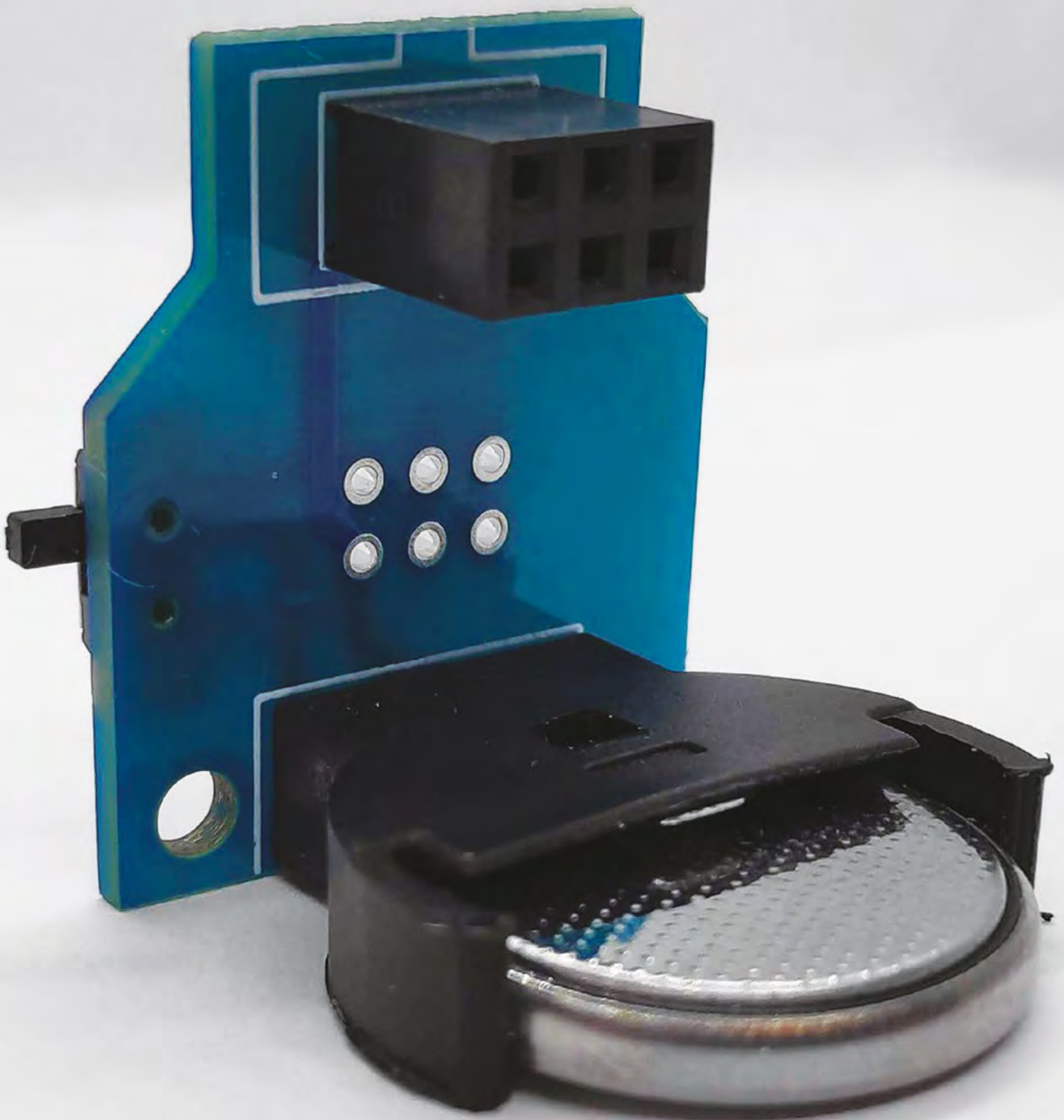
By Brian Lough

hsmag.cc/KRyLwG

As our columnist Drew Fustini has mentioned a few times, the field of PCB art is flourishing. Free software such as KiCad means anyone can create cheap, quirky electronics, and this has led to the art form known as badgelifelife. There's a standard for attaching these small badges to a larger PCB, and that's the SAO – as we're a family friendly magazine, we'll call it small add on – a standard created by Brian Benchoff.

This object is Brian Lough's stand for displaying and powering SAOs. It's a simple PCB incorporating a battery holder, power switch, and the necessary connector, and we reckon it has a kind of beauty all of its own. □

Right ♦
Display your badges at home when you're not at hacking conferences



Fibonacci LED display

By Evil Genius Labs LLC

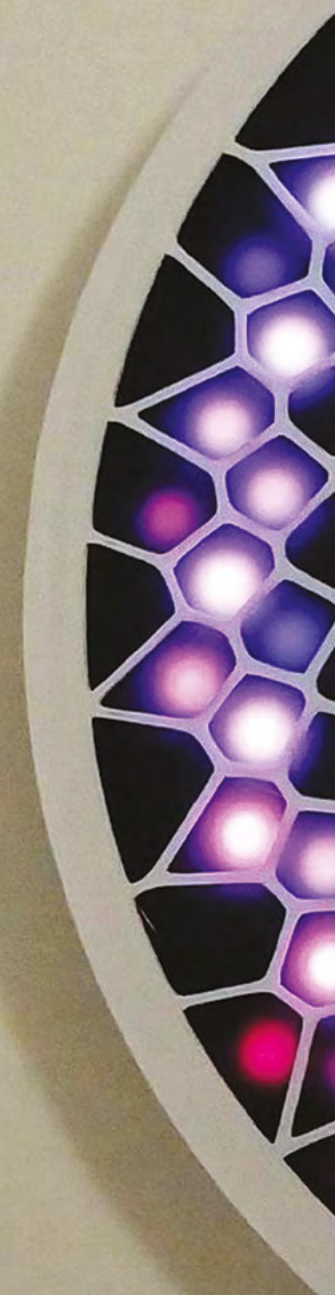
hsmag.cc/yH7Wjd

This piece of LED art by Evil Genius Labs comprises 256 surface-mounted RGB LEDs, controllable by an Arduino or ESP microcontroller to generate patterns based on the Fibonacci sequence.

If you're only aware of the Fibonacci sequence of numbers from Dan Brown's smash hit novel *The Da Vinci Code*, you might be forgiven for thinking that they're nothing more than a mathematical parlour trick. In fact, the sequence occurs in nature all the time, something which the maker says "makes the layout very organic and seemingly messy. But with the proper animation, spiral patterns emerge with spectacular results." ▣

Right ◆

The files for the white 3D-printed case are available on Thingiverse and PrusaPrinters





Sons of Patriots NERF gun

By Michael Darby

 314reactor.com

“ I’m a maker and I love to try and bring things from films and games to life using Arduinos and Raspberry Pis etc.

This particular project is trying to bring to life the ID-based control system from Metal Gear Solid 4 – where an AI system that rules the world has locked down all equipment with ID chips so that only certain people are able to use it.

In-game it is called the Sons of the Patriots System, as the AIs themselves are known as the Patriots. In-game equipment can be ‘laundered’ to bypass the ID system.

“It can also report back data on the users, so I’ve done my best to replicate that here with sensors that report the user’s ID via fingerprint, temperature/pressure, pulse, muzzle distance, and ammo count. As long as the Raspberry Pi is on a network and is accessible via SSH, this can be remotely administered to allow new users to unconditionally unlock the blaster, to re-lock it, and grab data from sensors.

“There’s also a GPS module on-board where, if I could get a satellite fix, it would report the blaster’s GPS location and movement etc. There is also an LCD screen that can give data to the user, such as when they are requested to log in and ammo count etc.

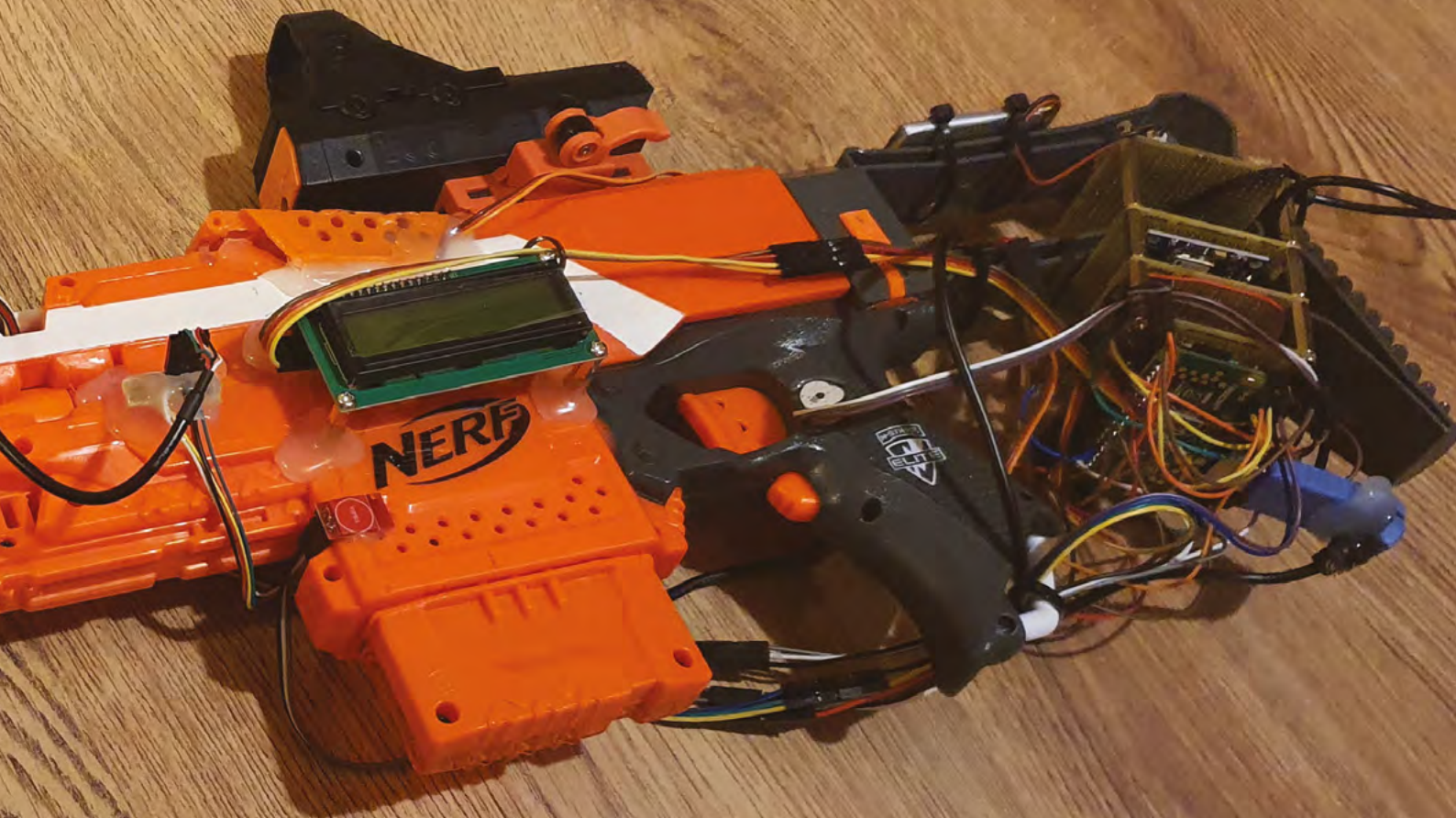
“Without power and an appropriate ID, the blaster won’t fire – if it is dropped it will lock again, as the user’s pulse is required in order to keep it running.

“It’s built with a Raspberry Pi, various sensors as noted above, and a servo that actuates the safety switch on the inside of the blaster – so when the user’s ID is confirmed, the servo moves and presses the button, allowing the blaster to fire.” □



Right □

Michael is looking forward to the day he can re-implement this build using nanotech rather than a Raspberry Pi



Shoji lamps

By Dheera Venkatraman

hsmag.cc/u29WJf

These beautiful lamps were printed on a Prusa i3 MK3S in PETG, not in once piece, or as standard parts made to be assembled with screws and glue, but as if they were pieces of wood cut for Japanese joinery. They assemble with no additional fixings, rather like a puzzle, so the printing on them has to be 100% accurate.

Although Dheera's examples here use RGB LEDs to achieve the final look, the files he's provided on GitHub don't specify what electronics to use – you could even, in theory, use a small candle. ▣



Right ◆

Japanese joinery evolved separately from Western joinery, to compensate for the lack of local iron for making nails

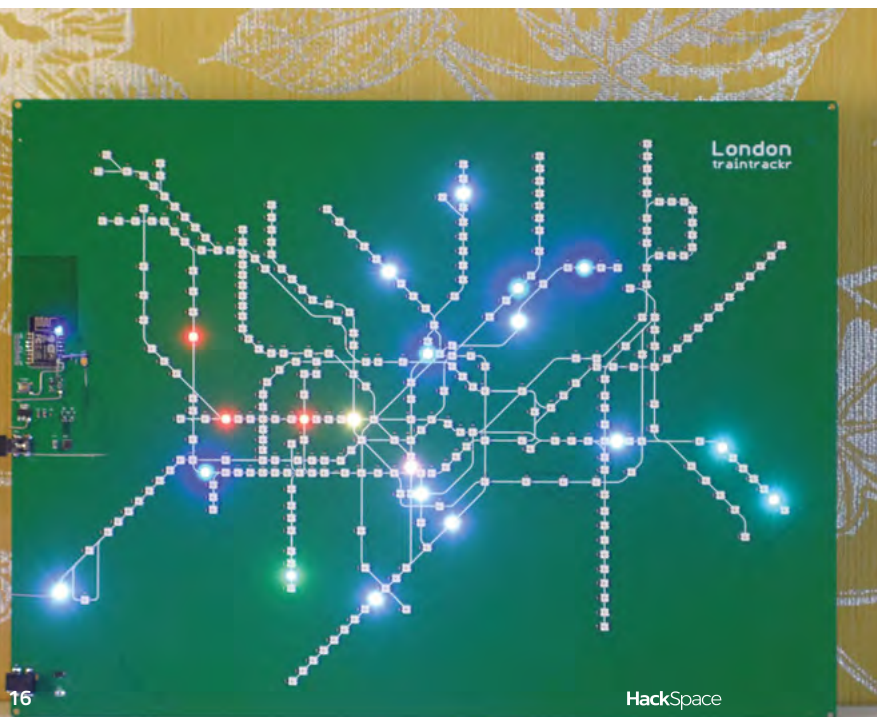


London Underground tracker

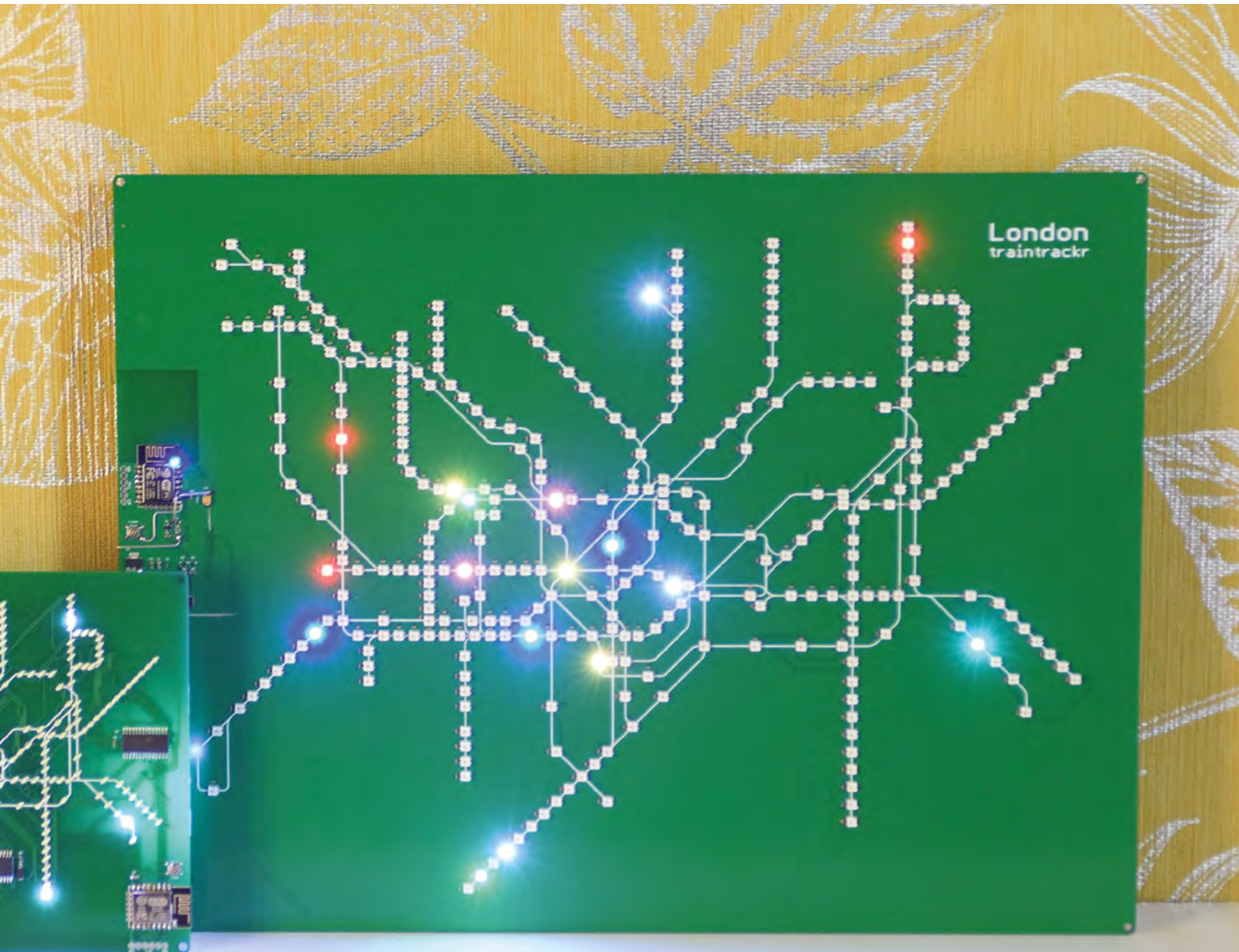
By TrainTrackr

 traintrackr.co.uk

To anyone who's got lost in England's bloated capital, this Tube map will be instantly familiar. To anyone not familiar with London's vast underground train network, this is a live map showing trains arriving at 333 tube stations on the Bakerloo, Central, Circle, District, DLR, Hammersmith & City, Jubilee, Metropolitan, Northern, Piccadilly, Victoria, and Waterloo & City lines. It updates once a second using live data from TrainTrackr.co.uk, and measures a substantial 40x30 cm. ▣



Below ♦
Harry Beck's design
for the London
Underground train
system lives on,
in surface-mount
LED form



Objet 3d'art

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

This ingenious ring pull helicopter was made by Stian Ervik Wahlvåg – we'll let the man himself describe it: "Many of my personal 3D-printing

projects are toy-related, and I am fascinated by articulated and mechanical parts. This project started when I was going to review a small and adorable 3D printer (Fulcrum Minibot) with a tiny build volume of 75×70×70mm, so I needed to make something that did not exceed the build volume."

"I am a self-taught 3D designer from Norway that started tinkering with 3D modelling 16 years ago. Currently, I work full time as a 3D designer, providing content for a shipping bridge simulator company that delivers solutions for maritime simulations. When I get home from work, I like to spend time with my girlfriend and two kids. Then, after the kids have gone to bed, I retreat to my home office to work on my personal 3D modelling and 3D-printing projects for my YouTube channel. □

➔ hsmag.cc/7NDnLQ





Meet The Maker: 8 Bits and a Byte



The duo making YouTube a little less serious

8

Bits and a Byte (also known as Dane and Nicole) have been making tremendously terrible tech (their own words) for almost three years now. They keep the ingredients

minimal, with usually just a Raspberry Pi, a sensor, and a bit of IoT logic, and film the result for their YouTube channel. What sets them apart from most other YouTube makers is that their builds take input over the internet, meaning you can interact with them from the comfort of your own home. Which makes them the perfect makers for our remote, housebound times.

//

We've never thought about an audience, but if we enjoy something, it's quite likely that someone else will as well

//

Dane: For my job, I was prototyping more standard stuff with Raspberry Pis. IoT was a buzzword, so everything had to be IoT. And a Raspberry Pi is an easy thing to start with. The company had its own cloud platform, and of course, they want everything IoT and connected. I don't know how it happened, but I ended up making these prototypes for customers. I got bored rather quickly, because it seemed like everyone wanted to do the same thing.

To cope with the pure frustration of having to do the same use cases over and over and over again, I decided to build something as ridiculous as possible as a kind of coping mechanism.

The corporate companion is a good example of that: everything he's programmed to say, I heard in a meeting.

Nicole: I started a course called Multimedia and Communication Technology. It's a long title, but it's really a bunch of design, a little bit of programming, a bit of 3D – a mix of things. It was during this course that I was introduced to the Arduino, so we started working with that and the laser cutters at school. We decided that we could do so much more with Arduino and Raspberry Pi than the boring work prototypes.

And then I had a video editing class, so I could borrow a camera at school. For our first few projects, I borrowed a camera, and everything was filmed in that moment. We thought the videos weren't that bad, so we decided to upload them.

We try to keep the builds not too techy. But apart from that, we just make for ourselves, and other people seem to like it. We have a really broad range of stuff as well.

Dane: I don't know if we come across as having thought things out, but we absolutely have not – we just do what we feel like. We've never thought →



Below ♦
 "It's a coping mechanism. If you work in corporate all day, you have to do something to reclaim your sanity"





about an audience, but if we enjoy something, it's quite likely that someone else will as well.

Nicole: We've got some really nice comments about the builds over the years on YouTube, for example. People have also started building stuff – I think we had The Goodbye Machine with the waving hand; someone was asking specific questions about certain things because they were building it too, and there's a second Coronavirus Slapper that someone built.

Dane: We're just chuffed because we haven't had a death threat yet. So I think our audience is pretty pleased with what we do. That's the bar we set ourselves, death threats.

Nicole: We took the Corporate Companion to a few events, Maker Faires and stuff. We added a big button so that people could press it – the reactions



were very mixed. Some people didn't see the humour in it. You could see with a few people that it hit a bit too close to the bone. We had a few managers in front of me – their wives were laughing at it, but they really weren't.

Dane: What we get a lot of is 'why didn't you?' questions. Why did you use a Raspberry Pi? And the standard answer is that it's the only thing we've got lying around. We have a few Raspberry Pis lying around that we reuse. If you go for a super-specific platform, it gets harder to reuse. We don't know what we're going to build, so it's nice to have options and not be tied down to anything.

We've always worked on a tight budget. I used a Raspberry Pi in a remote control cow, and I painted it green. A few people have noticed that it keeps popping up in other projects. And we've used the same speaker in different projects for the last three years, to keep the budget low.

Nicole: It's also because that's what we need for the project. We have some hardware lying around that we used once, and then we realised that it was easier to do it in a different way. If something works and it's easy, we just keep revising it. That's not thought out – that's just a desire to budget.

Dane: We had some other platforms that we use less that are very specific – some stuff that we won in a contest somewhere, but you need to program it in C. OK, so Raspberry Pi is overpowered maybe – it's running a whole operating system – but if I can get the job done in 20 minutes compared with programming in C for four hours, then that's exactly what I'm going to do. Laziness, plain and simple laziness.

WHAT NEXT?

Nicole: We're doing a fun project with some people, an international project, with Hannah Makes, TechnoChic, who's also on YouTube, and Clarissa of MakeAndFake YouTube fame. It's a challenge where we send each other stuff from Amazon and Wish and have to make something with it. It's already gone terribly wrong so far. Someone shipped to the wrong address, so we're all waiting for a part. That's going to be fun.

Dane: The second thing is, I have a habit of entering contests. Providing contest ideas that I know they →

Below ♦ It took two whole days to train the chatbot embedded in the talking toaster, and 95% of that was never used



don't want. There was a contest about AI that can read your mood and can enhance it. I thought it would be really funny if there were a little puppet that could read your mood, happy or sad, and it just stabilises your mood. So if you're happy, it gives you a upsetting fact, like 'everyone you love will die'; if you're sad, it will give you a happy fact. I expected that to be filtered out, but they liked it, and now I've actually built it. And the other thing is a bubble machine. That's the next thing that's in the pipeline.

Nicole: My favourite project that we've built is Ada. I spent a whole year on a talking artwork for school for my thesis. A lot of work went into that; a lot of research, a lot of writing. It used a Raspberry Pi and a touchscreen – the idea was that you could talk to Ada Lovelace.

Below ♦
"A friend of ours had that toaster – it was collecting dust, so we brought it to life"



We bashed apart an old picture frame and resized it to fit the touchscreen. It turns out that taking things apart is a lot easier than putting them back together.

Dane: I loved the Moomba because it was really wholesome, and some people left really nice comments under it. Very sweet and touching remarks. At Maker Faires, people loved seeing this little cow driving around. Kids saw it as a real cow and gave it kisses, petted it, which was lovely.

Then it's Ted the Talking Toaster. Because that was a great way to vent from the politicians and elections and everything. I could channel my dissatisfaction into a talking toaster. We now know that there was a talking toaster in the *Red Dwarf* programme, but at the time we didn't know what *Red Dwarf* was! It was from the *Fallout* series of games – there's an expansion pack where you help a talking toaster to take over the world. The toaster has been killed off now. His evil laugh and his comments got to us after a couple of weeks. □

Right ♦
The Zippedy Zap: an electric fly swatter that shocks the user – complete with added googly eyes





Dear Me

Making, life, and avoiding fizzy pop



Lucy Rogers

[@DrLucyRogers](#)

Lucy is a maker, an engineer, and a problem-solver. She is adept at bringing ideas to life. She is one of the cheerleaders for the maker industry. Her website is lucyrogers.com

I recently celebrated another successful orbit around the sun. It got me thinking about what I know now that I didn't know last year. And what I know now that I didn't know when I was 20. So, I am preparing a letter for the time travel device I am working on ...

Dear Lucy (aged 20),

I know when you went to university to study engineering, you had naively assumed that you would become proficient in workshop skills. Don't worry. Manufacturing and making runs deep through you – you will find a way to make the 'great-egg-race'-type fun and flippant contraptions you dream of. Having fun is not necessarily unprofessional. Being sarcastic is.

You have an insatiable appetite for learning – don't worry that you don't appear to focus on any one subject indefinitely. And ignore those who tell you that you have 'had enough education now'. You are different from many others – don't hide it – celebrate it. The courses you go on and the things you learn may not have an immediate benefit, but you have the ability to tie the threads together – so nothing is wasted. You have tenacity, adaptability, and resilience. These work in your favour. But learn how to fail safely.

Just because you are good at passing exams, does not mean that you shouldn't work with your hands – no matter what your elders say. This does not have to be restricted to DIY – but that's a great start and the skills you learn doing that – and the woodwork skills that Dad taught you aged eight – stay with you for life.

Communication skills are very important. Those technical reports that you write will not be read – you need to be able to explain it to non-technical people. Storytelling is not just for children. Humans understand technology a lot better if shown how it relates to them. Related: Don't drink fizzy pop before a presentation.

In the future, it will be simpler to find a community of people who enjoy making all sorts of things. It won't just be the woodturning club, full of retired men, or the Women's Institute. However, don't ignore either of these. It is possible to combine your practical skills with your academic skills – but you will have to make your own opportunities.

Sadly, the world is a lot more sexist than you currently realise. Try to find coping mechanisms for this. Related: You do not have to work for people who do not respect you. You do not owe them your loyalty.

You never find time. You have to MAKE time.

Even if you use your skills, they don't improve unless you consciously try to improve them.

Making for fun is not a waste of time.

Be nice. You may be requesting help from people you meet now in 20 years' time.

Education is not everything, but it does provide a passport, a short cut of having to 'prove' yourself.

Private life: that's a whole separate letter! Keep doing what you do. You rock!

Love

Lucy (from 2020). □

PCB design – the open way!

Getting started with KiCad



Drew Fustini

 @pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He is the Vice President of the Open Source Hardware Association, and a board member of the BeagleBoard.org Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

You've had an awesome idea for a new project, you've managed to get your breadboard prototype working, and you're ready to commit to making your design into a shiny new printed circuit board (PCB). To do that, you'll need to create a schematic and PCB layout using some kind of electronic design automation (EDA) software. Let's take a look at some of the options.

Altium is Windows-only proprietary software common in professional settings, where the company can afford hefty licensing fees. Autodesk EAGLE is also proprietary but runs on Mac, Linux, and Windows. EAGLE has a restricted free version that is popular with students and hobbyists. The commercial licensing is way less expensive than Altium, making EAGLE popular with smaller businesses, including many famous open-source hardware (OSHW) organisations like Adafruit, Arduino, and SparkFun.

In recent times, a free and open-source software suite called KiCad has been making waves in the PCB design world. KiCad has been around since 1992, when it was created by Jean-Pierre Charras. Until relatively recently, KiCad was a small fish in the EDA software pond, but in 2013 the iconic research organisation CERN started to invest in KiCad as part of their Open Hardware initiative.

This commitment from CERN improved KiCad dramatically, in terms of stability, functionality, and popularity. They worked on crucial features, including a push and shove router which is capable of routing differential pairs and interactively tuning trace lengths. These higher-end features allowed KiCad to handle more complex designs, including critical hardware controlling experiments at CERN, a complex 64-bit ARM single-board computer by Olimex, and the MNT Reform, a fully open-source laptop by Lukas Hartmann. Digi-Key is also investing heavily into KiCad, including developing a parts library and releasing a ten-part KiCad video series on YouTube with Shawn Hymel.

A great way to get started with KiCad is *Getting to Blinky 5.0*, a video tutorial by Chris Gammell (hsmag.cc/xBl7oG). There's also KiCon, a conference dedicated to KiCad where you can learn from other designers. The next KiCon will be held online in September 2020, (2020.kicad-kicon.com).

Like many open-source software projects, KiCad gets funding for developer time through donations. Hopefully, these donations will allow project leader Wayne Stambaugh, and other core developers, to dedicate more time to KiCad development. If you want to support professional-quality PCB design tools without cost, functionality, or intellectual property restrictions, you can donate at: hsmag.cc/VJU1jU. □

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



NOZZLES

What utter madness that something as simple as a new nozzle can make such a difference to your printing! After reading your review, I invested in the E3D Nozzle Fun Pack, and the difference is like night and day. I feel like an idiot for not getting new nozzles sooner, and that's when I know I've made the right decision – when I feel like an idiot for not doing something sooner.

Robert

Brisbane

Ben says: 3D printers are complex machines, and it's true that you can spend a long time adjusting parameters without knowing what's going to happen. It's also true that a slightly larger nozzle can give you much faster prints, and a smaller nozzle can give you much more detailed prints, without changing anything else in your setup.

TIN MACHINE

I take my hat off to the person who made the robotic guitar (Step Tranovich, How I Made, issue 32). But I want more... it takes MIDI input, so why not chain it to another input device – the 3D-printed Guitar Hero controller (issue 31)? It could be a guitar fest!

James

Wolverhampton

Ben says: Stop it. You're just being silly.





GREAT SCOTT!

Hmm, two *Back to the Future*-themed projects in one magazine? Are there fans in the HackSpace offices and, if so, when can we expect to see a functioning Mr. Fusion build?

Holly
Bristol

Ben says: Thanks for the idea – we’ve seen loads of fan builds of film props, plasters, droids, armour, all made using incredible techniques and dedication to details. Expect to see something movie-related soon! As for fans in the HackSpace mag office, my tastes run more to the brainless action of *The Expendables 2*, but I won’t look down on you if you want to waste your time on films that don’t star Wesley Snipes.

OCTOPRINTING

I saw in your interview that Gina Häußge, who made OctoPrint, felt guilty about not making any PPE during the pandemic. What? OctoPrint is what makes the world go round! I can’t imagine a world where I couldn’t remotely check up on what the status of my current print is. I probably wouldn’t have a printer at all if it weren’t for OctoPrint – I just wouldn’t be bothered. There’s no doubt in my mind that there are millions of face shields in the world that wouldn’t be in existence without her work. What an achievement!

Gary
Croydon

Ben says: I’m sure you’re right. It’s funny what we focus on though; for some people, no matter how much they do, they always know that they can do more, hence the talk of burnout in the same interview. It’s important that we support Gina – if you’ve used OctoPrint and enjoyed it, consider donating to keep the project going (octoprint.org/support-octoprint).



CROWDFUNDING NOW



Left
A DIY smartwatch for truly portable computing

CircuitMess STEM box

Get a discount maker kit every three months

From \$89 | [kickstarter.com](https://www.kickstarter.com) | Delivery: starting Nov 2020

CircuitMess, the team behind the Ringo phone that we looked at last issue, and the popular MAKERbuino games console, are running an ambitious Kickstarter to launch a quarterly maker kit subscription service. Every three months they'll release a new kit – the first six are already lined up:

1. **Spencer** – an internet-connected voice assistant
2. **Jay-D** – a DJing station
3. **Wheelson** – a self-driving car
4. **ByteBoi** – a retro games console
5. **Chatter** – encrypted wireless communications
6. **Clockstar** – a DIY smartwatch

These are intended as educational kits for people looking to learn more about technology, but that doesn't mean that they're just for young makers. By the looks of it, there'll be plenty of fun in these kits for anyone capable of safely wielding a soldering iron (one is included as part of the tools package with your first kit).

At the time of writing, the Earlybird offer of \$79 for the first kit is almost sold out, so if you've not signed up yet, the price will probably be \$89 (plus delivery). The kits will also be available as stand-alone products not part of a subscription, but the price will be noticeably higher. They've already hit their goal, so there's no doubt that the project will be funded (though with all crowdfunders, there's still a risk).

// **These are intended as educational kits for people looking to learn more about technology** //

While we've not had a chance to test out any of these kits yet, we have been impressed by previous CircuitMess products and are looking forward to playing with these. Previous CircuitMess kits have been sophisticated enough to really get your teeth into, but also good fun to play with straight out of the box if you don't want to write your own custom firmware. If these new kits can live up to our experiences with the older kits, then this service will be a great new way of learning about tech. □

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.



Above  Learn how to program and how to DJ with the Jay-D kit



Build a Makerspace for Young People

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

Sign up today: rpf.io/makerspace

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG
46

HOW I MADE: A DUAL-HEAD 3D PRINTER

Rebuilding a printer from the ground up for better speed, accuracy, and performance

PG
50

IN THE WORKSHOP

Hack a shower to save heat, water, and thus the lives of several polar bears

PG
52

INTERVIEW: ALICIA GIBB

Why open-source hardware is the sensible choice for makers of any size



PG
60

IMPROVISER'S TOOLBOX: SOAP

It's cheap, malleable, and smells good. Let's make something out of it!



PG
34

14 WILD HACKS

Record, monitor, and photograph
the natural world



14

WILD

HACKS

THE PAST FEW MONTHS HAVE GIVEN US AMPLE OPPORTUNITY TO STARE AT THE CREATURES THAT RESIDE OUTSIDE

Rosie Hattersley looks at ways to track them



Papilio Machaon



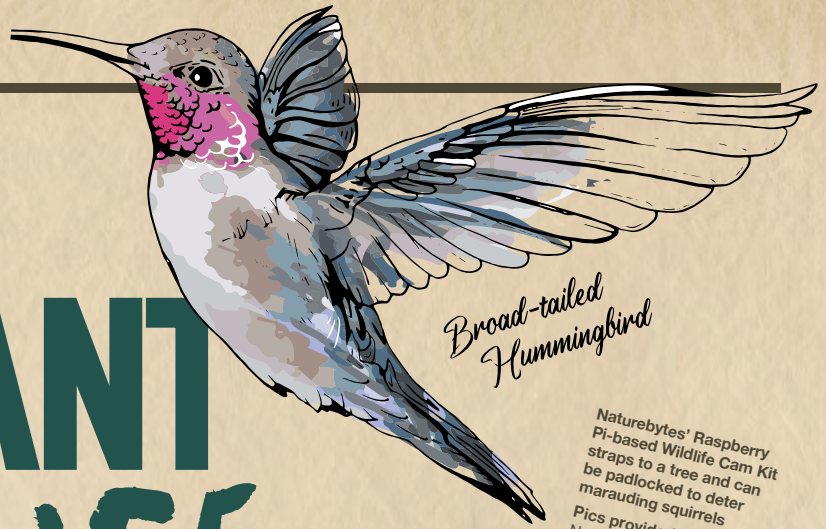
Great Tit

It's been a remarkable spring and early summer, and not just because many of us have had more time than usual to be able to appreciate our surroundings. The weather has been mild, the skies clear, and pollution levels low. As a result, it ought to be a bumper year for plants and wildlife. Unfortunately, the lockdown limited opportunities for embracing unexpectedly good weather while simultaneously making us more aware of the wildlife on our doorsteps.

If you're the outdoorsy type who likes to get out and stare intently at feathered friends from the comfort of a large shed on the edge of a lagoon, you may have spent the past few months getting to know suburban birds during your exercise walks, rather than ticking off unusual species. As things finally open up, it's a great time to take a fresh look at the world around us, and some of the projects focused on the creatures we share it with. →



Common Octopus



Broad-tailed Hummingbird

BRILLIANT BIRD-LIFE

Naturebytes' Raspberry Pi-based Wildlife Cam Kit straps to a tree and can be padlocked to deter marauding squirrels
Pics provided by Naturebytes



NEST BOX

The Nest Box (fb.me/NestBoxLive) is a great initiative that lets you watch live video streams of hatchlings in nests around the UK. At other times, you can view time-lapse video montages of what's been happening in the nests, and see how quickly the chicks grow and fledge. It's made from a Raspberry Pi and Camera Module, plus a custom PoE (Power over Ethernet) HAT to power the setup and provide internet connectivity. A custom HAT controls LED lights and the Raspberry Pi Camera Module's infrared filter so Nest Box Live viewers can see what's happening in the nest, even at night-time. A Python-based setup means live-streaming mode on Facebook is triggered by the motion sensor detecting activity in the nest. More details on the project can be found on the Raspberry Pi blog: hsmag.cc/CkyIN1.

Some of the nests are sited at the Wirral Barn Owl Trust, where kestrels, starlings, swifts, and owls are among the birds whose lives you can drop in on. You aren't limited to viewing online: there's an option to buy your own ready-to-stream bird-box (£250) and become part of the Nest Box Live network.

Creator Jamie Wainwright tells us he's recently shipped kits to Germany, the Netherlands, and South Africa, so more bird species will soon join the UK nest streams, but you could always make your own.

Nest Box kits enable owners to add their sites to the nestboxlive.com online community



Kestrel chicks in the nest at Wirral Barn Owl Trust live-streamed by Nest Box Live



CREATE YOUR OWN BIRD CAM

You can make a bird cam by training a webcam or a WiFi-enabled camera on a bird-box, or a popular plant in your garden. If you're making a self-contained bird video cam, it's a good idea to invest in a waterproof case, such as the

£40 Wildlife Cam Case made by Naturebytes (hsmag.cc/kuzl3T). It is IP65-certified waterproof, and is designed to work with a motion-detecting IR camera and a Raspberry Pi.

You can clip any model inside, strap it to a convenient tree or post, and padlock the case. A more convenient option is the ready-to-use £110 Wildlife Cam Kit (hsmag.cc/mbPmLz), which includes a Raspberry Pi +, Camera Module, and IR motion sensor.





Click on the penguins to help conservationists understand population numbers and how penguins behave

PINPOINT A PENGUIN

A near effortless way to help conservation efforts is to get involved in **Penguin Watch** (hsmag.cc/rRjiYn). This asks site visitors to help count the number of penguins in still photos extracted from Raspberry Pi Zero

cameras set to create a time-lapse. The project is organised by the British Antarctic Survey and World Wide Fund For Nature, alongside the Zoological Society of London, which originally set up a related Penguin Lifelines project back in 2014 as a means of monitoring Antarctic populations and behaviours.

Depending on the project's available data, given the Raspberry Pi camera's remote locations in Antarctica, Isla de los Estados near Ushuaia, Argentina, and Beagle Island near Cape Horn, live penguin viewing may be an option. However, site visitors are mostly asked to count young adults or chicks, by marking their locations on still images taken from the time-lapse photo feeds.

The Penguin Watch site contains full visual guides to help citizen scientists identify penguin species and sexes, as well as how to append your data to live projects.



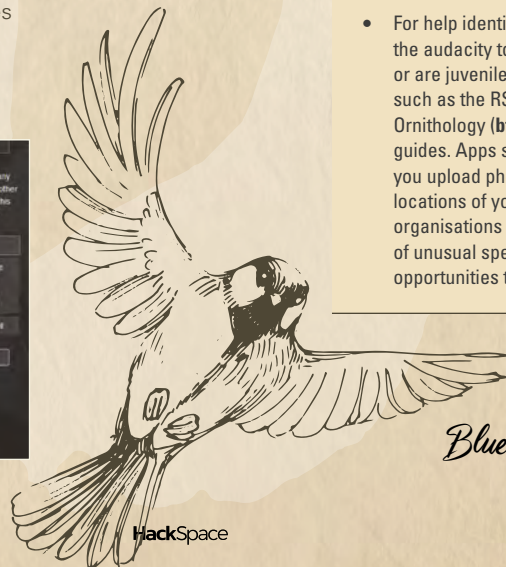
PUFFIN PUFFARAZZI

A nother really worthwhile citizen scientist-based project, for those lucky enough to spot one, is to send your photos of puffins to the RSPB. Their Puffarazzi project (hsmag.cc/f1CrDw) invites wildlife photographers to share puffin photos in order to help conservationists gain a better understanding of exactly what puffins feed their young.

Puffins can be found in remote spots around the UK including Mull, Shetland, the North Yorkshire coast, and Rathlin Island in Northern Ireland. →

Birdwatching basics

- If you've developed an interest in the birds in your garden, you've probably already set up a feeder to encourage avian visits. If you don't have a garden, a clear Perspex feeder, with suction cups to attach it to the window, makes an effective bird feeder.
- Try recording the sounds of visiting birds (or those you see in the countryside). The microphone and Voice Memo app on an iPhone are surprisingly effective. Recorded bird sounds are highly evocative memory triggers. You can also brush up on your bird identification skills via the RSPB Birdsong Radio app.
- For help identifying birds (especially if they have the audacity to be in different seasonal plumage or are juveniles with less distinct markings), sites such as the RSPB (rspb.org.uk) and British Trust for Ornithology (bto.org) provide online identification guides. Apps such as Audubon Bird Guide let you upload photos and video, as well as mapping locations of your sightings. Join any of these organisations to get involved with following news of unusual species passing through the UK, and opportunities to see them.



Blue Tit



Clouded Yellow

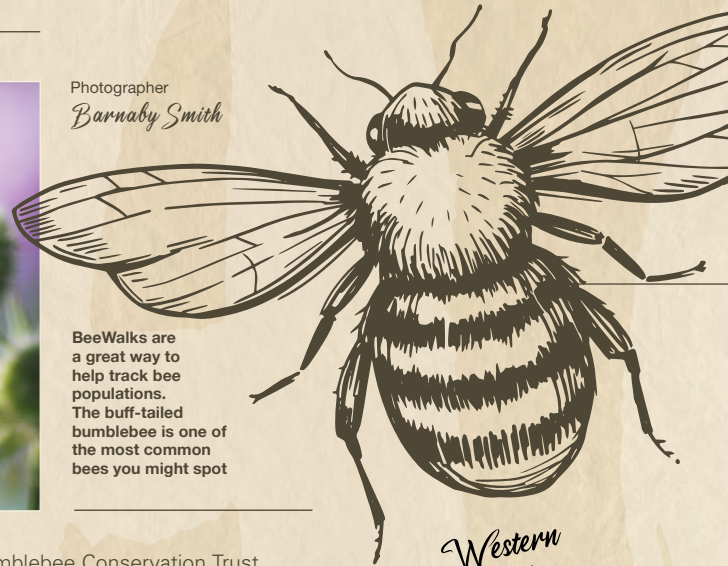
BUGS, BEES, & BUTTERFLIES

POLLINATOR WATCH

Credit Bumblebee Conservation Trust



Photographer
Barnaby Smith



BeeWalks are a great way to help track bee populations. The buff-tailed bumblebee is one of the most common bees you might spot

Western honeybee

If you watched BBC *Springwatch* this year, you may already know about the Open University's Pollinator Watch (hsmag.cc/6FkdWg). You may also know that the value of pollinators' aid to the production of crops in the UK is up to £660 million per year.

Pollinator Watch invites citizen scientists to report sightings of particular types of bee in order to build a picture of pollinator distribution across the UK. Having confirmed you want to take part, you fill out a multiple-choice form online and indicate the rough location. Uploading images helps confirm sightings – useful if you're not fully confident of your ID skills.

Given pollinators' important role, one of the most useful things to do is become a BeeWalker.

Organised by the Bumblebee Conservation Trust (hsmag.cc/bdntXL), a BeeWalk is a regular monthly, hour-long stroll from March to October, during which you, as a BeeWalker, record any bees you see, the number seen together, and the weather conditions. Choose a transect near enough to home so that it's convenient to walk regularly. Doing so will ensure that a picture can be built up of changes to the bee population over successive years.

You can also pick up 'lapsed' transects to continue an existing dataset. The BBCT says: "information collected by BeeWalk volunteers is integral to monitoring how bumblebee populations change through time, and will allow us to detect early warning signs of population declines".



Stag Beetle



Blue Morpho

BUILD A BEE MONITOR

Keeping bees is one of the best ways to help boost pollination levels in your area, and help the production of fruit and vegetables. It also means you can get up close to your bees and find out more about them. Because a beehive is a self-contained ecosystem, you need to ensure the hive's health by maintaining its ambient temperature and monitoring population levels. It's possible to set up a system yourself using an Arduino or Raspberry Pi to control things. You can then use a convenient app, such as OSBeehives (osbeehives.com), to monitor both the temperature and activity in the hive, or attach a mini temperature gauge.

Glyn Hudson has been running a multi-hive setup in Snowdonia in his parents' back garden for the past eight years. He created BeeMonitor (beemonitor.org) to check on their hives' health. He says: "Bees maintain a 34°C core brood temperature ($\pm 0.5^\circ\text{C}$), even when the ambient temperature drops below freezing. Maintaining this temperature when a brood is present is a key indicator of colony health." The

BeeMonitor consists of a low-power RF sensor connected to an Arduino, and a DS18B20 temperature sensor. Temperature readings and hive activity details are uploaded to the BeeMonitor site by a Raspberry Pi, where they can be viewed by anyone. Due to the remote location, Glyn used a PV solar panel and a car battery to power the setup.

For more details of the project, take a look at this link: hsmag.cc/nm3V9T. →



Pics provided by Glyn Hudson

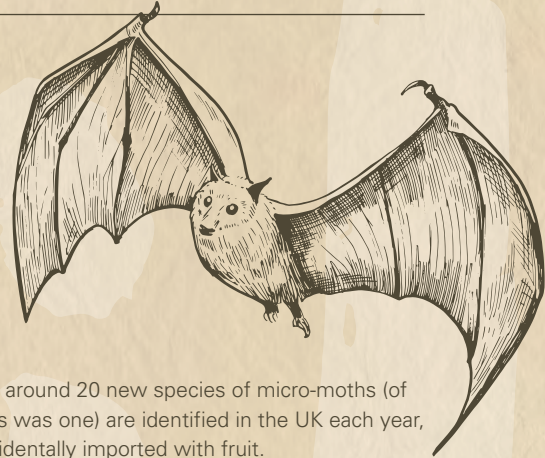
BeeMonitor streams details of several hives in Snowdonia using a DS18B20 temperature sensor connected to an Arduino and Raspberry Pi setup

BUTTERFLIES MOTHS & BATS



IDENTIFY BUTTERFLIES

Eptesicus serotinus



A suburban garden might not be the obvious place to spot something unusual in the shrubbery, but they often host a surprising variety of bugs and moths. The Butterfly Conservation charity estimates as many as 100 species visit Britain's gardens. Researching a tiny butterfly on a fuchsia bush a few weeks ago, it was fascinating to

learn that around 20 new species of micro-moths (of which this was one) are identified in the UK each year, often accidentally imported with fruit.

The 'traditional' way to see what butterflies are around is by using a net to catch them, take a long look at their markings, and then release them. Given their delicate wings, it's not ideal. Simply recording what you see with the aid of a guide-book or localised

crib sheet, plus a smartphone or camera, is a better option. You could also set up a nature cam (see page 42), or train a webcam on a plant that regularly attracts butterflies and other flying creatures. As with bats and birds, there are useful apps to help you identify species you encounter when out enjoying the great outdoors.

The best time to see butterflies is when it's sunny, particularly when it begins to brighten up just after a rain shower, as they are attracted by the emergent scent as the sun warms the flowers. Stand so you don't cast a shadow as you'll be mistaken for a predator.

Mint Moth



Photograph your wildlife findings to identify later



OPPOSITES ATTRACT

A simple way of discovering moths is to set up an egg-box moth trap. Based on the premise that moths are attracted to the light, you simply place a box with a light in cosy places to hide in your garden overnight. Since it's really the darkness adjoining the light that moths like, opened up egg-boxes, or soft acoustic padding, appeal. Get up at dawn, or just after, to see what has found its way into your box, taking photos of any you can't identify before they flutter away. An even simpler option is to peg a white sheet on the washing line for ten minutes as dusk falls and see what arrives.



Photographer
Ben Sale

Ben Sale photographed the *Monochroa palustrella* moth using a trap in his back garden in Stevenage

AUDIOMOTH MONITORING

AudioMoth, recommended by the British Bats Survey (bats.org.uk), is ideal for monitoring bats, moths, and birds – for example, one user, Adrian Bicker, identified migratory habits of rare *Nathusius' pipistrelle* bats (hsmag.cc/E68Nks). Having first identified several near where he lives in Dorset, he found many more on the Isle of Wight.

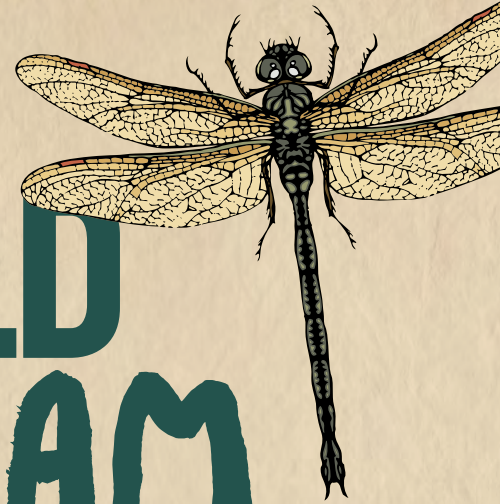
Different bat species use different frequencies to communicate, which means it is possible to determine which type of bat has pitched up in your belfry. AudioMoth, the brainchild of Open Acoustic Devices, a group of researchers at Southampton and Oxford universities, uses a full-spectrum acoustic logger and an analogue microphone centred around Silicon Labs' 32-bit Gecko processor. It writes



AudioMoth makes use of the echolocation signals bats emit to help you identify species

uncompressed audio to WAV files stored on a microSD card, and is AA battery-powered so it can be used anywhere.

The British Bat Survey has a useful setup guide at the following link: hsmag.cc/7zaPXq →



HOW TO BUILD A NATURE CAM

Attach a camera to Raspberry Pi, and you'll be able to watch the wildlife in your garden



Sparrow

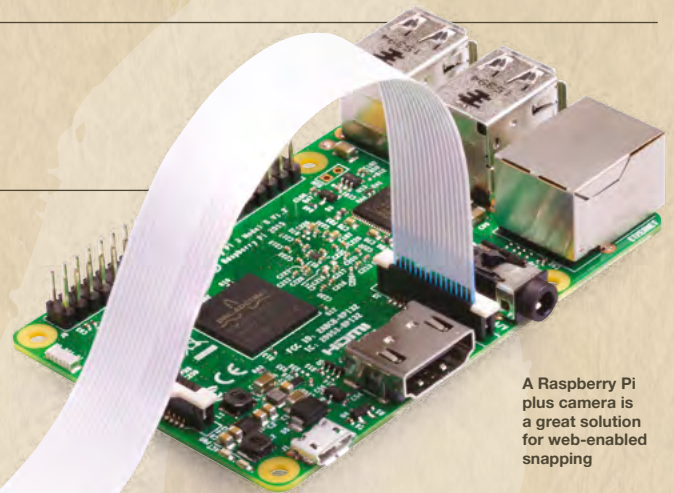
Equipped with a Raspberry Pi connected to a camera and USB power bank, we are able to spy on the wildlife in our garden. The

Raspberry Pi Camera Module

V2 is a good option here (it's less intrusive than the newer High Quality Camera, though that would make a superb critter-cam). It's important not to disturb wildlife with lighting, so use an infrared module, such as the NoIR Camera Module, if you want to snap evening or night-time wildlife activity. Connect the Camera Module to the Camera port on Raspberry Pi using the cable provided, then gently pull up the edges of the port's plastic clip and insert the ribbon cable. Push the clip back into place and the Camera Module will remain attached. (See [hsmag.cc/Olxk0z](https://www.hackmag.cc/Olxk0z).)

Set up your Raspberry Pi and let it perform any OS updates needed. (The Raspberry Pi Imager tool can help here: [hsmag.cc/XqxcJB](https://www.hackmag.cc/XqxcJB).)

You'll need a keyboard and mouse to set up the Raspberry Pi, but you can disconnect them at the end. Insert the updated microSD card and use a regular power supply to start it up (keep your power bank on charge separately while you set things up). Go through



A Raspberry Pi plus camera is a great solution for web-enabled snapping

the Raspberry Pi setup, making sure you change the default password (since it will be accessible to anyone), and connect to your wireless network. It helps if you can access this network from the garden.

Turn on the interface for the camera, and enable SSH and VNC so you can access Raspberry Pi OS remotely when it's sitting out in the garden. To do this, open Menu > Preferences > Raspberry Pi Configuration and click on Interface, then set Camera, SSH, and VNC to Enabled (see this documentation: [hsmag.cc/kdBSek](https://www.hackmag.cc/kdBSek)). Click Yes when advised that a reboot is needed.

Next, test the camera. Open a terminal window and enter:

```
raspistill -o Desktop/image.jpg
```

A preview window will appear. After a few moments, it will save an image to the Desktop. Double-click the **image.jpg** file to open it.

You can use Python to take pictures and shoot video. This is handy if you want to create a time-lapse

You'll need

- ◆ Raspberry Pi
- ◆ Raspberry Pi Camera Module or NoIR Camera Module
- ◆ Raspberry Pi case with camera mount
- ◆ USB battery pack
- ◆ YouTube account

Grey Petaltail

or video camera. This Raspberry Pi Project guide explains how to control the camera with Python: hsmag.cc/kySZqx.

Note that recording video will quickly fill up your storage space and drain the battery. A better idea is to leave the preview running and use VNC to view the camera remotely. A neater option is to hook up your Raspberry Pi to YouTube (as explained in this Raspberry Pi infrared bird-box project: hsmag.cc/Rpplim).

Open a web page and go to studio.youtube.com. Sign in, or set up a YouTube account. You will need to enable permission to live-stream. This involves providing YouTube with your phone number. Click Settings, Channel, and 'Feature eligibility', expand 'Features that require phone verification', and click 'Verify phone number'. Type in your phone number, then enter the code that YouTube sends you as a text message. For security reasons, it will take 24 hours for YouTube to activate this feature on your account.

Get your key and add to terminal

On the left-hand side of the screen you should see a menu with the My Channel option available:

In the middle of the screen you should see the Video Manager option. On the left you should see a Live Streaming option. Look for and select the 'Stream now BETA' option.

Scroll down to the bottom of the page and you should see the ENCODER SETUP option.

Here there is a Server URL and a Stream name/key. The key is shown as a line of asterisks, until you click the Reveal button. Keep the key secret and don't share it online. Copy your Stream Key to a text document (password-protect it, ideally).



You can use a USB power bank to run your Raspberry Pi wildlife camera



Open a terminal window and enter this command (replacing <key goes here> with your own key:

```
raspivid -o - -t 0 -w 1280 -h 720 -fps 25 -b 4000000 -g 50 | ffmpeg -re -ar 44100 -ac 2 -acodec pcm_s16le -f s16le -ac 2 -i /dev/zero -f h264 -i - -vcodec copy -acodec aac -ab 128k -g 50 -strict experimental -f flv rtmp://a.rtmp.youtube.com/live2/<key goes here>
```

Once enabled for live-streaming via YouTube, place your Raspberry Pi and NoIR camera module inside a waterproof case

With this running on Raspberry Pi, you can view the stream from your camera on YouTube on any computer. This infrared bird-box project explains more about the command options: hsmag.cc/Rpplim.

You'll want this script to execute on startup. Create a file for your startup script and add the aforementioned **raspivid** stream command to it:

```
sudo nano /etc/init.d/superscript
```

Make the script executable:

```
sudo chmod 755 /etc/init.d/superscript
```

And register the script to run at startup:

```
sudo update-rc.d superscript defaults
```

You can see details of scripts running at startup at hsmag.cc/VD6HnH.

Shut down Raspberry Pi and fit the computer and Camera Module inside a case (if you are using one). Position Raspberry Pi in your garden and power it with the USB power bank. It will connect to your wireless network, and run the YouTube streaming key.

Navigate to your channel on YouTube at any time to see the action taking place in your garden. □

Tip

More information on the camera can be found on the Raspberry Pi Camera Module Documentation page:

hsmag.cc/cNfuM4

Ladybird



SUBSCRIBE TODAY FROM ONLY £5



SAVE UP TO 35%

Subscribe today and get:

- ▶ **FREE delivery**
Get it fast and for FREE
- ▶ **Exclusive offers**
Great gifts, offers, and discounts
- ▶ **Great savings**
Save up to 35% compared to stores

➔ **Subscribe online:** hsmag.cc/subscribe

SUBSCRIBE TODAY

Subscribe for 12 months

Rolling monthly subscription

£55 (UK)

£90 (USA)

▶ **Low monthly cost** (from £5)

£80 (EU)

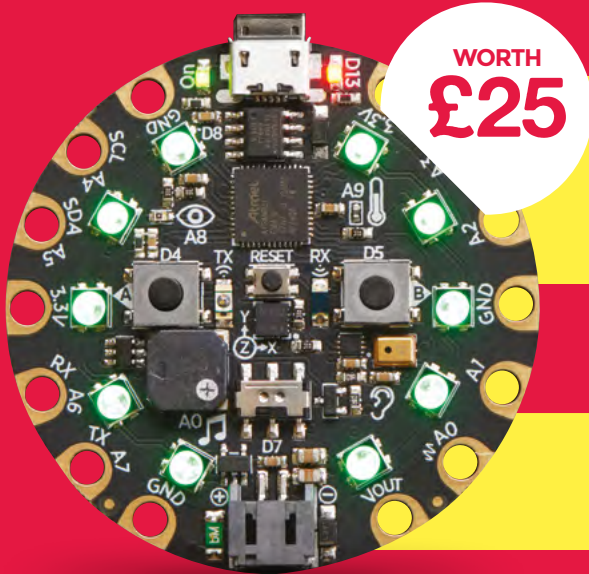
£90 (Rest of World)

▶ **Cancel at any time**

Free Circuit Playground Express with 12-month upfront subscription only (no Circuit Playground Express with rolling monthly subscription)

▶ **Free delivery to your door**

▶ **Available worldwide**



FREE!

Adafruit Circuit Playground Express

With your 12-month print subscription

This is a limited offer. Offer subject to change or withdrawal at any time.

SUBSCRIBE on app stores

From **£2.29**



👉 Buy now: hsmag.cc/subscribe



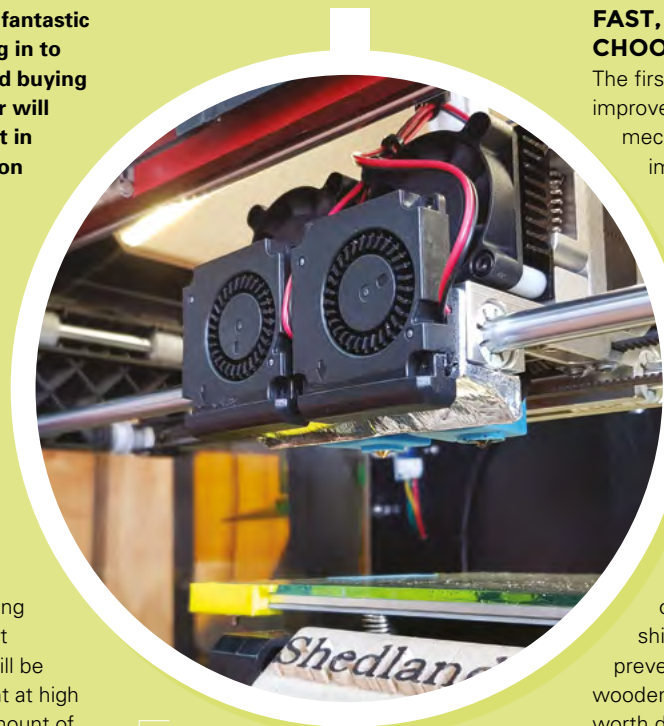
How I Rebuilt A DUAL-HEAD 3D PRINTER

Turn a cheap 3D printer into a reliable workhorse

By Andrew Lewis

A 3D printer is a fantastic tool, but giving in to temptation and buying a cheap printer will probably result in some frustration

once you start experimenting with different materials and techniques. If you have one of the dual-head Replicator clones like the CTC Bizer Series, you can do plenty of upgrades and modifications to improve your printer and prevent some common problems before they happen. Exactly how you upgrade your printer will depend on your intended use. For some people, increasing the accuracy of prints is important, while for others speed is the driving factor. In some cases, the amount of noise that the printer makes will be important. You might want to print at high temperatures, or minimize the amount of support you'll need. If you're like me, you'll just want all the upgrades! Here's what I did.



Above ♦
Two extruders
give you extra
print options

FAST, CHEAP, OR ACCURATE - CHOOSE TWO

The first upgrades are mostly designed to improve the accuracy of the printer, both mechanically and electronically. The important areas for upgrade are the z-axis and the extruder/hot-end assembly. The z-axis of the printer uses linear bearings that run on steel bars and support the weight of the print bed. The bars are held in place with a wooden bracket, bolted at the top and bottom. This bracket can work loose over time, and the bars can move around enough to affect the accuracy of the print. To fix this problem, you can simply wedge a plastic shim between the side of the case and the steel bar. The shim holds the bar under pressure and prevents it from wiggling around in its wooden bracket. It's an easy fix, and it's worth doing.

Adding an LED light strip to the build chamber of your 3D printer isn't just a cosmetic fix. Proper lighting makes it easier

Web control

OctoPrint is a 3D printer monitoring system that can be run on Raspberry Pi. OctoPrint provides an easy web interface to control printing, and allows you to add a webcam to monitor your printer from a distance. To use OctoPrint on a Replicator-style, dual-head printer, you'll probably need to install the OctoPrint-GPX plugin, which is very easy to do. While it's not really an upgrade to the printer, it is a highly recommended upgrade to your printing workflow.



to see problems as they're happening, rather than when the print is finished. If you do fit LED lights, make sure to hold them in place with screws and clips, because the adhesive used on the strips doesn't hold up well to heat.

I added silicone boots to help insulate the heater block, which means the printer can hold its temperature more easily. They also help prevent short circuits if the heater or thermocouple wires start to break down. It's worth inspecting these wires regularly, as a short circuit can permanently damage the control board.

Fixing the printer head was more complicated, but it solved a few different problems. Problem one is that some

dual-head clones still come with very poor moulded plastic extruders that don't use a spring to tension the filament against the drive wheel. This makes the motor or filament slip under pressure, which reduces the amount of extruded material. Changing out the extruder for an all-metal version (such as the MK8) is an easy way to improve reliability.

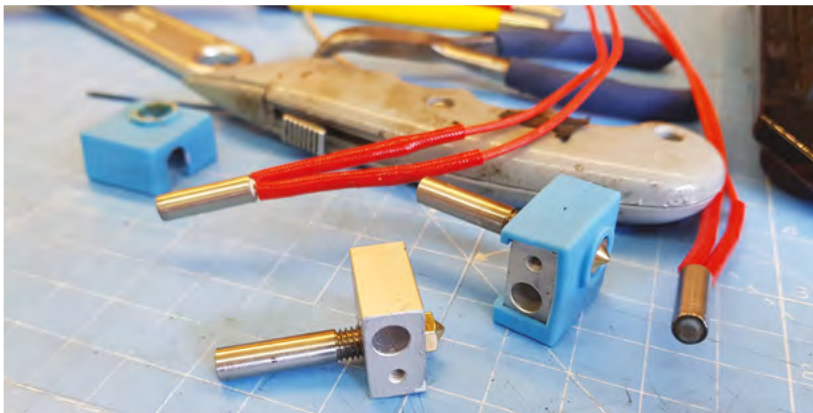
The factory-fitted fans to the print head are generally of poor quality. This makes the printer noisy, but more importantly, it makes the extruder get hot. Hot extruder parts can soften printer filament too early, leading to slippage or blockages in the print head. Swapping the fans will give your ears and your filament some well-deserved support.

Above ♦
A bit more illumination makes it easy to see when things are going wrong

The 0.4mm nozzle is a huge source of inaccuracy for some printers. Aside from the need to have properly adjusted heights for each nozzle, the hole in the default nozzle is not always 0.4mm, is not round, and sometimes has burrs on the rim that make filament behave strangely as it is being extruded. Again, it's recommended to buy replacement nozzles to fix this problem.

KEEPING COOL

The most glaring problem with most dual-head printers is the lack of an active cooling fan. Active cooling improves the finish quality of PLA prints, and allows you to perform more complex printing operations like bridging. There are some 3D-printable ducts for Replicator/Flashforge/CTC printers, but from experience, they aren't particularly effective, and will melt, even if printed in ABS. It's actually easier to make a simple bracket from some aluminium angle, and mount impeller fans to the head that way. Mounting the fans to the head is only half the problem. It's likely that your dual-head printer was built to the minimum →



Left ♦
A bit of insulation helps your hot-end stay at the right temperature

FEATURE

Different nozzles

Replacing nozzles is an opportunity for improving the functionality of the printer, rather than just correcting poor performance. If you add one 0.6 mm or 0.8 mm nozzle, and one 0.4 mm or 0.3 mm nozzle, you will be able to design your prints so that your infill and coarse areas use the larger nozzle, while the finer nozzle is used for fine detail. This can save a lot of time when printing, and will let you print lower-detail parts very quickly.

spec, so adding active cooling support will mean soldering a surface mount transistor to the mainboard to control the fan. This isn't as daunting as it sounds, but you can always compromise by controlling the cooling fan with a simple switch if you don't feel up to small-scale soldering.

Moving on from the extruder and hot-end, you can address some of the issues with the printer chassis itself. It's common for the bolts that hold the wooden panels together to work loose. This makes the printer noisier, and can reduce print quality. Make sure all of the chassis bolts are tight, and add a little dab of glue to reduce the chance of them working loose in the future.

If you are going to be printing with ABS filament, you will probably need to enclose your build plate. This is an easy upgrade, and involves nothing more than gluing or bolting some acrylic sheeting to the chassis of the printer. The corners of the chassis can be closed with 3D-printed parts (just search your favourite model repo for CTC top corners). An appropriately sized plastic storage box makes an inexpensive enclosure for the top of the printer, and it can be cut or drilled to allow cable and filament access from the top or side. A side effect of adding the enclosure is that it will reduce the noise from the hot-end fans and chassis motors. Unfortunately, the enclosure will also decrease the effectiveness of cooling fans inside the chamber, so keep this in mind when you're printing.

If noise is an issue for you, there are a few more modifications that you can make to reduce the volume of the printer. The power supply and motherboard of the printer use 40 mm fans for cooling. You can replace these to reduce the noise they make. Larger fans move more air, and so you can even experiment by using resistors to reduce the overall speed of the fan. For the PSU, you'll need to cut the case to accept a larger 80 mm fan. For the motherboard fan, it's easiest to remove the vertically mounted 40 mm fan, and cut part of the wooden base panel away and mount the fan horizontally on the base. You'll need to make some



Above & right
3D printing is great, but when you need parts that withstand heat, you might need to switch to something else, like aluminium

QUICK TIP

Printing quickly is all about balancing inertia and momentum. A light-weight hot-end with a Bowden extruder will be able to print more quickly than a dual-head carriage with head-mounted extruder motors.





Left ♦
The wiring to the x-axis motor and end-stop runs very close to the pulleys on the top right-hand side of the chassis on most dual-head replicator clones. Re-routing the cable can minimise this problem

larger feet for your printer if you make these modifications, as the new fans are bulkier than their 40mm counterparts.

As a final step, it's worth looking at preventative actions you can take to avoid some of the common problems associated with these dual-head printers. The stepper motors used on some of these printers have a four-pin plug at the motor end that can work loose. This will cause skipped steps on the motor, and can even cause the motherboard to reset in the middle of a

The stepper motors used on some of these printers have a four-pin plug at the motor end that can work loose

print. The fix is very simple – apply a dab of hot glue to keep the plug in position.

So, is it really worth making all of these changes to a cheap printer, or should you just spend more money out of the box? It's a difficult question, given that it depends on exactly what you want to use your 3D printer for. If you're looking for a commercial-quality printer and have an unlimited budget, then probably not. If you're working on a budget and have time on your hands, then you can get a commercial-quality printer for a fraction of the cost of a big-brand equivalent.

You'll also have the benefit of a deep understanding of how the printer works, and be able to do further upgrades more quickly as your needs change. □



Left ♦
The underside of an upgraded dual-head replicator clone (motherboard fan and cover removed), showing the modified PSU fan, sprung leg extensions with anti-vibration cork pads, and additional 5V PSU for an OctoPi server

Shhhh

If you're really serious about reducing the noise from the printer, you might consider replacing the bearings in the printer with bushings. This is quite a big job, and you'll have a period of 'running in' the new bushings when you might find your printer is not quite performing as you expect. Using linear bushings is a worthwhile change if noise is a real issue for you, and it will slightly improve the quality of your prints if your bearings were particularly worn or low-quality to start with. However, it does mean disassembling most of the printer, and you might find that you actually miss hearing the sound of the printer moving. If you're still interested in making the changes, then you can use igus drylin® bushings for the job. The instructions for disassembling the printer are too long to include here, but there are many igus® bushing how-to guides on the internet.



make my 3/4-inch tap connector link to a 1/2-inch shower hose. This all cost under £10, and wouldn't be that much more if you needed to buy everything.

Assembling the shower is just a case of pushing everything together, and using the cable ties to secure the pipes on the push-fit connectors. Suction cups and wires hold everything in place. This is a little untidy – I'm considering some 3D-printed parts to keep everything neat.

For historical reasons, I actually have two showers – one electric, and one powered off the main hot water system. This means I can set this mist shower up on the electric shower, so could still use the other one if required.

First, a bit of testing. The main shower uses about six litres of water per minute. This is actually quite low flow compared to many showers (which can be as much as 25 litres per minute for rain showers).

When it comes to mist showers, the question of water usage comes down to the number of nozzles you have. While it's possible to shower with one nozzle, it's not particularly pleasant. I went for five nozzles, just because that's the size of the pack the nozzles were sold in. This led to a water usage of about 1.5 – 2 litres of water per minute (depending on the nozzle settings). So, even with all these nozzles, it's a huge drop in water consumption.

There is a slight problem with the fact that showers aren't designed to work with small flow rates. The shower it's connected to has low, medium, and high settings. If I put it on the high setting, the low-flow state is triggered and the water isn't heated. On the medium setting, it doesn't trigger low flow, and the water is pleasantly warm, but when the air temperature drops in the autumn, I suspect that it'll be too cold. At which point, I'll have

to decide whether to hook it up to the other system, or switch back to a conventional shower.

The proof of the pudding is in the tasting, and the proof of the shower is in the washing. Using the mist shower is a very different experience to using a traditional shower. You barely feel the impact of the water, but it hits you over a much larger surface area, so even though there's less water, it feels more enveloping. It's perfectly capable of washing soap off, and can get shampoo out of my hair (but I have short hair – longer-haired people may struggle with this).

It's comforting, but in a different way, and when I look down at the gentle trickle of water going down the plug hole (rather than the usual torrent), my conscience feels cleaner as well. It's too early to tell whether this shower will stand the test of time, but for now, I'm quite happy. □

Left ◀
Each nozzle is held in place with wire – something that will need to be changed if this shower is to become a permanent fixture of our house

Below ⚡
You can get parts like this in garden misting kits from many sites on the internet



Safety!

When doing anything with water systems, you should be mindful of the risk of Legionella. This bacteria grows in warm water and can cause pneumonia. While a mist shower shouldn't impact the risk of the bacteria growing in the water system, it does create a lot of droplets which may be inhaled, so if there are any bacteria present, it can make them more dangerous. If you're planning on creating your own mist shower and are unsure about the risks, consult a suitably skilled person before starting.

HackSpace magazine meets...

Alicia Gibb

You've gotta fight for your right to invent things

Buying a sealed, perfect box that does only what it's supposed to isn't our idea of tech. We like to take things to bits, to see if we can improve them, or (more often) to see whether we can put them back together again without having any mysterious extra screws. Things are made to be unmade, improved, learned from. That's a philosophy that's guided all our favourite makers, including Adafruit, Evil Mad Scientist, Pimoroni, and most likely anything else that you'll ever see in these pages.

That's also the philosophy behind the Open Source Hardware Association (OSHW). This group exists to give us all as makers a middle way between big-bucks monetisation and creations never seeing the light of day, that we believe suits the small tinkerer perfectly. They're doing a good job; now, more than ever. We spoke to Alicia Gibb, executive director of OSHWA, to find out what they're doing, why we need them, and how much it costs to be an inventor nowadays. →



OSHWA's summit is hopefully going to be on 9 April 2021 in New York. Tickets are available already – fingers crossed we'll be allowed to see people again by then

HackSpace: Can you tell us what open-source hardware is?

Alicia Gibb: Open-source hardware is hardware whose files are publicly available to study, modify, remanufacture, resell, and redistribute the hardware itself. Basically, it means that you are able to make copies of that hardware as long as you give attribution and keep the same licence.

I should make the caveat that that's the most popular definition out there, but there are different flavours of open-source hardware licences. People can pick and choose a [relatively permissive] Apache-style licence, or a [more restrictive] GPL-style licence, or whatever suits.

HS: That sounds very much like the definition of free software.

AG: When we wrote the definition, we really based it on the GPL, because we didn't want to reinvent the wheel. Free software has a method: let's use that and go along similar legal lines. There's a big caveat though, which is that software is copyrighted and hardware is patented. It's a different world in the legal ramifications and how things are licensed. But the general idea of the freedoms behind what you can do, we're very much in the footsteps of the open-source software movement.

HS: What benefits do companies gain from releasing hardware as open-source?

AG: A lot of companies that we have within our fold will tell you that the biggest benefit they gain is that they make more money; they make profit because their hardware is open-source. Other benefits happen to be a really vibrant community, so if you open up your source you've got people poking at it all the time, you've got pull requests,

you've got people who are interested in personalising it for what they need to do and allowing you to then put their improvements back into your product.

It's not that every single open hardware project gets free engineering – I don't think that's why open-source hardware people do it – but it is a little added benefit. If you're going to put your source code out there to share, people will dig through it and they will let you know what's wrong with it! People have lots of opinions, and so whether or not you want the free engineering, sometimes you get it.

But there are also benefits in terms of your geolocation. The internet is pretty pervasive, which is great, but it doesn't always reach absolutely everywhere.

Because it doesn't always reach everywhere, people might be able to

"

Two heads are better than one, and open-sourcing gives your product multiple heads

build hardware in their specific location regardless of whether they can get the original, and I think that's a huge benefit for places that might have taxes so high that they can't get materials from specific countries. I was visiting Croatia a few years ago, and at that time the tax on parts ordered from an American company was so great that importers would spend more on tax than on the parts themselves. And so they were looking at creating their own boards.

It's really a way that your project, or your product, can live on in different forms through different people if you open-source it. And as well – the Arduino project is a great example of

this – people might find a way in through one of the derivatives but then end up back at the parent project, and end up purchasing and becoming part of that community.

Two heads are better than one, and open-sourcing gives your product multiple heads, and the freedom to grow in ways that you would have never thought of or which you would never have considered profitable. And you're always free to take those modifications and put those back into your product.

Those are a few of the benefits. It depends on what field you're in, what the various benefits are. For academia, there's a huge provenance benefit that can be there. Academics are always looking for citations, and this is the citation of the hardware world.

If you think about RepRap, which came out of the University of Bath, you can look at the amazing amount of growth there. If you talk about a paper that's been cited, and you apply that to the RepRap printer, it's absolutely amazing the benefits that this printer has put out in the world, and now is reaping all these 'citations' back from. There's notoriety to it too – your name is on all the hardware that gets published and attributed to, and all that.

"

HS: You mentioned that software is copyrighted, and hardware patented. What's the difference practically?

AG: One of the things that's different between copyright and patents is that copyright is automatically given to you, and it doesn't cost you anything. Whereas a patent costs a lot of money. It costs engineering time, legal time, and patent fees.

When people say "I'm open-sourcing this project because it's more profitable", there are two tiers to that. One is that they don't have to pay all those patent



fees. It's an enormous amount of money. The second is that there's a certain subset of people who are motivated to purchase open-source products, and so that drives marketing. It's a reason to buy things for some people. And I don't think it works the other way around: there are people who will choose one product over another because it's open-source, but I don't think there's a community out there that buys a product because of how many patents it has.

I've done a lot of talks, and at some point I usually ask whether anyone's ever bought a product because there's a patent on it. I've never had a single person from the audience raise their hand. But if people's values are aligned with open-source, then they'll buy a product that is open-source over a product that is patented.

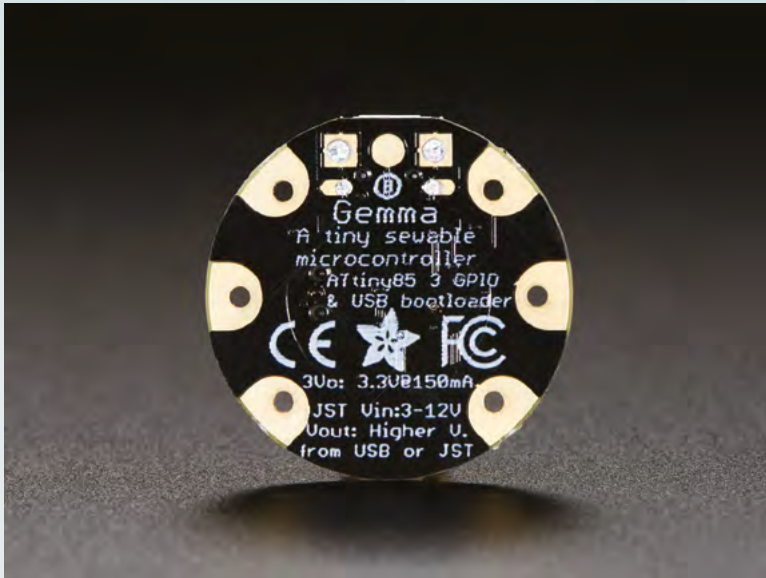
HS: I know that the Free Software Foundation spends a lot of time and effort enforcing the terms of open-source licences – effectively guarding against infringers stealing other people's work. Do you find that you have to do the same thing with patents?

AG: We haven't had the call to do that with patents; our community hasn't asked us to do that with patents. There's actually the 'Mark Cuban Chair to Eliminate Stupid Patents' at the →



Above ♦ Evil Mad Scientist is one of many maker companies producing open source hardware




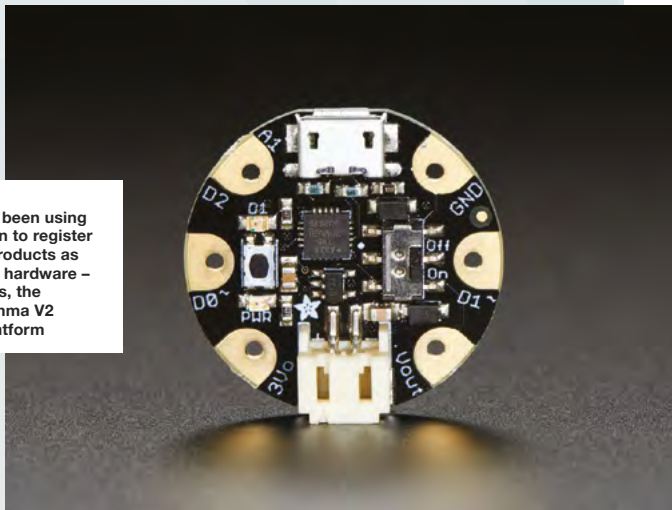


Electronic Frontier Foundation that does that a lot more than we do. That must be such a fun job!

Sometimes we get complaints that somebody's got a broken link or they need X, Y, or Z in the source, and we look into it and either agree or disagree and report back to the person who brought the complaint. But we haven't really had that to the level of patents. That might be because it hasn't happened yet, and it might also be that the US PTO (which is the only patent office that I've had any experience of dealing with) is full of overworked and underpaid workers, your stereotypical government problem, and within their line of work they're pushed to search certain places for prior art on a patent, and if they can't find it in those few places, then they'll issue the patent.

It's a really big dream of mine, but it would be awesome if we could get patent offices around the world to start searching our certification database as prior art. That would calm some creators who, I think, are nervous about opening their hardware; right now we just hope that the US PTO finds your work and you can prove prior art. Of course, you could take the person to court and attempt to prove that you had prior

Right  Adafruit has been using the lockdown to register its eligible products as open source hardware - including this, the Adafruit Gemma V2 wearable platform



art before, but that's an expense that nobody wants.

HS: How much are we talking to patent a creation?

AG: It's usually about \$50,000 in the US. That entails the patent fee, engineering fees, and legal fees. Open-source hardware is a response to the patent system by people who were just making small projects for whom the patent system is completely broken. It's supposed to help inventors and invention, but when you price most of your single inventors out of the market, you only get inventions from those large companies that can afford those patent fees.

HS: Is it mostly smaller companies and creators who you deal with, or have any big companies gone in with open hardware?

AG: Recently, big companies have realised that open-source is marketable, and people like that. They feel good about supporting open-source hardware, and they want to be part of the game. And so there have definitely been a lot of big companies that have been interested. They haven't all been successful at creating their own open-source hardware, but they're going down the right paths and the right avenues and are working towards creating open-source hardware the right way.

It's like trying to turn an oil tanker; these giant companies have done things in a certain way for a long time; sometimes their legal teams are just baffled at the concept of open-source hardware, purely because it doesn't fit with the ways they've always done things.

The interesting thing is that there are inventors in all these big companies who are all in favour of open-source hardware. The shift to open-source

product is usually a bottom-up shift, where an engineer gets interested in open-source, comes to us and says "can you help us open this?".

A lot of government entities are already doing this. Look at NASA: they've got the JPL Rover, and I like the tag line on that. It's something like 'We provide the plans, you provide the elbow grease'. And that's open source. I can't describe it any better than that. It's fantastic. So there are giant entities out there that are doing open-source hardware, and a lot more that are still scratching their heads working out how they're going to jump on board, but they are interested in jumping on board.

It's usually about \$50,000 in the US. That entails the patent fee, engineering fees, and legal fees

At OSHA, we want people who are moving in the right direction. We understand that this is a little bit scary and it's not for everybody, so we do what we can to help companies continue down that path. A lot of companies have produced products that are almost open-source hardware – we consider that a win.

Look at how Covid-19 affected the world. Open source has had a huge effect on how we're dealing with this. You look at places like Medtronic, a giant medical company in the US. They didn't exactly open-source their ventilator, but they relaxed a ton of things on their ventilator patents so that other makers could help them rebuild things. That's a huge step in the right direction. They understand that for humanity to continue down the path of survival, we might have to share

things instead of hoarding all that information as patents.

And Ford, the motor company, started manufacturing 100,000 or so face shields a day, and the design that they used was open-source hardware from the University of Wisconsin.

The benefit has changed. It's not that we necessarily care about the profit right now, but we care about efficient manufacturing more, and we care about distributed manufacturing. Allowing that to happen means that you need to share the source. For medical equipment, quality might mean life or death; whereas for a 3D printer, quality means a crappy print – it's not going to

kill anybody. Open-source hardware is only for the intellectual property. It doesn't mean that this thing has been certified by the FDS or NICE or any other government body whose job it is to keep you safe.

That's what I couldn't believe: that Medtronic got their lawyers to say "Hey, for a certain amount of time these plans, with limited use, are open to all." That would have been a giant legal hurdle to overcome. I can imagine they'd be otherwise not wanting to share things or difficult to work with or whatever.

HS: When you say that companies release 'almost' open-source hardware, what do you mean?

AG: Releasing it with a non-commercial clause. People might use the Creative Commons Licence, and then they'll add the non-commercial clause to it, but you can't do that and still call it open-source hardware. Again it's a step in the right direction, but when you put that non-commercial clause in there you're not allowing people to remanufacture, resell, redistribute these items, and that's really important in open hardware.

Those actual atoms cost money. It costs money to get the copper out →



As I mentioned, the non-commercial clause is in there – if you answer ‘yes’ to the question ‘Is your product only for non-commercial use’, you get asked to fix it and reapply. We examine the

source files and make sure that everything is there that people would need to reproduce the hardware, and that’s it.

HS: This may be a silly question, but is it taking off? What does the growth curve look like for open-source hardware adoption?

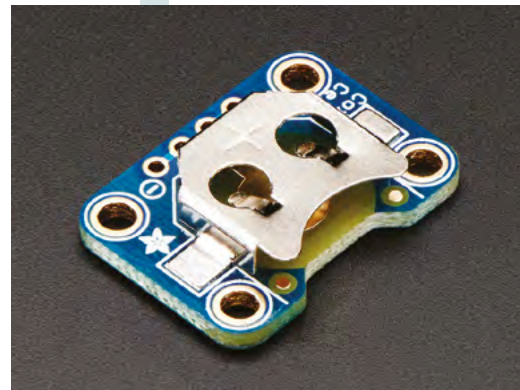
AG: I really look for that at our Open Hardware Summit, which has been going for a decade now. For me, it’s really interesting to see the sponsors who joined us ten years ago at one of the lower tiers, and how so many of them can contribute more in sponsorship, because they’re making money out of open-source hardware. And every year we get new sponsors out of absolutely nowhere, totally new sponsors, and that tells me that our reach is continuing and growing.

It’s the same thing with the certifications. Open-source hardware was really couched in electronics to begin with, but beyond that, it has grown into so many different instances. The first time I saw that someone was open-sourcing camping equipment, my mind was blown; someone’s open-sourcing hammocks and tree

straps... this is a community that we didn’t do any outreach to, but who found us somehow. I think we’re going to continue growing the fields that open-source hardware is moving into, and that’s an interesting metric for me.

In particular, I think we’re now going to get a lot of medical equipment. We’ve seen a number of certifications come in through the whole Covid crisis as well.

HS: What are you working on at the moment?



AG: Right now we’re working on what I call the Edu initiative. We’re looking to get this more onto academia. Businesses understand the benefits of open-source hardware, and understand the use cases etc., and they’re educated about the fact that open-source software exists. But in academia, professors, researchers might

of the ground and into a board, and it costs money to put components on that board. We are really adamant that you have to allow commercial use of your hardware if it is to be considered truly open-source hardware.

HS: Let’s say I’ve come up with something and I want to register my creation and resell it to the world as open-source. How do I do that?

AG: You just have to send the design files. We basically took the open-source hardware definition, and we broke that down into a series of questions on a



form that you have to answer ‘yes’ or ‘no’ – based on those questions you either get a ‘yes this is open-source hardware’, or you’re pointed towards some things you need to look at.





go to the tech transfer office (TTO) with a thing that they've made, and lot of the time they don't actually care whether it's patented or not; they're just going through the university's process of making an invention, of doing research,

you've had, or how many projects you've contributed to, all those things could be very useful on an academic CV.

We're targeting four branches within higher education. Students, first of all, letting them understand the power it gives them when they go out and look for jobs if they've released hardware with their name on it.

The second group we're trying to reach is the researchers and professors. We're providing lectures and people who can deliver lectures.

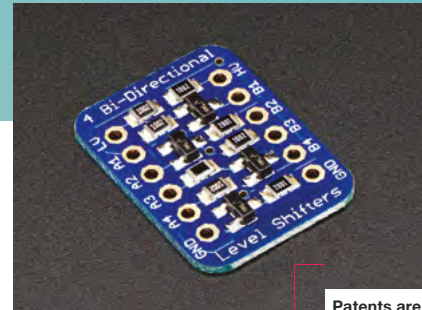
The other group is the TTOs, letting them know that there is another IP option out there. It won't be lucrative in the way that it is when universities sell patents, but it could be lucrative when it comes to getting the university's name out there on projects that people want to come there to work on.

And finally, the granting organisations, which is where the money is coming from.

We've had talks at the NSF (the National Science Foundation of the US) which is a large granting arm of universities, and it's publicly funded with taxpayer

and then the TTO will come back either 'yes' or 'no' with a patent. They will never say 'there's this other thing called open-source hardware that you may be interested in'.

Earlier on I mentioned academic citations; that helps with professors' tenure at a university, and all that other kind of stuff, so in a very similar way, if you can show how many people contributed to your open-source hardware, or how many pull requests



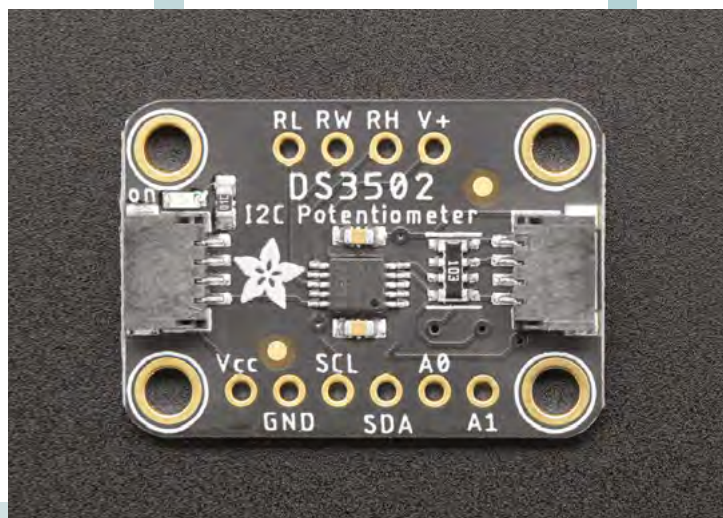
Patents are supposed to protect innovation, but a typical price of \$50,000 to register an invention effectively protects big players already in the market from smaller competition. This is the opposite of innovation!

dollars; the inventions from those dollars should be available to the people. And therefore X% of their funding is going to go towards open-source projects. That's how we're going to get the funding people educated about open-source hardware.

HS: With university patents, there must be a big *Indiana Jones*-style warehouse of patents that just never get used.

AG: There must be. And TTOs must have realised this too, because you hear about them spending boatloads of money on patents and then not getting any return back. I think they've realised that not everything needs to be patented and not everything will be lucrative as a patent. In those instances, they just tell those

professors 'we're going to pass on the patent'. And the professor misses out because they've made this invention but they don't have any claim over anything to do with it – there's no finished product that can count on a résumé towards tenure. But, it doesn't have to be like that. If you've got open-source hardware, we've got a platform to certify it. We're always excited to have new companies, new friends, and new hardware projects onboard. □





SOAP

For builds that are clean as a whistle



Mayank Sharma

[@geekybodhi](#)

Mayank is a Padawan maker with an irrational fear of drills. He likes to replicate electronic builds, and gets a kick out of hacking everyday objects creatively.

Legend has it that soap was discovered by accident, but throughout most of its history, it has been used by physicians to treat diseases. Historians often credit the ancient Babylonians as the first people to use soap, thanks to a documented recipe on a stone tablet dating back to 2800 BCE. Though many argue that the Babylonian soap wasn't used for cleaning bodies. In fact, even when the Egyptians concocted the soap, they too used it first for treating skin troubles. It wasn't until the 2nd century CE that soap became mild enough to be accepted for personal cleanliness.

Like all good things, the basic recipe for soap hasn't changed much, though its ingredients have undergone significant refinements. It's still a combination of oils with salt and water that, when

online, though they can largely be divided into two techniques. Cold process involves mixing fatty acids such as olive oil or hemp oil with lye (sodium hydroxide) to create soap over several weeks. Or, you can accelerate the saponification process by heating the mix.

Even if you aren't interested in the science of soap-making, you should understand the science behind how it works. Soap doesn't kill germs on our hands per se (though it does destroy some viruses, including Covid-19), but is very effective in removing them. Since water and oil don't mix well, water alone can't remove the germs stuck to the grease on our hands. The soap molecules act as the mediator since they can bind with both of them at the same time. When you rinse everything off, the soap carries away the germs with the water.

According to scientists, for the most effective hand washing, you must use soap to work up a lather, as the friction helps lift dirt from your skin. How long you should scrub depends on how dirty your hands are, but in the midst of the Covid-19 pandemic, most doctors recommend at least 20 seconds.

It is a sign of the times that the centuries-old quintessential personal hygiene substance finds itself embroiled in a major battle of effectiveness against sanitisers. However, washing with soap is universally recommended as a better alternative to using sanitisers alone.

"WASHING WITH SOAP IS UNIVERSALLY RECOMMENDED AS A BETTER ALTERNATIVE TO USING SANITISERS ALONE"

combined in the right proportions, undergo a chemical process called saponification to create soap.

Making soap is quite a hobby these days, since the process of creating handmade soap isn't equipment-intensive. You can find several recipes

SNOW GLOBE SOAP DISPENSER

Project Maker
DIY READY

Project Link
hsmag.cc/O19bFe

It's a bit early for Christmas gifts, but this snow globe dispenser might just be the thing you need to get your kids to wash their hands without any coaxing. You'll need a Mason jar, or something wide enough to fit Christmas decorations, and a soap pump. Drill a hole in the cap of the Mason jar that's wide enough for the soap pump. Depending on the length of the jar, you'll also have to cut off the extra length of straw. It's a good idea to keep it a little short from the bottom of the Mason jar so that it doesn't touch the decorations. Once that's ready, remove the cap and superglue the Christmas ornaments (or any of your kids' favourite figurines). Once it's dry, pour the clear liquid soap and perhaps drop in some small item that can just float around the jar in the soap. The original instructions suggest adding small amounts of glitter as well. However, be advised that some of the glitter can remain on your hands even after washing the soap. So while it does make the build look more festive, we'd suggest you skip the glitter. →



Left ♦
Dress up the jar with ribbons and gift tags to make it more appealing

TEA LIGHT CANDLE

Project Maker
THE GREEN GENTLEMAN

Project Link
hsmag.cc/34H9hm

In case the Covid-19 pandemic gets out of hand, you can take comfort knowing that your stockpiled bars of soap will not go to waste. Thanks to its ingredients, you can use the soap to create a tea light candle which, as the Instructable suggests, will help you entertain guests even during a catastrophe. A good part of the build

involves cutting two glass bottles to create a tea light holder and the cover. The Instructable has details about selecting the bottles and how to score them before cutting them with a hand-held glass-cutter. Be super-cautious when working with glass and take all necessary precautions. If you haven't cut glass before, make sure you read the maker's other Instructable, where he shares lots of tips

in detail on cutting glass.

The next major endeavour is to shred a bar of soap and do some chemistry to remove the non-essential additives, like perfumes and stabilisers, until you have the fatty acids. The process is fairly involved, though well-documented, and uses ingredients readily available in your kitchen. The final bit involves creating a wick and attaching it to a weight, before filling up the tea light holder with the soap extract, which will give off a smooth, white creamy colour when burned.



Right ♦
For best results, avoid detergent-based soaps that aren't as easy to break down chemically

LIQUID SOAP

There are many recipes for creating a bar of soap, though it's an involved process. On the other hand, creating liquid soap from leftover scraps is relatively straightforward. Gather scraps of old bar soap equal to the weight of a regular bar of soap, which is about 125 g. Shred them using a cheese grater, then heat them in a pot with about eight cups of water until the mixture turns into a smooth liquid. Michael says you should use less water if you want to make a creamy body wash instead of liquid handwash. Let it thicken overnight before you whisk it into a nice blend, and then transfer it into a dispenser. You can optionally add a tablespoon of glycerine before whisking the soap to ensure it isn't too rough for the skin. Note, however, that your homemade liquid soap won't lather as much as the store-bought variety.



Project Maker
**MICHAEL
TIMMERMANN**

Project Link
hsmag.cc/JhFGEE

Left ♦ Although Michael recycles leftover scraps of soap, in these times it's best to use a new bar

ORGANIC INSECTICIDE

Indoor plants can improve the ambience of any room but, depending on their variety, will also attract nasty little insects. While you can use store-bought insecticides, Linda has an easy-to-follow recipe to create organic insecticide that's a lot cheaper. She takes 2½ tablespoons of pure liquid Castile soap and mixes it in a gallon of tap water. You can also use distilled water if your tap water has high



concentrations of minerals, like calcium and magnesium. Linda suggests adding a tablespoon of vegetable oil to the mix, which will prevent the insecticide from washing off the plants immediately. Give it a nice stir and transfer the solution into a spray container. Linda suggests you avoid spraying on a hot, sunny day, preferably

doing it in early mornings or early evenings to slow evaporation. It's also best to move the plants somewhere you can spray freely, like a patio.

Project Maker
LINDA LY

Project Link
hsmag.cc/xePAmR

Left ♦ Visit Linda's blog for advice on selecting a soap for the insecticide, and tips on spraying it



The
MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CUSTOMPC

3 ISSUES FOR £10



FREE BOOK



hsmag.cc/hsbook

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

FORGE

HACK | MAKE | BUILD | CREATE

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

PG
74

FLEX

3D-print yourself something bendy

PG
76

KNITTING

Learn to craft fabric with this plant pot holder



PG
80

RC-DUINO

Radio-control your Arduino robot

PG
82

FURBY

Turn this digital pet into your personal assistant



PG
66

SCHOOL OF MAKING

Start your journey to craftsmanship with these essential skills

66 Mbed threads

68 Custom keyboard



PG
86

DEBUGGING

Find the faults in your circuits using an oscilloscope

PG
92

SHOWCASE

3D-print your own Lego-compatible display shelf

Do lots of things at the same time

SCHOOL OF MAKING

Do lots of things at the same time

Use Mbed's threads to run code in parallel



Ben Everard

[@ben_everard](#)

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

M

bed is an operating system for ARM Cortex-M class microcontrollers. It provides

access to many features of development boards that you'd expect. For example, turning pins on and off, accessing peripherals like PWM, SPI, and I²C, and communicating using protocols such as WiFi.

This environment has another trick up its sleeve: threading. This allows you to write code where more than one part of it appears to run at the same time, and the OS shuffles these different threads around to give them each a chunk of CPU time. This might all sound a bit complex, so let's dive in and look at a case where it's useful: separating user interface actions.

The exact process for getting started with Mbed varies slightly depending on the board you have, but it should start with plugging the board in via USB, then opening the **MBED.HTM** file on the USB device that pops up. You should be taken to the Mbed website where you can add this board and access the online compiler. If you have any problems, head to hsmag.cc/dPGWxl.

Let's start with a blinking LED. To make it interesting, we'll have it blink in a saw-tooth pattern where it gradually gets brighter, then turns off and starts again. On the board we're using (NUCLEO-H743ZI2), there are three user-controllable LEDs and a button. If your board doesn't have these, you could always hook up a couple of LEDs to pins on the board.

Left

The 480MHz Cortex-M7 processor on this board is overkill for blinking an LED, but the NUCLEO-H743ZI2 is an easy-to-use board for getting started with Mbed

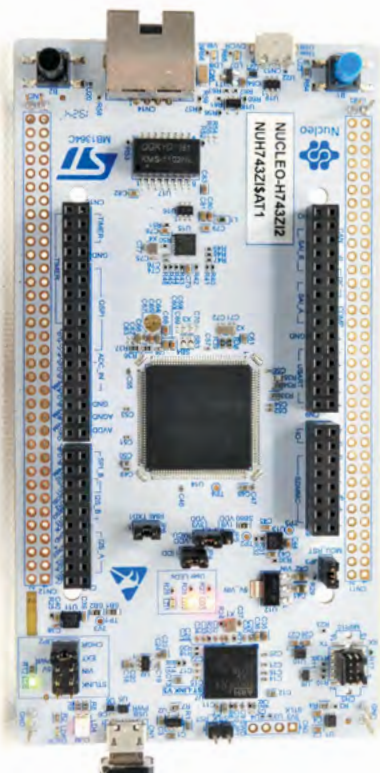
The code to slowly blink the LED is:

```
#include "mbed.h"
PwmOut led3(LED3);
void saw_tooth(int wait, PwmOut led)
{
    while(1) {
        for (float out_val = 0.0; out_val < 0.6;
out_val += 0.1) {
            led.write(out_val);
            wait_us(100000);
        }
    }
}
int main()
{
    led3.period(0.001f);
    saw_tooth(100000, led3);
}
```

Mbed is a form of C++ in much the same way Arduino is, but there are a few differences. Firstly, there are no **setup** and **loop** functions. Instead, there's a **main()** method that runs when you start to program. Generally, you'll include a loop in here, or as we have, include a loop in a separate function called from here. You'll see that some keywords are different, for example, there's **wait_us** for pausing rather than **delay** as in Arduino.

This blink is straightforward, but what if we wanted to blink another LED at a different speed? If we were writing a single-threaded program, we'd have to do something like have a loop with a counter and then check the counter to see how bright each LED should be at the time, and turn it on and off appropriately. However, with a multi-threaded OS like Mbed, we can simply start a thread and put the code in there, and it'll run side by side with the original code.

This code displays a saw-tooth pattern on LED3 while LED2 blinks on and off (there's no PWM available on LED2 on our board, so we can't fade it in and out).




```

#include "mbed.h"
PwmOut led3(LED3);
DigitalOut led2(LED2);
Thread thread_led2;
void saw_tooth(int wait, PwmOut led)
{
    while(1) {
        for (float out_val = 0.0; out_val < 0.6;
out_val += 0.1) {
            led.write(out_val);
            ThisThread::sleep_for(wait);
        }
    }
}

void blink2() {
    while(1) {
        led2 = !led2;
        ThisThread::sleep_for(1000);
    }
}

int main()
{
    led3.period(0.001f);
    thread_led2.start(blink2);
    saw_tooth(100, led3);
}

```

As you can see, we can just call the thread's start method with a function (notice that there are no brackets at the end of `blink2` when we start the thread – that's because we're passing the actual function, not the result of it). This lets us run two threads side by side with each one turning an LED on or off. Another difference is that we've used `ThisThread::sleep_for()` to pause a thread rather than `wait_us()`. You can use either, but the former is a little more efficient.

This threading makes it easy to control different 'things' that need to run simultaneously, particularly if they don't need precise timing between the two (or more) things that are running. Just because things are running in different threads, they don't have to be completely independent. Let's add a speed controller to our simple LED flashers. There's a user button on the board we're using, so we'll use this to slow down the speed of the flashing. Pressing the button slows it down until the ninth press which goes back to the original speed.

```

#include "mbed.h"
PwmOut led3(LED3);
DigitalOut led2(LED2);

```

```

DigitalIn user_button(USER_BUTTON);
int speed = 1;
Thread thread_led3;
Thread thread_led2;
Thread thread_button;
void button() {
    while(1) {
        if (user_button) {
            while(user_button) {
                ThisThread::sleep_for(10);
            }
            speed +=1;
            if (speed > 9) { speed = 1; }
        }
        ThisThread::sleep_for(100);
    }
}

void saw_tooth3()
{
    while(1) {
        for (float out_val = 0.0; out_val < 0.6;
out_val += 0.1) {
            led3.write(out_val);
            ThisThread::sleep_for(50*speed);
        }
    }
}

void blink2() {
    while(1) {
        led2 = !led2;
        ThisThread::sleep_for(500*speed);
    }
}

int main()
{
    led3.period(0.001f);
    thread_led2.start(blink2);
    thread_led3.start(saw_tooth3);
    thread_button.start(button);
}

```

Again, you can see that we use a new thread for the button. This allows us to use a loop to wait for the user to release the button without having to worry about the timing of the other things running at the same time. The three different threads communicate via a global variable called `speed`. In this case, it works because only one thread updates the variable while the other two read it. There are a few methods Mbed provides for allowing more complex communication between threads including mutexes, semaphores, and queues. However, we don't need them for our simple program. □

YOU'LL NEED

◆ **Mbed-capable board** with two LEDs and a button

HOW TO BUILD YOUR OWN KEYBOARD



CUSTOM PC

This tutorial originally appeared in Custom PC magazine issue 203. To get tips and tricks on getting the most out of your computing hardware each month, head to custompc.raspberrypi.org

YOU CAN MAKE A UNIQUE KEYBOARD TAILORED TO YOUR EXACT SPECIFICATION, AND IT'S NOT EVEN THAT DIFFICULT. RYS SOMMEFELDT GUIDES YOU THROUGH THE WHY, WHAT AND HOW OF BUILDING YOUR OWN KEYBOARD



The mechanical keyboard market is brimming with models from the usual PC component vendors as well as a vast network of lesser-known companies. They usually follow a well-worn formula of RGB lights, Cherry MX mechanical switches, and that typical black tactical chassis design that usually marks out a product as being aimed at gamers.

There's nothing wrong with mechanical keyboards like that, so why go to the effort of building your own? The key reasons are the endless amount of choice and customisation available and the ability to build a device with a truly premium typing experience.

I'm writing this feature on a board that no vendor of pre-built boards can come close to replicating in terms of feel, sound and general

typing experience, and of course the overall design is unique, which is a key hook for many people.

I'm a writer at heart, so my boards are usually optimised for long sessions in my favourite text editor. But I could just as easily have put together a board that was optimised for gaming, data entry or even tried a completely non-standard layout and key mapping, with the aim of increasing typing rate or helping to avoid repetitive strain injury.

Having built almost two dozen custom boards over the year, it's fair to say I'm hooked. It's a deep rabbit hole of a hobby, there's no doubt about it, but having the ability to put together something that's unique and that you've designed and tuned, is a big draw. Just beware that you'll never only build one.

THE GROUP BUY

Because building a custom keyboard is a very niche activity, the hobbyist scene around it is yet to tip over into the kind of volume that justifies long-term mass production. Stuff gets made in low volumes instead and is therefore usually bought with what's known as a group buy (GB).

This is where a designer will pitch an idea for a keyboard or keycaps in a process called an interest check (IC). If enough people signal they'd be willing to pay for it, a GB is formed, so people can pitch in their cash and the designer can place an order, without taking on all of the cost. The designer then works with the manufacturer to take their idea to production.

It's often a long and expensive process, but there are shops that sell more generic custom keyboard parts, with some keycap sets being easy to obtain without a long lead time or breaking the bank. Likewise, buying standard Cherry MX switches is easy and you can readily get hold of bare bones keyboards with a base and switches but no keycaps. It's just that the end result won't be as custom, nor as premium.

Given the economics of buying components individually and participating in group buys, it can get expensive. The keyboard used to write these very words is worth around £750 fully built, and that's increasingly the kind of price you can expect from limited-run boards made by designers such as Smith and Rune, ProtoTypist and LZ. There are designers who offer a discount in larger numbers, but it's rare – the price is usually driven by rarity and quality.

As for where to find more information and get in on those GBs, Geekhack is your friend. It's a site where designers post their interest checks for feedback during the design process, and then their group buy details if everything is successful. Also, check out the custom keyboard community on Reddit and Discord

The switch plate sits at the heart of any custom keyboard, and while metal is the most common material, others such as carbon fibre are used too



(reddit.com/r/MechanicalKeyboardsUK and discord.gg/qKqwpag), where you'll find each major storefront or designer has a server that covers what's happening with their products.

CASES, WEIGHTS AND PLATES

There are six main components to a custom keyboard: keycaps, keyswitches, stabilisers, PCB, switch plate and case. We'll get to the first four in due course, but the last two provide the foundation of your custom keyboard's design.

The switch plate is, as its name suggests, the central plate onto which the keyboard's switches are mounted. The plate will either offer a single fixed key layout or some degree of variation, so you can choose the layout that suits you best. For example, on the keyboard I'm using now, the plate supports a handful of different possible layouts, including the common US ANSI and UK ISO layouts.

Often the plate is a separate physical piece from the case mounting to it, but sometimes the plate is integrated with the keyboard's top case as a single piece. It can be easier to work with an integrated plate, but a separate one is usually more flexible. Regardless of which style the board uses, most custom keyboards

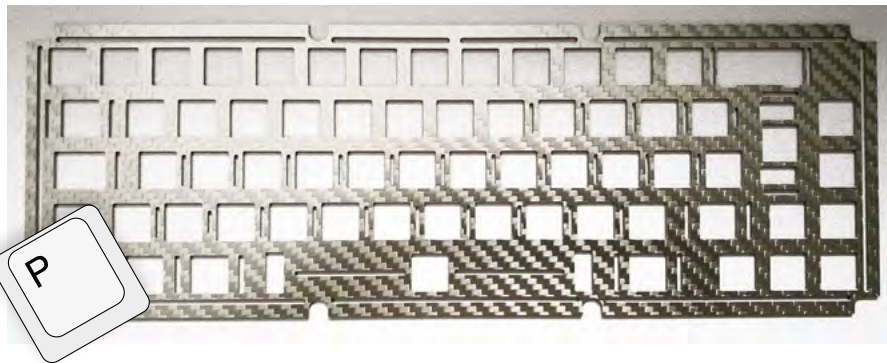
are sold as a kit case, switch plate and PCB kit.

As far as the overall shape of the keyboard, there are some common sizes, such as 60, 65, 75 and 80 per cent (often called TKL) and full-sized (sometimes called 1800), all of which roughly refer to the size of the board relative to a full-sized, 105-key unit.

There are no absolute standards though, so always double check exactly what you need. Likewise, for the key layout, there are some standards, such as US ANSI or UK ISO, but designers can and do play fast and loose with them, and there are many esoteric variations, such as ortholinear layouts where all the switches are in line with each other.

Your choice of layout depends on how you intend to use your keyboard, and how you're optimising your workspace. I tend to choose smaller 60 and 65 per cent boards since they provide more room alongside for a trackpad or mouse, and because I very rarely need a numpad or the F-row. However, having tried a bunch of 60 per cent boards, where you just get the core alphanumeric keys and modifiers, my preference has shifted towards 65 per cent boards. Having an arrow cluster is very handy, and the handful of extra keys (Del, PgUp, PgDn, Home, End) are useful both in themselves and for mapping other functions.

Along with the layout, the material from which the case and plate are made are important factors. Plates that use hard metals, such as brass, will feel very rigid as you're typing, and will sound different to aluminium. Non-metallic materials are increasingly common for plates too, especially carbon fibre and certain thermoplastics. Entire keyboard cases can even be made out of semi-transparent or dyed polycarbonates.



TUTORIAL



The switch plate and case combine to form the main structure and layout of the keyboard

High-end keyboards also tend to come with internal weights to give them some heft and prevent them from moving around on your desk as you type, which affects how the board sounds. And while you may rarely look at the bottom of a keyboard, designers will often use weights to provide hidden external detailing.

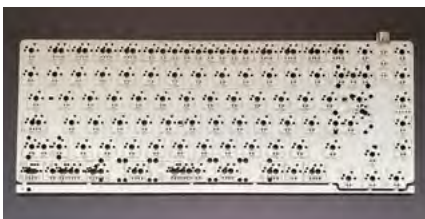
The most common arrangement for a custom keyboard, called top mount, has the switch plate affixed to the top case, and then the bottom part of the case attaches to that assembly, but there are umpteen different variations. The most complex mounts in the hobby today are of the isolation variety.

They use layers of isolating material to surround the plate and effectively suspend it inside the case, so that when you're typing, the noise isn't transferred to the case material as much, and the feel is cushioned.

PCB

Next on our shopping list is the PCB, which is there to take the input from your switches and turn it into something that your computer can understand. Although you can get cases and switchplates that fit a variety of PCBs – and vice versa – PCBs generally come with the case and plate, forming a full kit that you only then need to source stabilisers, switches and keycaps to complete. If you're buying separately, you can expect to pay upwards of £25, with some of the most premium PCBs costing double that.

Keyswitches don't do much of value without a PCB



A large metal weight and isolation mounting pads make the Iron 165 by Smith+Rune a pricey board

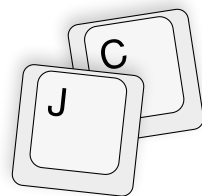
Most PCBs, or at least the majority of models designed outside Japan and China, are programmable and use an open source keyboard firmware called QMK. QMK supports a wide range of embedded microcontrollers that read the switches' signals as you press them, working out what you've pressed for how long, and turning that information into a USB Human Interface Device (HID) signal.

Some PCBs feature integrated lighting, which you can also program via QMK, and activate and modify via key combinations on the board. Some PCBs also feature spots for you to solder in LEDs yourself, including in-switch LEDs that light up underneath the

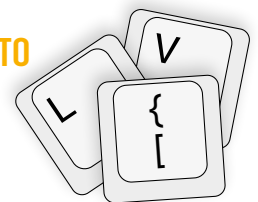
SWITCHES

The custom keyboarding world has almost completely standardised on Cherry MX-compatible switches, with most boards and keycaps sold today supporting them. Plenty of PC peripheral companies such as Logitech, Razer and SteelSeries have designed their own mechanical switches but you won't find them used in the custom market.

However, there is also some niche support for high-end rubber dome switches from a Japanese company called Topre, which are used in pre-built keyboards from the likes of PFU, Leopold and Realforce. There's also a very small niche for boards that take Alps switches,



HIGH-END KEYBOARDS ALSO TEND TO COME WITH INTERNAL WEIGHTS TO GIVE THEM SOME HEFT



keycaps. There's a wide array of options here, from a handful of single-colour LEDs to full underglow lighting and per-switch RGB.

The last thing to note about PCBs is whether or not you need to solder the switches. The majority of PCBs require soldering, but if you're not comfortable soldering, there are also hot-swappable PCBs with sockets into which you just push the switch.

They're much easier to assemble but are generally more expensive and can't support much in the way of layout flexibility.

but considering these switches are no longer made and you need to harvest them from older boards, it's not a sensible place to start.

The choice of Cherry MX-compatible switches has also exploded, especially in the past year or so. There's an increasing number of companies other than Cherry making MX switches from all manner of plastics, and with all manner of springs and stems, making Cherry's own lineup look positively boring. It's no stretch to say that the custom keyboard community has moved on from

Cherry switches almost completely, although of course, their ready availability makes them a good place to start for beginners.

As for types, there are three top level categories of switch: linear, tactile and clicky. Linears have no tactile event of any kind in the switch travel, other than when you bottom the switch out. Tactiles have a tactile bump as you push pass a point in the switch travel before bottoming out. It's usually at the point where the switch has activated so you can get the feedback you've pressed enough to activate the key without having to bottom out. Clicky switches are tactile in nature too, but the tactile event has an audible click as well.

There are three factors to consider for springs: the amount of force required to actuate the switch, the amount of force required to bottom out, and the progression of the spring rate as you press it. Each of those attributes affects how the switch feels, which is why you can buy switch testers, for trying anywhere from a handful to around 100 different switch types, to see which you prefer.

Switches cost from 20p to £1.50 each, depending on desirability. Some switches have attained meme status, usually after being used by a popular build streamer making a board for a high-profile gamer.

The more hardcore end of the community is also fond of creating Frankenswitches, using parts from two or more switches to create a new one. It's very difficult to get a good read on whether certain hybrid switches are actually better than a stock switch, but certain hybrids break through and get enough exposure that you can tell there's something to them. Holy

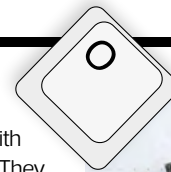
Pandas are the canonical example, taking the tactile stem from one switch using it with the housing, spring and leaf from another. They are top-tier tactiles when tuned.

KEYCAPS

Keycaps are the start for most mechanical keyboard enthusiasts. All it takes to customise your typical MX-switched keyboard is to pull off the caps and buy some compatible alternatives. There's a vast range of colours, shapes, legends (lettering/fonts) and build materials, making it fun to simply mess around with this one aspect of custom keyboards.

Aside from style considerations, the type of plastic and manufacturing process are important factors, as the different types of plastic all wear differently over time and the manufacturing can determine how long the legends on the keycaps stay looking good. Most notable is the trend for double-shot PBT caps, which use the tougher, more chemically resistant PBT plastic (compared with ABS) and are moulded twice (double-shot) such that the legend on the cap isn't painted on but formed from a second coloured plastic that runs the full depth of the cap, so it won't wear off. Plastics remain the best choice for keycaps due to the light weight and ease of moulding, but some designers have tried low volume manufacture in other materials, including metal.

The final aspect to consider when it comes to keycaps is their profile or shape. The most common are Cherry profile caps, which can be found on the vast majority of pre-built boards, and most keycap group buys are in this profile too. However, Cherry profile caps



There's a vast array of switch manufacturers and styles, but they're usually Cherry MX-compatible

are relatively short in height (compared with other mechanical keycap profiles) and the bare minimum of plastic is used to make them, so they're light and thin, which affects how they sound.

Instead, a lot of keyboard makers prefer larger keycap profiles. Keycaps in the MT3 and KAT profiles are my favourite, since they're higher in profile than Cherry caps and are quite a bit thicker than Cherry-profile sets, so they have a different sound. If you want to see what sort of designs and profiles are out there, take a look at keycaplendar.firebaseio.com

STABILISERS

Most keycaps are attached to the switch underneath by a retaining stem in the centre of the cap, which is fine for normal single-width (1U) keys. However, for wider keys, such as a normal-width Backspace (2U), ISO Enter, and especially spacebars (commonly 6U+), a stabilising system is required. Without a stabiliser, the switch will be very wobbly and, if you press anywhere off-centre, there's a chance the switch won't activate.

Stabilisers hold up the extremities of the cap on retaining stems that are joined with a wire, so the whole assembly moves up and down with the switch when you press the cap. This helps the longer caps feel stable and lets you press them off-centre without any issues.

Stabilisers take two basic forms: plate mount and PCB mount. Plate mount stabilisers are more common on pre-built boards than customs, but you'll still find them on some customs too. They aren't as stable as PCB-mounted ones due to the fairly flimsy clips that are often used to mount them. PCB mount stabilisers tend to be screwed to the PCB from the underside, making for a more secure fit.

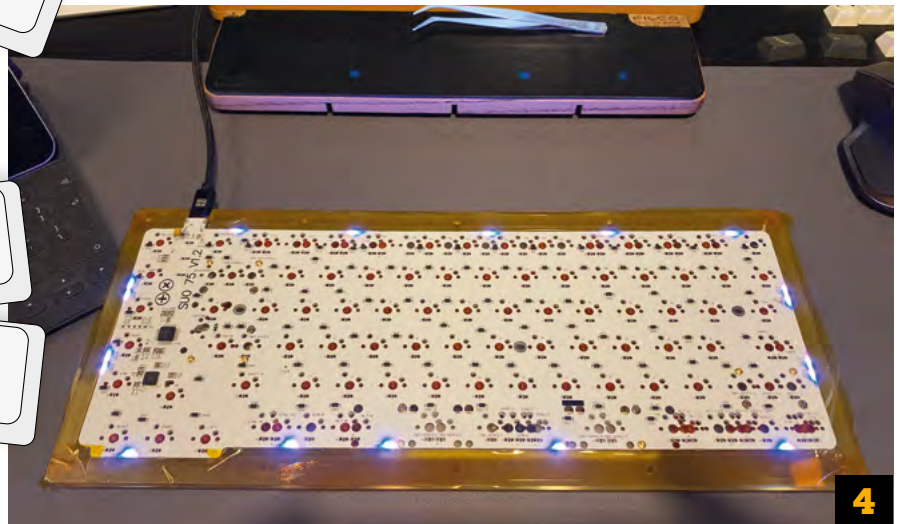
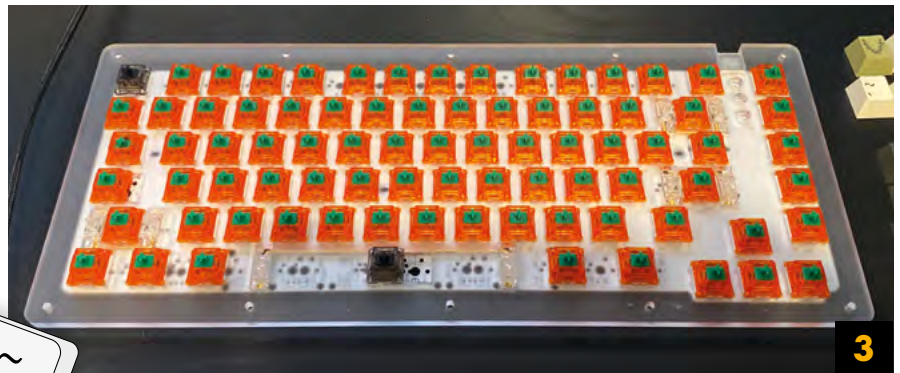
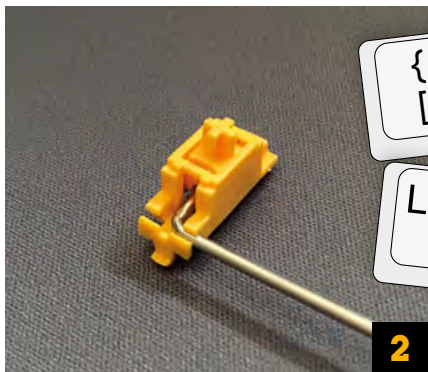
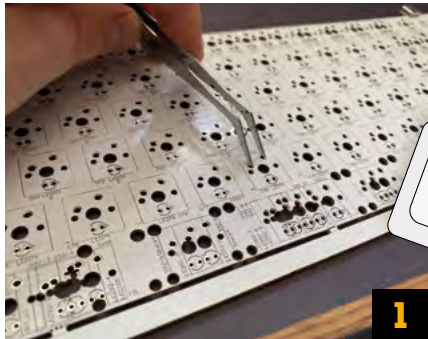
Since stabilisers are just a few pieces of plastic and a metal wire, they're fairly cheap, at around £4 for 2U and £5 for the longer ones for spacebars. That said, some vendors sell sets of stabilisers that the community considers to be higher quality, which cost more.



Different keycap profiles such as these MT3 caps make for a different feel and sound

TUTORIAL

HOW TO BUILD YOUR KEYBOARD



1 TEST THE PCB

As soldering is hard to reverse, always test your PCB first. Plug it in to your PC using a USB cable, then short each switch by touching the two switch pads with a pair of tweezers to see if the switch activates. Using a program such as Switch Hitter will let you also see if the keys that don't print anything, such as the Shift keys, are also working properly. It happens rarely, but it pays to be careful. The suo PCB by TX Keyboards was my choice for this build. It supports in-switch and underglow LEDs, and is compatible with the HJ75 kit.

2 FIT THE STABILISERS

Next, you need to attach the stabilisers to the PCB for all of the keys that need them. On a UK ISO build without a numpad, that's usually two 2U stabilisers for backspace and enter, and one 6.25U stabiliser for the space bar, but double check the layout you want to build to be sure. Push your stabilisers into the right mounting holes on the board and screw them to the PCB through the bottom with the provided screws.

3 MOUNT SWITCHES ON THE PLATE AND PCB

The next step is getting all of the switches into the switch plate and pushed into the PCB. Some plates are tight due to the fine machining tolerances, so don't be afraid to apply a fair amount of pressure, to make sure your switch is securely planted. Make sure the metal legs on each switch are straight before you push them, to minimise the chance of bending them.

Start with the corner switches and a few in the middle, to ensure the plate and PCB are aligned and nicely anchored. This should mean the rest of the switches slide into place easily, reducing the chance of any pinch points. You can choose to solder those initial switches, if you're comfortable they're perfectly in position.

Something to look out for when fitting the switches is that mounting points may be ambiguous on plates or PCBs that support different layouts. In such situations, the trick is to fit the required keycaps onto the offending switch and its neighbours, and use those as a

guide on where exactly to affix the switch. This scenario is especially common on the bottom row where it's most common for people to tinker with the space bar size and key layout.

When you have all of your switches in place and you're happy that they're snugly in place, lift up the whole assembly and eyeball that everything is lined up perfectly, and that all of the switch legs are through their holes and that none of them is bent. Making sure switches are flush to the plate and against the PCB is the most important job when building your board.

For this build I chose the TX Keyboards HJ75 case and switchplate kit, which is a 75 per cent size layout and includes the aluminium top, and several clear acrylic plates. I also used C3 Tangerine linear switches lubed with Krytox 205 Grade 0 grease.

4 SOLDERING

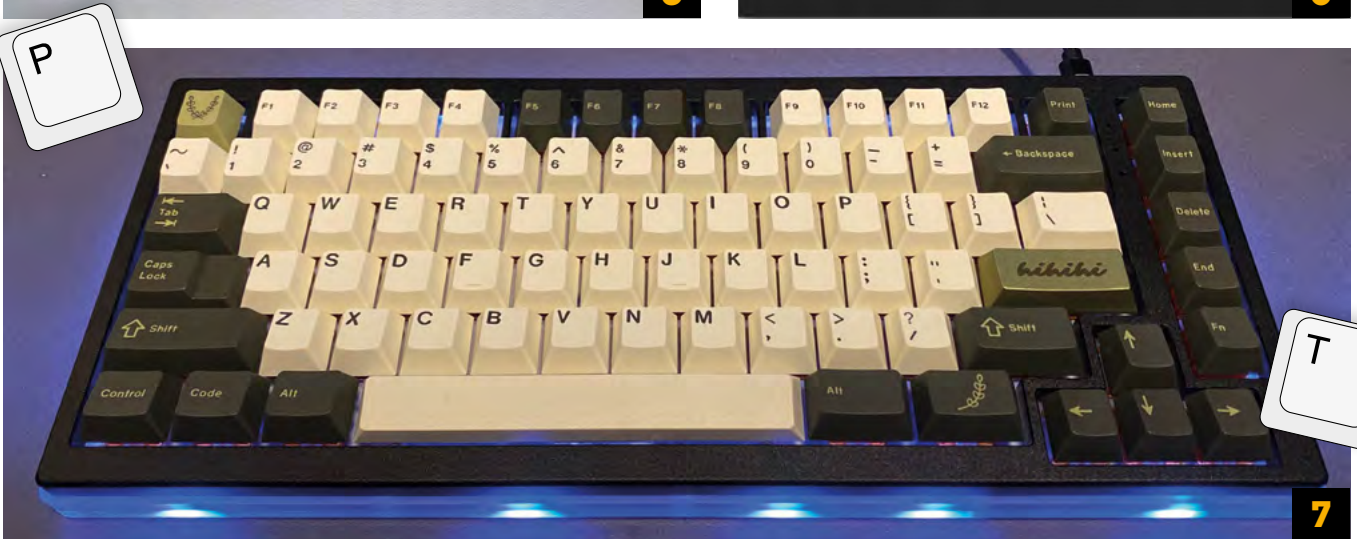
The next step is to solder all of the switches to the PCB. There are just two pins per switch so there's very little to be done other than work your way across the board, taking your time



5



6



7

until you've soldered each switch. Double and triple check you've done them all, since it's surprisingly easy to miss one.

After you think you've got them all, test your PCB again. With all the switches installed, this is a much easier process than before – just plug it in and press each bare switch and make sure it registers.

5 CASE ASSEMBLY AND KEYCAP FITTING

Every board is different here because of the variation in potential mounting options, but the basic idea is to get the switch plate retained to the top case first, before assembling the rest of the case.

Some cases only have a single piece and the PCB screws into it with nothing else to do; others are more intricate with top and bottom pieces, gaskets to lay in the right place for isolation and damping, and some have outboard USB ports that you have to mount separately, that connect to the main PCB with a small connector.

Follow the instructions for the case and other parts that you have and just proceed with caution. Now is also the time to add your keycaps, and for this build, I used GMK Olive keycaps and a metal RAMA Works enter key.

6 PROGRAMMING

Most custom keyboards have a PCB that supports the (qmk.fm) firmware. QMK is a highly flexible system that allows for almost endless remapping and configuration of your board. There's an online configurator (config.qmk.fm) that lets you remap your board and generate the new firmware file to flash the PCB once you've set it up, and the documentation site (docs.qmk.fm) has loads of useful guidance on how to get started flashing and programming your board.

If you're lucky, your board might also have a QMK-based firmware that supports a great configuration tool called VIA that requires no extra flashing to remap your layout.

Not all QMK-compatible boards support VIA, so head to (caniusevia.com) and take a look at

the board and PCB list to see if the one you're interested in is supported. New boards are being added to VIA regularly, so even if your particular board isn't on the list today then it might be soon enough.

7 YOU'RE DONE!

There's so much variation in builds that it's impossible to show you examples of every possible combination of ideas. However, almost every board you'll find online will come with an associated guide or thread somewhere that explains the build process, or you'll find that one of the major keyboard streamers has built one on Twitch or YouTube and you can watch how they did it.

Build streams offer a great way to get a feel for the process, and to give you exposure to the variety of what's out there. We highly recommend checking out (topclack.com) and (taehatypes.com). You'll find their channels on Twitch and YouTube, and between them, they've built hundreds of different boards of all types over the years.

Flex filaments

3D-print your own bendy creations



Ben Everard

[@ben_everard](#)

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

A

ll 3D-printed parts are flexible to some extent. The question is, how much flex do you need? Regular PLA can be used if only a little bending is required (and if the model is well-designed).

PETG and nylon are both a little more flexible, but when most people think of flexible filament, they

think of something that feels like rubber, and in 3D printing terms, that usually means thermoplastic polyurethane (TPU). NinjaFlex is the most famous form of TPU, but it's sold under other brand names (and non-branded) as well. TPU isn't the only option, and more generally, flexible filaments are known as TPE (thermoplastic elastomers). TPU is a type of TPE.

Flexibility is a, well, flexible term and what's flexible enough for one use may not be flexible enough for another. While you can compensate for this to a certain amount by adjusting infill percentages and wall thicknesses, really, you need to pick your filament to match your use case. The flexibility of a filament is measured on the Shore hardness scale (see box).

BEND IT, STRETCH IT

Printing flexible filament provides certain challenges. The inherent stretchiness of the material makes them less suitable for Bowden tube extruders compared to direct drive. You can get them to work with Bowden machines, but particularly as you go down the Shore hardness scale, you'll have to be very careful with your settings. Flex filaments are also very prone to stringing, and you may find that this is something that you just have to live with, and not something that you can solve with slicer settings.

As the prints are flexible, any thin parts are prone to flexing as you try to print on top of them, so



Right Two different designs of cable ties printed in TPU

taller structures have to be stiff enough to stay still when printing. Keeping the speed slow can help with this, but you may find that certain designs that are printable with stiffer plastics just don't work with flex.

Another problem with flexible filament is that they tend to be very sticky, and getting prints off the print bed can be a challenge, and if you're not careful, it is possible to stick them so tightly that they damage the bed. Depending on your print surface, you might need to add something such as tape or adhesive.

Always check with your particular filament before accepting default 'flex' settings in a slicer

Prusa recommends using Windex to clean its beds. On our printer with a flexible bed, we find that prints come off OK, as long as we don't clean the print bed with IPA each time (as we do with PETG or PLA).

Flexible filaments don't require really hot temperatures, so most printers should be able to manage the heating. Around 220–230 degrees C is common, though you may get away with a little less. Though, as there are a few different materials used, always check with your particular filament before accepting default 'flex' settings in a slicer.

Printing in flex adds a whole new set of problems you can solve with your 3D printer, though it comes with its own set of design and manufacturing challenges. If you've got a 3D printer capable of it, it's well worth experimenting with some of the different options to give you an idea of what's possible. This author is currently working on a 3D-printable sandal design, which will hopefully be shared in these pages before the end of summer. □



Left □ Depending on the infill percentage and the hardness, flex filaments can be easy to bend, or fairly stiff

MEASURING HARDNESS

Hardness is measured on the Shore hardness scale. Rather confusingly, this isn't one scale, but several. The A and D scales are most relevant to 3D printing. Typically, stiff 3D printer filaments will be measured on the D scale and flex filaments on the A scale, but there is some overlap between the two: 100A is 58D.

Flex filaments are usually 95A (or 46D) and below, with the most flexible being around 74A. Some manufacturers differentiate between 'semi-flex' and 'flex', but there's not really a standard for what fits into which category, so we'd recommend going by Shore hardness rather than a rating like this. Picking the right filament is probably more important with flex than it is with most other types of filament. Many filament sellers offer small samples of different filaments for reasonable prices, so you can test out a little bit before committing to a large order.

Knit a plant pot hanger

Show off the results of your gardening with a simple décor hack



Rosie Hattersley

@RosieHattersley

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

Gardeners have been rejoicing in one of the best spring growing seasons for years. Hanging up your plants in a window or by a doorway effortlessly brightens up a home, while also showing off your best

blooms. A hanging plant pot holder is also a great way of presenting a plant as a gift.

Knitting a plant pot hanger is an easy option: you only need to learn a basic knitting stitch and be able to sew up the edges. Visual interest is created by using large needles to create a holey effect. We've used a second knitting technique to make the straps that hold up our glass jar, setting you up with two skills by way of an introduction to the mindful and creative craft of knitting.

We recommend bamboo or plastic needles for this project. Chunky ones are much easier to use. Buy them from places such as Flying Tiger and Hobbycraft, as well as knitting websites such as [Lovecrafts.com](https://www.lovecrafts.com) and [Deramores.com](https://www.deramores.com).

For visual impact, choose a bright yarn colour. We're using a chunky (denoting yarn that's knitted on 5mm or 6mm needles) acrylic yarn. It's hard-wearing, doesn't shrink if it gets wet, doesn't attract moths – the bane of knitters' lives – and comes in hundreds of different shades. Bamboo or cotton are good, non-animal-derived yarn options. You could also use twine or string.

CASTING ON

Casting on creates the stitches that are the basis of your knitted fabric. Wrap the end of the yarn round

Left Straps for our knitted plant pot hanger were knotted together at the top to form a strong loop



YOU'LL NEED

- ◆ 50 g yarn chunky acrylic or cotton
- ◆ 12 mm single-pointed bamboo knitting needles
- ◆ 10 mm double-pointed knitting needles or a knitting Nancy
- ◆ A glass jar or decorative plant pot
- ◆ A blunt-end sewing needle
- ◆ Sharp scissors
- ◆ A metal curtain ring [optional]
- ◆ A hook to hang up your plant pot



QUICK TIP

Use this children's rhyme and useful video clip to help you knit your first stitches: hsmag.cc/8iWSE9.

Above ♦ Bring the working yarn round the outside of the right needle, then forward through the middle of the needles

Below ♦ The cast-on row has all the stitches close together. Knitting the first row will pull them into shape



your finger, slide the end through the loop to form a slip-knot, and place it loosely over the pointed end of one of your needles. Gently pull the tail end of the yarn – the bit that isn't attached to your ball of yarn – so it's at least 10 cm long. This way you're less likely to lose your knitting as it unravels, as you manipulate it while working out your first row of stitches. Wiggle the slip-knot so it's not tight against the needle. Lay the knitting needle on the table and place a finger on top to lightly hold it in place. Gingerly push the other needle up into the slip-knot so that both needles are embraced by the knot and are lying parallel with the knot, approximately 3 cm below the pointed ends.

Using a thumb or finger to hold the tail yarn against the left needle, take the working yarn (the bit attached to the ball of yarn) around the outside of the right needle and to the back. Pull it forwards between the two needles. Use your thumb to hold this yarn against the right needle. Lever your arm upwards so the tip of the right needle points down and can slip underneath the loop formed from the yarn brought forward a moment ago. Use the right needle to pull this loop up and twist it, before hooking it onto the left needle. You should have two loops on the left needle, meaning you now have two stitches cast on.

Hold the needles point upwards and parallel again, and take the working yarn round the outside of the right needle to the back, before bringing it to the

front between the needles. Repeat the process of levering the right needle while keeping your right thumb on the working yarn, so you can pick up the middle loop from beneath and then twist it on to the left needle, making a new stitch. Continue this process until you have cast on 30 stitches.

As you create new stitches, you will probably need to shuffle them down the needle a bit to make room for the newer ones. Lay

the left needle horizontally on the table and, with all 30 stitches created, slide them evenly along the left needle, making sure none of them cross over each other.

// As you create new stitches, you will probably need to shuffle them down the needle a bit //

TIME FOR A ROW

To start the first 'proper' knitting row, hold the needle with the stitches in your left hand and put the tip of your right needle in the top loop from below. With the top loop over both needles, take the yarn round the outside of the right needle as before and bring it through the middle to the front. Move the →

QUICK TIP

Knitted items are often made from stocking stitch – alternating rows of knit and purl stitches. We've used garter stitch: every stitch is a knit stitch.

Knit a plant pot hanger

TUTORIAL



Right

Use a blunt-ended needle to sew the edges of the knitted fabric together to form a base

QUICK TIP

A chunky yarn cake is ideal for making two-tone designs with no extra effort from you.



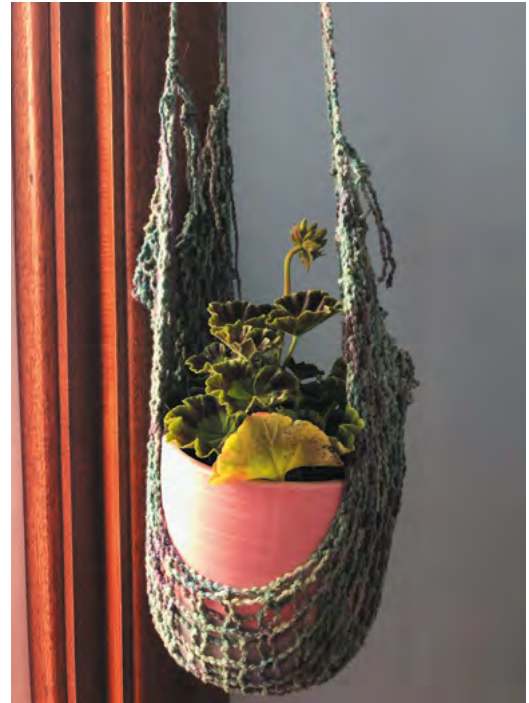
left needle up and over to tip the stitch from that needle to the right needle. Return the left needle to the upright position and take the yarn round the outside to the back of the needles once more.

Repeat this process until all the stitches that were on the left needle are on the right needle. You've just knitted row 1.

SLIDE RULE

Slide the stitches down the needle so that they sit evenly before switching hands so that the new row of stitches is on the needle in your left hand, ready to be knitted. Try not to pull the yarn tight, as it will make the stitches harder to manipulate. The garter stitch pattern we're creating is deliberately very stretchy.

At the end of a row, give the stitches furthest from the needle a tug and you'll see the knitted fabric begin to take shape. Do this at the end of each row to keep things neat and your knitting even.



HAMMOCK-STYLE

Our design covers a plain pot or jar. To instead show off your plant pot, knit a garter stitch strip of fabric and make four i-cord or plaited straps. Attach them to each corner of the fabric so they act as a hammock for your container.

Knit 16 rows in all, then take the glass jar or ceramic pot holder and, with your knitting needle upright, gently pull the knitting around it so that the knitted fabric reaches all the way around.

CAREFUL CAST OFF

Now we need to cast off the knitting from the needles (also known as binding off) such that it won't unravel. Knit the first two stitches of the next row, then carefully insert the left needle downwards into the first stitch and pull it up and over the second stitch you knitted. Be careful not to lose the second stitch in the process.

Holding the right needle so that it points slightly away from the left needle as you hook the first stitch over the second can help stop you losing both stitches at once.

Continue knitting a stitch and then casting off the preceding one until you have only one stitch left on the needle. Lay the needle down, snip the



CREATE A 1970S VIBE

You can create a fake macramé effect with chunky yarn by adding a spacer row between your rows of knitting. Wrap the yarn twice round the needle before knitting it. Drop the extra wrap off the end of the needle when you encounter it when knitting each stitch on the next row. Here, we've used super chunky acrylic yarn (recommended needles 10mm) on 12mm needles. Wrap the yarn round the needle more times to create a bigger gap between rows.

yarn about 15cm from the end of the needle, and then pass the end of the yarn through the remaining stitch to secure it. Do not pull it tight.

Stretch out your knitted fabric and lay the jar or plant pot on it. Use one end of the yarn to sew the cast on and cast off edges together, and the other end of the yarn to make a seam across the bottom of the jar or pot.

A STRAPPING YARN

The acrylic yarn we've used has a lot of 'give'. Knitted straps made from i-cord will provide a much stronger, but still stretchy, hanging mechanism. To make i-cord, cast on five stitches on the slightly smaller double-pointed needles. (If you don't want to buy more needles, you could use a sharp knife to whittle down both ends of a stick or pencil.) Knit

one row. Pull the stitches quite tight and make sure none of the stitches are twisted (it's very easy for one stitch to slide over the other). Gently slide the stitches down the needle so that they are about 1 cm above the start of where it begins to narrow towards the point.

Holding the two points of the needle between thumb and forefinger, turn your knitting 180 degrees anti-clockwise. Your working yarn – the bit attached to the yarn ball – will now be at the top, while your waste yarn or 'tail' – the bit left loose when you cast on – is at the bottom. The next stitch you make will close the loop of yarn so that your i-cord becomes a 3D tube.

Knit the next row, ensuring you bring the working yarn round the needle, not the tail yarn. Continue knitting five-stitch rows, switching ends of the needle at the end of each row, until the i-cord is roughly 30cm long. Remember, it will stretch with the weight of the plant and its pot. Cast off, leaving 15cm of spare yarn at the end. Make two more i-cords of the same length.

GOING LOOPY

With the vertical seam of your knitting facing away from you, work out roughly three equally spaced points where you will attach the straps, and tuck each strap in between the fabric and the jar/pot. Use the yarn ends from the i-cord to sew backstitch into the first row of knitted fabric and neatly tie the ends.

Now take the three i-cord straps, and tie them together in an overhand loop, wrapping the excess yarn tightly round it before tying it off and snipping off the ends. If you wish, you could loop the three strands through a metal curtain ring, or even fold and stitch them on to the metal ring, rather than simply making a loop. □

QUICK TIP

The pot holder we've knitted here fits a Mason jar, or a cleaned-out gherkin jar. The holey fabric we've created also works well if you prefer to use your jar as a night light holder.



Below

When making i-cord, you knit alternate ends of double-pointed needles. This makes the working yarn coil the knitting into a tube



QUICK TIP

Because it's hollow, you can thread wire, pipe cleaners, or even LED light strips through i-cord.

Radio-controlled Arduino for motor control

Let's learn how we can convert off-the-shelf radio control signals to be read by an Arduino



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!

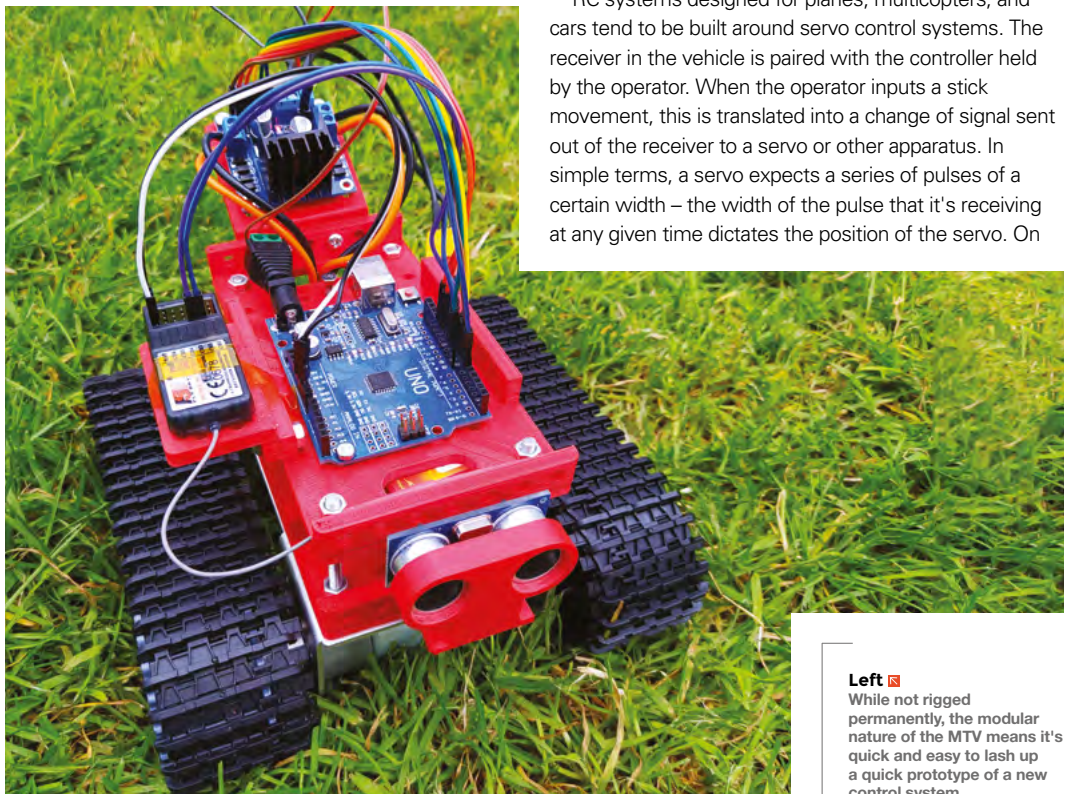
In issue 32, we wrote about designing and building the open-source modular tracked vehicle (MTV) project, and we set it up using a micro:bit as a radio control (RC) system, building on our knowledge from other tutorials.

The beauty of a modular open system is that you can change everything, including the control system – you can then make modular mounts for whatever you decide.

We wanted to play around with another control system for the MTV project, preferably one that had

some speed control, rather than the simple on-off configuration from last month, and also with an increased range over the micro:bit radio – not that the range on those is particularly bad for a small rover. In part, due to the lack of brushless speed controllers in our parts bin, we used an RC system, an older FLYSKY FS-T6 with an FS-R6B receiver, coupled with an Arduino. We aimed to convert the signals received from the RC system, primarily designed to control servos, to control the same L298N motor controller we already had mounted on the MTV robot.

RC systems designed for planes, multicopters, and cars tend to be built around servo control systems. The receiver in the vehicle is paired with the controller held by the operator. When the operator inputs a stick movement, this is translated into a change of signal sent out of the receiver to a servo or other apparatus. In simple terms, a servo expects a series of pulses of a certain width – the width of the pulse that it's receiving at any given time dictates the position of the servo. On



Left While not rigged permanently, the modular nature of the MTV means it's quick and easy to lash up a quick prototype of a new control system

YOU'LL NEED

- ◆ **An Arduino** (we used an Uno)
- ◆ **L298N motor controller**
- ◆ **Radio control transmitter and receiver** (we used a FLYSKY FS-T6)
- ◆ **Jumper wires**
- ◆ **DC motor**

GET THE CODE

You can get the code that we've described in this tutorial from hsmag.cc/issue33. It's also a really good idea to read up on how the `pulseIn` and the `map` functions work – we would recommend looking them up on the excellent Arduino reference pages. Within our sketch, we have commented to describe the pin connections between the Arduino, the L298N motor controller, and the RC receiver.



Left ♦ The FLYSKY FS-T6 is a budget radio control system that is primarily designed for flying models, but is equally at home controlling robots

most systems, including ours, each channel of the RC system can create pulses between 1 thousandth and 2 thousandths of a second – more commonly called 1 or 2 milliseconds (ms). With the Arduino, they're measured in microseconds, where 1000 microseconds equals 1 ms. Most RC systems are set such that when the stick is in the mid-position, the corresponding channel will be filled with pulses lasting 1.5ms, and this midpoint is considered the neutral position. A servo attached to this channel receiving pulses that are 1.5ms in length should sit in the centre position and not move at all. Moving the stick one way, say to the left, will reduce the pulse width – at the end of the stick travel, the pulse would be 1ms long. If it's a servo with 180 degrees of total travel, the servo should rotate 90 degrees to its limit of travel in the corresponding direction. Similarly, moving the stick fully to the right should expand the pulse width up to 2ms, and then the servo would move over to the 90-degree position at the opposite end of its travel.

To use an RC system with the L298N motor driver that we have in the MTV robot, we needed to capture the current pulse width values of the RC system pulses and scale those values to a value between 0 and 255. This, in turn, can be written to one of the Arduino pulse width modulation (PWM) pins as an `analogueWrite` function.

LET'S TAKE ITS PULSE!

We used the `pulseIn` function to read the output from a channel pin on the RC receiver, which was connected to

pin 10 on the Arduino. This returns the duration of pulses in microseconds and creates a variable called `RC2` (because that was the number of the corresponding channel pin we were connected to on the RC receiver), so `RC2` should always be a value between 1000 and 2000 microseconds. We then used the `map` function to map this range of values to our required range of 0–255.

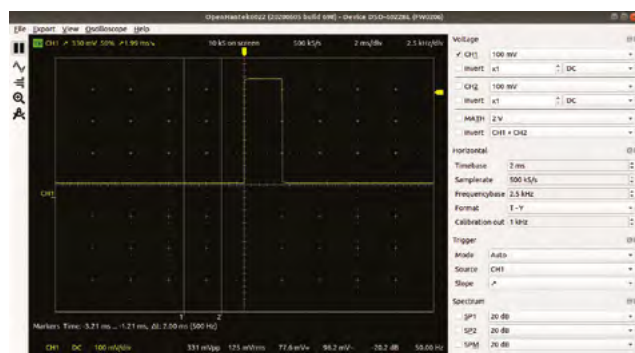
If the EN1 pin of the L298N pin has the jumper on, this connects it to the supply voltage – it's the equivalent of sending it a 'full steam ahead' message, or a '255' value in our case. We have removed this jumper pin, therefore connecting and sending our analogue value between 0 and 255 acts as a speed control for the connected motor. The other two pins per motor on the L298N are used to

RC systems designed for planes, multicopters, and cars tend to be built around servo control systems

set the direction of rotation, so sending one pin high and the other low from a microcontroller sets a direction of motion, reversing those values swaps the direction. So, to test our code, we wired motor channels IN1 and IN2 to pins 8 and 7 respectively, and wired a PWM-capable pin, pin 5 on the Arduino to the EN1 pin on the L298N. To test our emergent system we used the `digitalWrite` function to write pin 8 to LOW and pin 7 to HIGH in a sketch, and an `analogueWrite` sending our mapped value between 0 and 255 to EN1 via pin 5.

Connecting the motor and battery, we can now control the speed of the motor from zero to full power using our RC transmitter. Repeating this for the other motor and the other track gives us a rudimentary one-direction-only, skid-steering system for our MTV robot. Next step is to investigate mapping the receiver channels so that we can detect the midpoints of the signal from the controller, and do forwards and reverse for each motor for true tank-style control. □

Left ♦ Using a cheap USB oscilloscope, we can visualise a pulse from one of the RC channels – this is with the controller pushed fully one way, making the pulse the full 2ms in length



TU



Furby connected companion

Give an old toy a new lease of life by taking control over Bluetooth and adding a WiFi connection



Rob Miles

robmiles.com

Rob Miles has been playing with hardware and software since almost before there was hardware and software.

Do you fancy making something that can tell you the weather forecast using 'fart' noises? Or an alarm clock with a difference? You can do all this and more if you start with an old Furby Connect toy and add a bit of IoT magic. By hooking into the Bluetooth connection to the Furby, you can add skills and use it as a novel peripheral. By adding WiFi, you can make it truly connected. In this article, we'll discover how to use Bluetooth BLE to create a connected toy for all ages. Note that at no point will your Furby be harmed during the project, although you might want to make a power connector for it to cut down on battery costs.

The first Furby toy was released in 1998 and has been through several incarnations. The latest, Furby Connect, was released in 2015. It connects via Bluetooth BLE to an Android or iPhone app that lets you 'feed' it and go on adventures together. The Bluetooth messages are not encrypted, and the Furby was famous for a while as an example of how easy it can be for hackers to reverse-engineer and commandeer consumer devices. We're indebted to Florian Euchner, whose GitHub page at hsmag.cc/ZoqCwl told us all we needed to know about controlling a Furby over Bluetooth BLE.

Our programs will trigger preset behaviours built into the Furby. There are many behaviours, with a substantial number devoted to different 'fart' sound

effects, which the eight-year-old in us rather likes. Furby also has a multicoloured LED and a bunch of touch sensors, plus a joystick we can use.

We are going to control Furby with an ESP32 device programmed in C++ using the Arduino environment. We are going to turn Furby into a Weather Reporter using the weather service provided by OpenWeatherMap (openweathermap.org). Tweaking the antenna on Furby's head will get either the temperature or the current conditions. If the weather is raining, Furby will make a characteristically unhappy sound. You could use this code to make your Furby report on anything. You could make an alarm clock or add an MQTT connection to allow you to control Furby from anywhere in the world.

GET TO THE CODE

The Furby code is on GitHub at hsmag.cc/k9HHgq. To talk to a Furby from your Arduino programs, you just need to drop the **Furby.h** and **Furby.cpp** files into the same folder as your Furby sketch and then include the Furby libraries in your Arduino program:

```
#include "Furby.h"
```

Next, you create a Furby object and interact with it. The Furby object handles all the BLE messages between the Furby and the ESP32:

```
Furby* myFurby;
myFurby = new Furby();

while (!myFurby ->connect()) {
  Serial.print('.');
  delay(500);
}
```

The code above creates a new Furby object called **myFurby** and then tries to connect using it. The

FURBY SPIES?

The first Furbies contained cunning timed-release software that made them appear to learn from things they heard as they 'grew up'. The learning behaviour was so convincing that the US National Security Agency banned Furbies, in case they learned something they shouldn't.

connect() method returns true when it finds and connects to a Furby over Bluetooth BLE. The code above tries repeatedly to connect and prints a "." each time it fails.

AT YOUR COMMAND

Once we have a connection to the Furby, we can send messages to it. A message is a block of 8-bit values. The **Furby** class has a **sendMessageToFurby** method. You give this method a block of data and its length, and the message is sent to the Furby. The code below sends a six-byte command to the Furby to make it say "Connected".

```
unsigned char message[] = { 0x13, 0x00, 34, 0x00,
                             0x00, 0x00 };
myFurby ->sendMessageToFurby(message, 6);
```

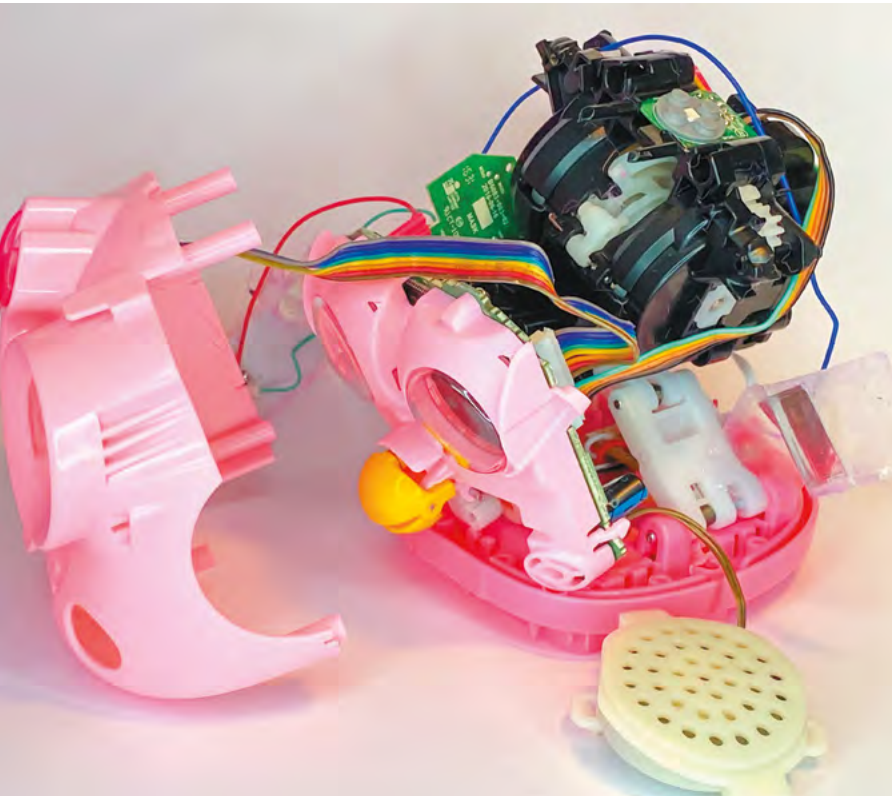
The first byte of a message is the type of command. The above message uses command number 0x13 (hex) which triggers an action. The action with the number 34 (a decimal number) makes Furby say "Connected". The reason why we've specified the action number as decimal is that the action list compiled by Florian Euchner at hsmag.cc/whtgSj gives command values in decimal. You can have a lot of fun working through the messages in the action list and finding out what the Furby does for each of them. →

YOU'LL NEED

- ◆ **A Furby Connect toy.** You can find these on eBay for around £20. Make sure you get one of the most recent models (after 2012) which has the LCD eyes. These are fun devices for kids of all ages.
- ◆ **An ESP32-based device for the BLE connection.** We used the M5StickC from M5Stack, which has a handy LCD panel. Make sure that you get an ESP32 with at least 4MB of flash memory so that you can run programs that use Bluetooth and WiFi at the same time.



- ◆ **Left** The Furby Connect that you want will have a Bluetooth tag on its bottom. Make sure that yours has one too



Above ♦ All of the Furby movements are powered by a single motor and a system of cams that animate the different elements. The cams are in the large gearbox at the top right of the figure. The Furby uses a rotary encoder to detect how far the cams have turned

QUICK TIP

The Furby will not shut down when it has an open Bluetooth connection. This can be rather hard on its batteries. However, the Furby will run quite happily on rechargeables, and you can also power it from a 5V adapter which you can wire to the battery terminals.

To stay connected to a Furby, you have to 'tickle' it by sending regular commands. You can do this in the 'loop' part of your Arduino program by calling the `loop` method on the Furby object.

```
myFurby ->loop();
```

Furby will not shut down (go to sleep) when there is a BLE connection active. If you want to let Furby go to sleep, your program can disconnect 'properly' by calling the `disconnect` method:

```
myFurby-> disconnect();
```

If you allow a Furby to sleep, your program will be unable to connect until the Furby is woken up again by moving its antenna. Your program can use the `connected` method to find out if the Furby is still connected:

```
if(!myFurby ->connected())
{
    Serial.println("Furby disconnected");
}
```

LISTENING TO FURBY

The Furby has four touch sensors around its body, and a four-way joystick in its antenna. If any of these inputs changes, the Furby sends a sensor status message. We receive the messages using a function that accepts a buffer pointer and a length value. The

function `messageDump` below displays the data received from the Furby.

```
void messageDump(unsigned char* buffer, unsigned
int length){
    Serial.print("Message from Furby: ");
    for (int i = 0; i < length; i++) {
        Serial.print(buffer[i], HEX);
        Serial.print(" ");
    }
    Serial.println();
}
```

We can connect the `messageDump` function to the Furby by using the `bindToIncomingFurbyMessages` function, like so:

```
myFurby
->bindToIncomingFurbyMessages(messageDump);
```

Now, whenever the Furby sends a Bluetooth BLE message, the contents will be delivered via a call of `messageDump` which will print the message contents.

UNDERSTANDING INPUTS

A message from the Furby that starts with the value 0x21 contains sensor data. These messages are sent every time a reading changes. The messages contain 8 bytes. We've decoded the parts of the message relating to the antenna joystick and the tail switch. For example, bit 6 of the third byte in a message starting with 0x21 indicates that the antenna has been pushed towards the front of the Furby.

```
if (buffer[2] & 0x40) {
    // Antenna front
    wantWeather = true;
}
```

Note that C++ numbers items start at zero, so `buffer[2]` is the third byte in the message. This code sets the `wantWeather` flag to `true` if the antenna is pushed towards the front of Furby. This is how we trigger Furby to produce a weather report. The `loop` function in the Arduino application checks this flag and calls a function to make a weather report if it has been set. It then resets the flag:

```
if (wantWeather)
{
    sayWeather();
    wantWeather = false;
}
```

The `sayWeather` function gets the weather from OpenWeather and then makes Furby do something

appropriate. We've wrapped the weather code into a simple class containing an **update** method that contacts the OpenWeather service and gets the latest values. The **update** method accepts a timeout value which is set to five seconds. This code updates the weather and then gets a copy of the weather description message. If the message is 'rain', the Furby will play a suitable sound.

```

if (weatherClient->update(WEATHER_TIMEOUT_SECS))
{
    char weatherDescription[WEATHER_DESC_LENGTH];
    weatherClient->getWeatherDescription(weatherDescription);
    if (strcasecmp(weatherDescription, "rain") == 0){
        unsigned char message[] = { 0x13, 0x00, 7, 1, 0, 0 };
        myFurby->sendMessageToFurby(message, 6);
    }
}

```

There is a similar mechanism that makes the Furby deliver temperature information when the antenna is pushed towards Furby's back.

USING OPENWEATHERMAP

To use OpenWeatherMap in your application, you will need to register for free at openweathermap.org and get an API key to authenticate your weather requests. You give this key along with your location when you create the weather service. The WeatherClient code is provided in the form of the files **WeatherClient.h** and **WeatherClient.cpp** that you copy into your application folder:

```

#include "WeatherClient.h"
WeatherClient* weatherClient;
while (!weatherClient->connect("Hull", "uk", "--API key goes here--", 5))
{
    Serial.print(".");
    delay(500);
}

```

The code above gets the weather for Hull – you can change this to any other location should you wish. The WeatherClient uses the WiFi connection on your ESP32, and yes, you've guessed it, we've made a little class for that too. You can find this in the files **WiFiClient.h** and **WiFiClient.cpp**. You use it in the same way as you use the WeatherClient and the Furby classes:

```

#include "WiFiConnection.h"

```



```

WiFiConnection = new WiFiConnection();

while (!WiFiConnection->connect("Your WiFi SSID", "Your password", 10))
{
    Serial.print(".");
    delay(500);
}

```

Above Find this setting in the 'Tools' menu of the Arduino application. A program that uses both the WiFi and Bluetooth libraries might be too large to fit into the standard memory configuration for ESP32 applications. Use the Tools menu in the Arduino application to turn off 'Over The Air' (OTA) updates, and allow a large application (APP) to be loaded

The WeatherFurby application in the sample code makes a Furby weather reporter. It's a great starting point for your Furby fun and games. You can make your Furby respond to inputs on the ESP32, or trigger it using the real-time clock. You can even get the Furby to stream real-time values from its accelerometer into your program. The samples also contain an MQTTFurby application that shows how a Furby can be connected to an MQTT broker to send and receive remote commands.

Below The Furby contains quite a lot of processing power. The two connectors at the base of the picture are for the LCD eyes. The round copper spiral at the bottom is an electromagnet used to animate the mouth movements



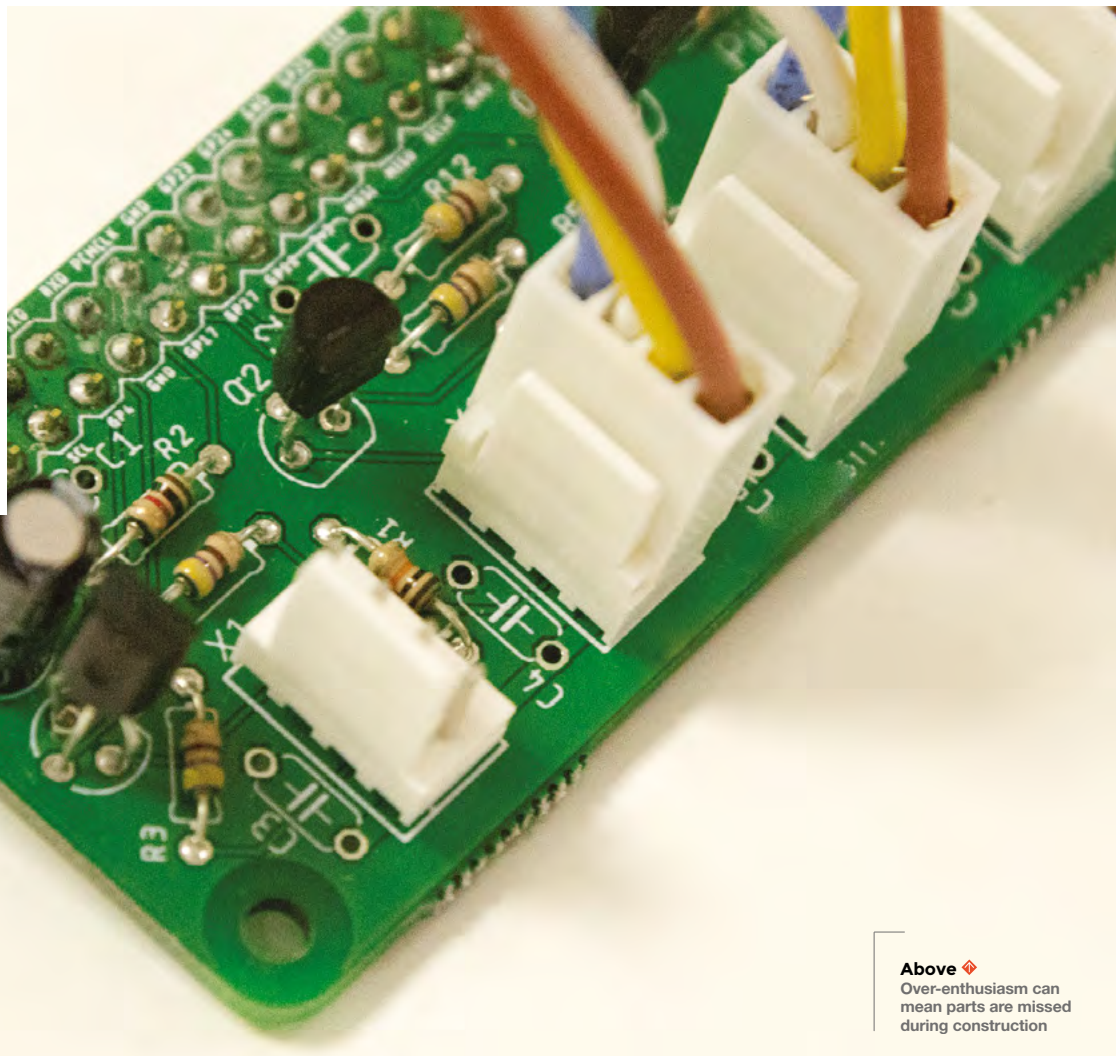
Debug techniques with multimeters and oscilloscopes

Spend less time debugging with these tips to success

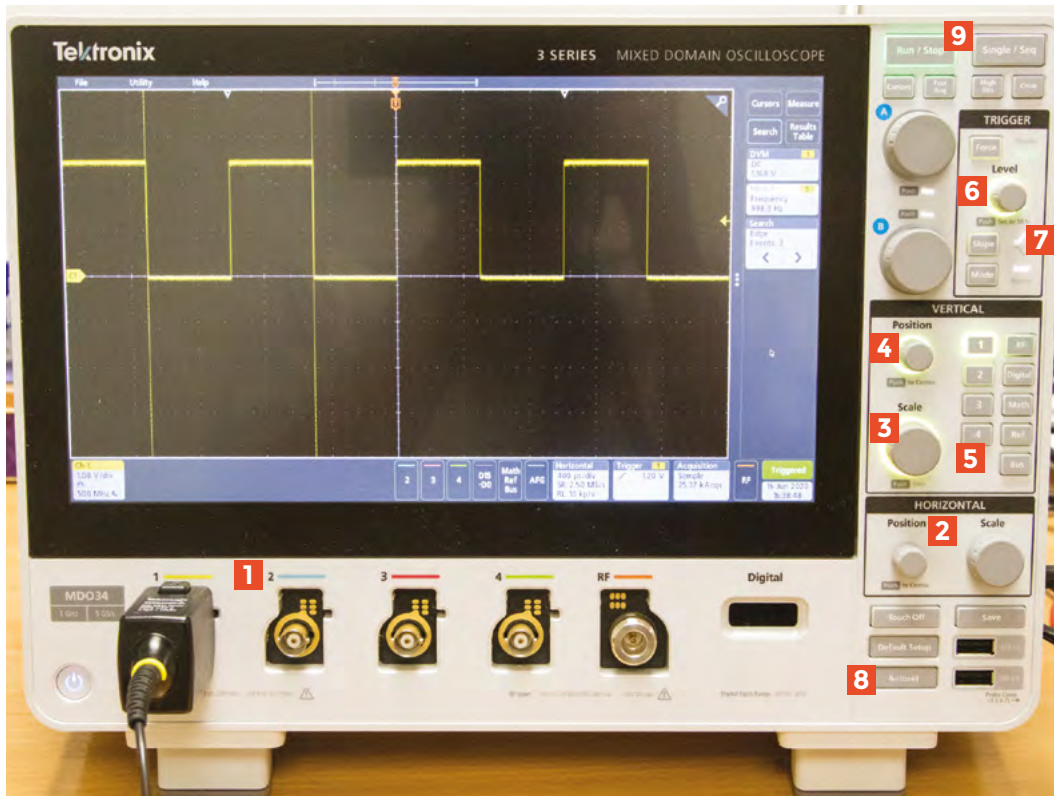


Dr Andrew Robinson

Dr Andrew Robinson is a part-time university lecturer and runs his own design and manufacture electronics consultancy in Manchester. He's responsible for creating CodeBug, PiFace, and the Quizit.net interactive quiz site.



Above ♦
Over-enthusiasm can mean parts are missed during construction



- 1 Signal inputs
- 2 Time base – sets the horizontal scale
- 3 Gain – sets the vertical scale
- 4 Vertical offset
- 5 Channel selection
- 6 Trigger level
- 7 Trigger edge (rising, falling, or both)
- 8 Autoset button
- 9 Continuous/single shot mode

Modern software and hardware projects are complicated beasts with thousands or even millions of places for bugs to hide. In this series of articles, we'll look at the tools, tips, and techniques to find them quickly and stop them appearing in the first place.

There are no fixed rules for how to debug a project; it's part art, part methodical checklist. It's about coming up with hypotheses, testing them and deducing what's going on. You can make the debugging process less painful if you consider it at the design stage of a project.

Our two golden rules for designing for easy debugging are:

1. Make it easy to observe behaviour, and
2. Make it easy to control behaviour.

You can only find a bug if you can see what's going on, but you also need to be able to control the conditions to make the bug pop up.

Make your bug pop up on demand. A key technique is to narrow down your search space. Split your project up into logical blocks, and test each one separately so that you know which ones work and which ones need deeper fault finding. There are no hard and fast rules on where to split it; you might

start with software-hardware split, or you might start by isolating particular parts of the circuit, such as the power supply or software functions. Try to divide it into large subsections at first, and then further subdivide so you can quickly eliminate large chunks of your design. Of course, this is only possible if you have followed the two golden rules and are able to inject a stimulus to exercise a section and observe its output to confirm correct operation!

Divide and subdivide for success. Do you really need to design for debugging, and do you need all this expensive test equipment? Well, it's really about saving time: you might be lucky and your project works perfectly first time, or you may be able to fix →

NOISE EXPLAINED

Electrical noise is the unwanted 'extra' varying voltages that combine with signals in a circuit. It may come from external magnetic or electric fields, the random motion of all subatomic particles (called thermal noise), or from events that happen elsewhere in the circuit. Too much noise can cause digital devices to measure incorrect values or malfunction. Noise can be reduced with filters composed of capacitors, inductors and resistors, or through re-routing or shielding wires to make them less susceptible. Having 100 nF capacitors across the supply pins of chips is a great help in preventing noise that would cause voltage levels to drop and reset the chip. Shortening long wires reduces induced voltages that might otherwise corrupt communications between circuits. Noise cannot be completely eliminated, but as long as it is managed to be below a certain level, a circuit will continue to operate correctly.

QUICK TIP

If your circuit has hazardous voltages (either in normal operation, e.g. mains, or in the case of faults, e.g. incoming cables struck by lightning), you want your test equipment to protect you. Make sure it and any probes meet the necessary standards, are in good condition, and are qualified for where you're measuring. Look for the EN 61010 standard and check the CAT rating that defines what the equipment can be used on.

it with trial and error, but the question is how quickly and easily do you want to fault-find.

In this article we look at spotting bugs in an analogue electronic environment. And before you say you only design digital circuits, it's important to realise all circuits are analogue circuits! If the voltage on an input isn't quite high enough when the microcontroller reads it, it isn't going to read it as being on. Signals in the real world don't change instantaneously, and they can pick up 'extra' voltage from stray radio waves or be affected by fluctuations in the voltage of the power supply. The act of signals transitioning can cause this noise in the power supply, or it could be from the switching of power-hungry devices such as motors, relays, or speakers.

As makers, our projects are getting more sophisticated with performance being pushed, be it either for power efficiency or faster computations, or complex software. Margins are getting tighter, and stray electrical noise becomes more significant. An oscilloscope showing exactly what a signal looks like has saved us days of trial-and-error debugging.

FAULT-FINDING AND BRINGING A NEW CIRCUIT UP

Making and building a project is exciting, but sometimes over-enthusiasm can lead to disappointment, and there are some simple good habits that can save tears later. Before powering up a circuit for the first time, it's good to get into the routine of visually inspecting it. Are the wires connected correctly? This is especially important if they're polarised, like the power supply. Are all the components populated the correct way round in the correct place? Are there any shorts, solder blobs, or

OSCILLOSCOPE GROUND

Without taking care, it's possible to damage your circuit, yourself, or your scope. Always consider what else is connected to a point in your circuit before you connect the oscilloscope ground clip.

The ground, or chassis, on many oscilloscopes is connected to the mains earth, and thus whatever you clip the probe ground to in your circuit will also be connected to the mains earth. If your circuit is powered by a non-isolated power supply or your computer's USB port, then the ground of your power supply will likely also be at mains earth potential. As such, connecting your scope ground to a point in your circuit is the same as connecting it to the ground of your power supply, i.e. equivalent to short-circuiting your power supply.

whiskers of wire? A quick double-check can save the expense and pain of replacing a component after it's been powered on and its gubbins have disappeared in a plume of white smoke.

If you've got a lab power supply and you can set a limit on the current, set it only just above what your circuit needs as this will reduce the damage that can occur. If you don't know what to set it to, do a rough sum of the figures on datasheets, start low and increase if necessary, all the time checking that no component is getting too warm.

Including a serial port connection to your processor or even just having a few outputs is useful for allowing the software to report its status. If you don't see any output, or the software isn't responding to the inputs, then it's time to get measuring.

With the circuit powered up, check the voltage at key points in the circuit. With the aid of the

circuit schematic and multimeter, check the correct voltage is present on all chips' supply pins. Next, use a multimeter or your lab power supply to check if a sensible amount of current is flowing: too low suggests a broken wire or parts of the circuit not turning on; too high suggests a short circuit. A logging multimeter or logging power supply can really shine now as it'll show you how the power

TYPES OF ANALYSER

SPECTRUM ANALYSERS

Rather than drawing a waveform with time horizontally, spectrum analysers plot increasing frequencies left to right and intensity of the different frequencies on the vertical axis. They effectively sweep a range of frequencies and show the strength of each frequency. Unless you're fault-finding the reception or transmission of radio or checking for electromagnetic interference, they're expensive and are less versatile than an oscilloscope. You could consider a much lower-cost, software-defined radio, such as the LimeSDR Mini, or repurpose a very cheap USB digital TV tuner with free Spektrum software: hsmag.cc/uPFoue.

VECTOR NETWORK ANALYSERS

These are basically spectrum analysers with a built-in signal generator. They can show how a signal is attenuated or reflected at different frequencies. Like spectrum analysers, unless you're troubleshooting an antenna or radio circuit, you're unlikely to need one. Again, some software-defined radios can be used for the purpose, or low-cost (\$60) antenna analysers are available online.



Above ♦
Adjust probe compensation to achieve maximum accuracy

requirements change over time and from this you can get an idea of the activity in the circuit and even the activity of a microcontroller.

Although the multimeter shows that apparently the correct voltage is present, only an oscilloscope can show if the voltage changes over time. To show if the power supply dips under load, set the scope

“ The latest multimeters have connectivity options, such as USB and Bluetooth ”

to trigger on a falling edge, slightly less than the full voltage. A scope will also show if there's too much electrical noise on the power supply.

ELECTRICAL TEST GEAR

Since we can't see electricity, you'll need something to make it visible. This could be as simple as an LED or a multimeter, but if you need to see how the signals change over time, then an oscilloscope, signal analyser, or spectrum analyser might be more suitable. Here are some tips for buying and using test gear.

MULTIMETERS

A good multimeter is a great starting point and, since you can pick them up for just a few pounds these days, there should be one in every maker's toolbox. They're typically able to measure voltage,

current, and resistance, with some also measuring capacitance and frequency. Most multimeters make an instantaneous measurement; if a signal is changing, you'll either see the numbers on the display jumping around, or it may show some sort of average value.

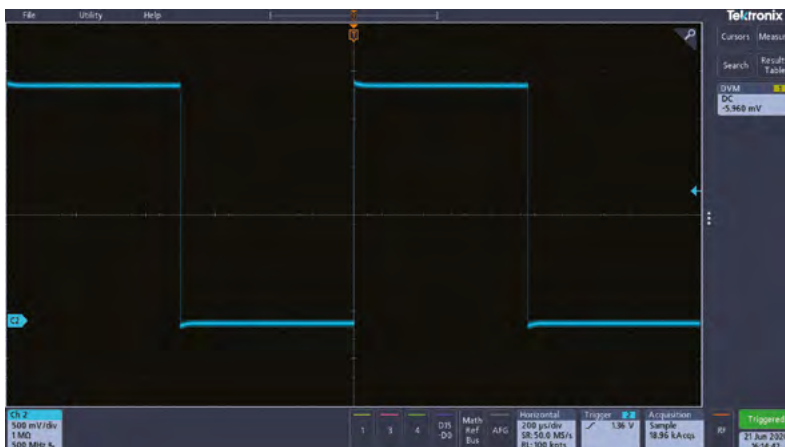
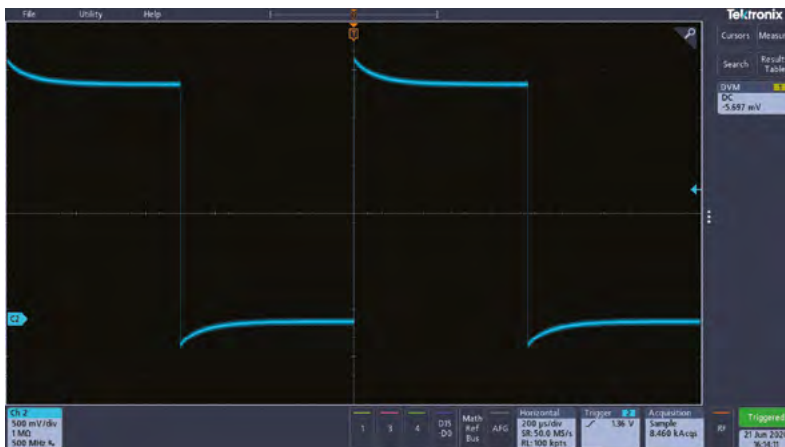
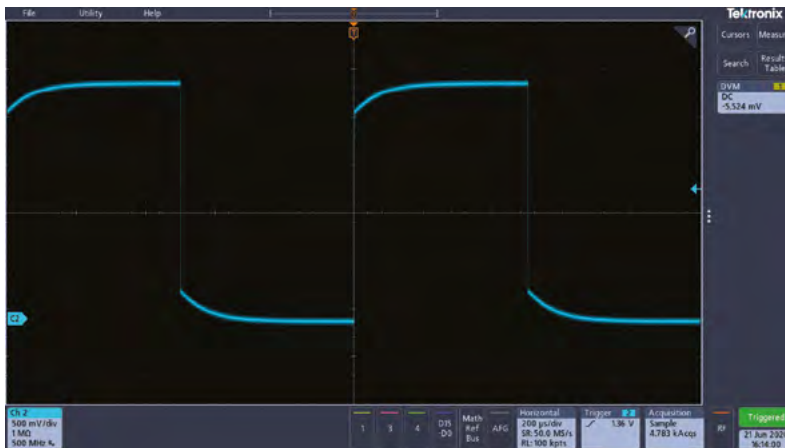
When selecting a meter, look for accuracy, measurement range, measurement functions, and that it meets necessary safety standards. Consider ease of use too: are the controls/sockets intuitive? Is it clear which mode it's operating in? Is the display easy to read? Does it have auto power shut-off to save the battery when you forget to turn it off? Bench multimeters offer the most features and have very high accuracy, but aren't particularly portable.

The latest multimeters have connectivity options, such as USB and Bluetooth, that don't add a significant price premium. These allow a series of readings to be logged, and are great for using with a computer to plot a graph to show how readings change over time. For example, you could see how a battery discharges over time. For meters with a high enough sample rate, you can record power consumption by determining the time spent and current taken in sleep and active modes. However, even the high-end logging multimeters have limited sample rates compared with oscilloscopes. →

COMPENSATING THE OSCILLOSCOPE PROBES

For accurate measurements, passive oscilloscope probes need to be 'tuned' to your scope; this is known as compensating your passive probes. Do this by connecting the tip of the probe to the reference signal and the ground clip to the reference ground, which are both usually near the front panel. With a small screwdriver, adjust the screw in the probe until the corners of the square wave are right angles.

TUTORIAL



Above ♦
Probe compensation: undercompensated, overcompensated, and just right

OSCILLOSCOPES

The history of oscilloscopes can be traced back to using electromagnetism to move a marker over a piece of paper on a rotating drum. In the late 1800s, when he wasn't studying microbes and urine with Louis Pasteur, Jules François Joubert needed to record changing voltages for his AC electricity experiments. Joubert's device was refined by adding a motor to rotate paper on a drum, and the result was the Hospitalier Ondograph, the first automatic means to capture an electrical waveform.

The motivation for developing these early devices was the same reason that we'd use an oscilloscope today: to show how a signal changes over time.

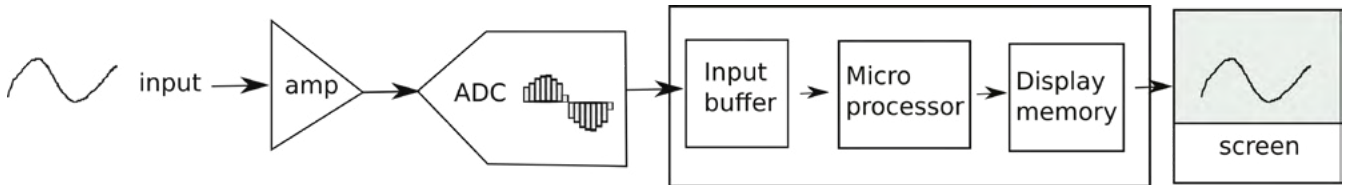
In 1930, the first oscilloscopes based on cathode-ray tubes began to appear. Electron guns fired a beam of electrons in the vacuum tube that caused phosphors on the inside of the screen to glow for a short period. Coils surrounding the tube were used to steer the electron beam. An internal timer circuit generated a periodic ramp that scanned across the screen, while the voltage of the signal under test caused the vertical deflection. Controls were used to set the amplification of the input, and control the rate of the horizontal scan.

With the development of computing and analogue-to-digital converters, the 1980s saw the development of digital oscilloscopes. These measure the input voltage at regular sample periods, and store in memory before displaying to the screen. For many years – and for some manufacturers, still to this day – the controls on digital scopes were copied from their analogue predecessors.

In recent times, in an era of touchscreens and swipe gestures, some manufacturers have redesigned their interfaces to be more intuitive. Most notable are the efforts of Tektronix, whose award-winning user interface was designed around the user rather than the legacy of 1930s hardware. The general-purpose nature of computing means that modern scopes can provide more features, with software to automate measurements, provide mathematical analysis, decode protocols, and unify the digital and analogue worlds.

USING A SCOPE

In some ways, scopes are very expensive Etch A Sketches. Scopes are essentially graph-drawing machines with knobs on! The main controls and principle of operation are the same across all scopes, regardless of price, model, or manufacturer. A scope draws the amplitude (usually the voltage it is measuring) of the signal vertically against time from



Above ♦ Processing pipeline for a digital oscilloscope

left to right. The rate of drawing the graph (i.e. the horizontal axis scale) is set by the time base control, and the amplification (vertical scale) is set by the gain control. Scopes with multiple channels draw a line for each input channel, each with separate gain, but all share the same time base. An offset can be applied to each channel to move its base position up and down the screen. Most modern scopes have an autoset button that usually makes a decent job of setting the gain, time base, and trigger.

A scope will repeatedly wait until it is triggered, draw the input signals for a period of time, and then wait for the trigger again. A trigger is commonly when a signal on one of the channels either rises above, or falls below (the edge direction), a settable voltage threshold (the trigger level). Some scopes can have more complicated triggers that may look for particular waveform shapes or even the evaluation of logical

expressions. Scopes also have a trigger delay that let you start drawing the waveform a set period of time before or after the trigger point. (Digital scopes have a buffer that stores the data pre-trigger to allow this apparent going back in time.)

It's important to realise that, in normal operation, a scope will repeatedly redraw the waveform. As such,

Some scopes can have more complicated triggers that may look for particular waveform shapes

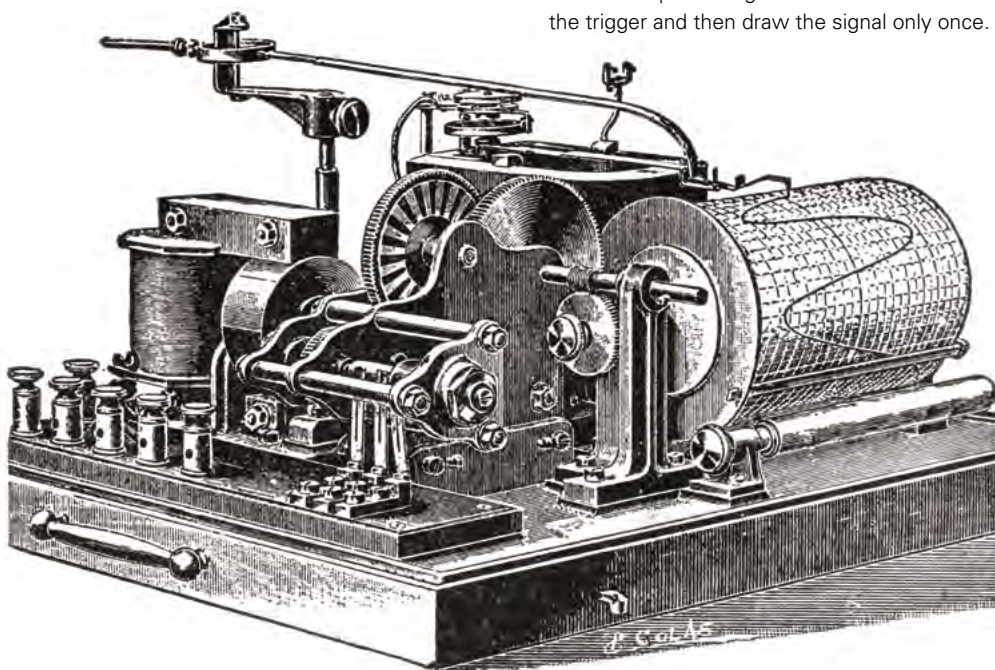
to get a constant display, the input signal must repeat, and the scope must trigger at the same point in the waveform. If not, the display won't 'lock on' to the signal, and it will appear to scroll around or flicker.

If the signal you want to capture isn't periodic, e.g. you want to capture a one-off pulse, then you should use the scope in 'single shot mode'. This will wait for the trigger and then draw the signal only once. □

QUICK TIP

Meters measure current by measuring the voltage drop from the current flowing through a fixed resistor. When measuring small currents, the resistors in some meters cause a voltage drop to be significant enough for your circuit to not receive enough voltage to operate or to affect the current that actually flows. Manufacturers will state this drop as the burden voltage.

Left ♦ The Hospitalier Ondograph, precursor to the oscilloscope and perhaps inspiration for a makers' challenge to recreate with Lego and a biro



Build useful structures out of Lego

Design and 3D-print a Lego-compatible wall mount so you can make a display shelf





Above  A well-organised Lego collection makes it easier to find parts

Lego is brilliant. For many of us, it was our gateway 'making' experience, leading us to explore woodworking, metalwork, electronics, and more.

However, there is no need to move on from Lego just because you now have other making skills – if you've got a box of Lego in your attic, why not incorporate it into your current projects?

There are many reasons to use Lego as part of a maker project. Often, it's the quickest way to prototype a certain design, especially if that design requires a complex mechanism. You may also be looking to do the opposite, and incorporate non-Lego elements (perhaps a Raspberry Pi) into a Lego project. In this tutorial, we'll be getting somewhat meta, and building a Lego model display shelf out of Lego, with 3D-printed, Lego-compatible wall mounts. Of course, you can use the techniques shown here to build anything you want – this project is just meant to give you a good grounding in how to design and build Lego-compatible things.

WHAT ARE WE MAKING?

The shelf itself will be made of Lego (mainly Lego Technic), and will be large enough to display a medium-sized Lego model, around 30 cm wide by 15 cm deep. The non-Lego parts which we are making will be the wall mounts – these will be 3D-printed (although you could certainly make them in other ways). We will make two identical mounts, allowing us to securely attach the shelf to the wall without harming any Lego pieces.

Before we start designing, we need to know how to make our mounts compatible with Lego pieces. There are a few ways that Lego pieces can be attached to each other: the iconic standard bricks have studs on the top which click tightly into corresponding spaces underneath, while Lego Technic pieces are often joined using pegs and →

WOOD OR METAL

Even if you don't have a 3D printer, this project is perfectly possible using more traditional tools. For example, you could mark a piece of wood or metal using a Lego Technic beam's holes as a template, then use a pillar drill to make nice straight holes in the material. Note that a standard Technic beam is 8 mm wide (and a standard Lego brick 16 mm wide), so if you can make your design out of something which is already some multiple of this measurement (e.g. 8 mm stripwood), it will make your life easier!









Matt Bradshaw

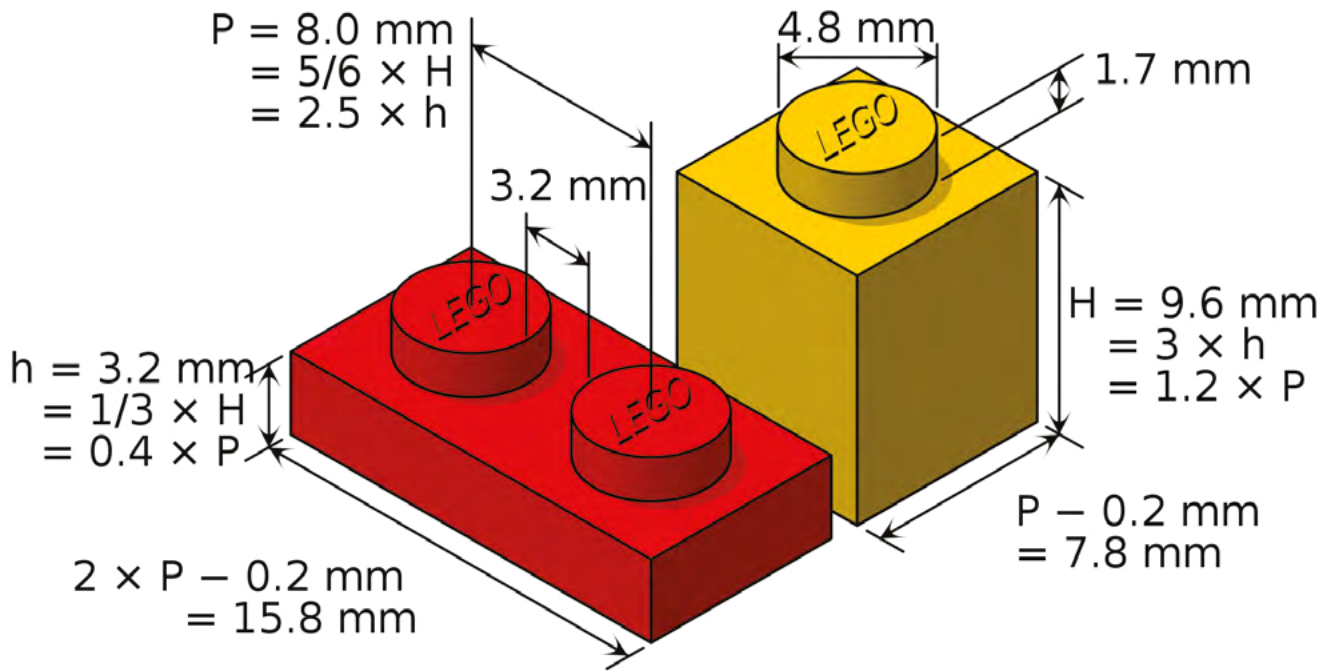
mattbradshawdesign.com

Matt Bradshaw is a programmer, maker, and musician from Oxford. He likes to build instruments to play with his band, Robot Swans. Find more of his projects at mattbradshawdesign.com

YOU'LL NEED

-  **Lego**
-  **3D printer**
-  **Screws and screwdriver**
-  **Drill**
-  **Spirit level**
-  **Wall plugs**

TUTORIAL



axles which go through holes. The sizing and positioning of these studs and holes are standard across all Lego pieces, and there is plenty of information on the web about this system. Essentially, the standard unit in these bricks is 1.6mm, and most things are some multiple of that.

I DON'T WANT NO STUDS

The studs on official Lego pieces are made with really precise tolerances (less than 0.01 mm!), which is why they fit together so nicely, but it does make it

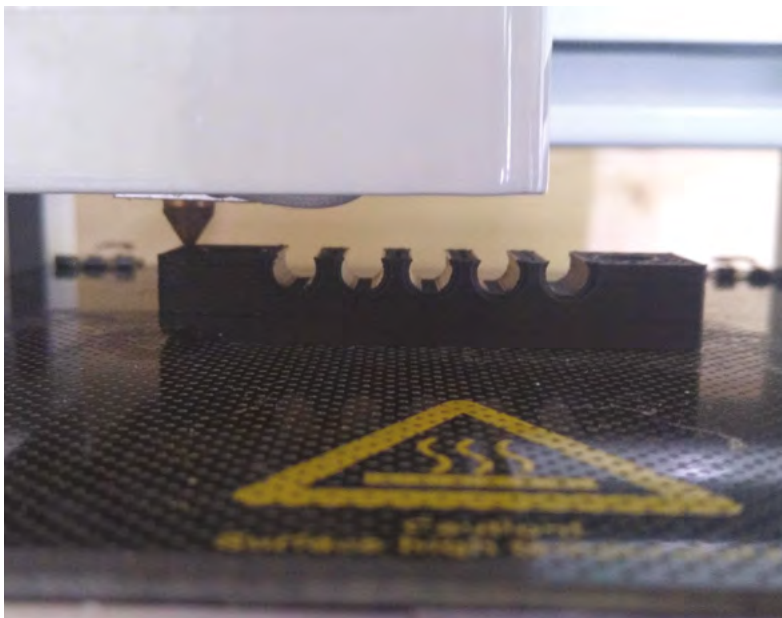
hard to make your own versions. Instead, we're going to use the Lego Technic peg/hole system, because it's a bit more forgiving (especially if you're not using CAD).

We're going to use FreeCAD to design our mounts because it's free and open-source – you can download it from freecadweb.org. FreeCAD consists of a number of different 'workbenches', which you can switch between using the drop-down menu at the top of the screen. We want to start in the 'Part' workbench, and we're going to begin by adding a 'cube solid', which will form the main body of our part. Next, set the length, width, and height of the solid. In brick terms, our part is going to start off

Above Lego has precise, well-defined specifications

Credit
CC BY Cmglee

Below You may need to try a couple of prints before you get the fit right



LEGO POWER

Besides making your own parts which attach directly to proper Lego bricks, there are various other ways to make something compatible with Lego, particularly Lego Technic. Rubber bands and string are often used in Technic models as miniature equivalents of real-life drive belts and steel cables. Using these items can allow a non-Lego motor to drive a Lego mechanism, providing a bridge from Arduino and Raspberry Pi circuits into the world of Lego.

You can even go one stage further and interact directly with Lego's electric system. There is plenty of information on the web about different Lego motors, such as their required voltage, current draw, etc. There is also a kit called BrickPi which connects a Raspberry Pi to Lego Mindstorms sensors and motors.

And if you're really ambitious, the inner diameter of Lego pneumatic tube is 1.5mm...



Right  Angled struts can add strength to your shelf

as the size of two classic bricks end to end: two studs wide (ten units), eight studs long (40 units) and six units deep. This works out to 64 mm by 16 mm by 9.6 mm.

PRACTICE MAKES PERFECT


If you haven't used FreeCAD before, you may want to experiment a bit before pushing on with this design – try and get the hang of zooming, panning, and rotating the camera. Then try creating, moving, and resizing a few different shapes. You will also want to learn about 'cuts' between different shapes,

which is where you subtract one shape from another – this is done by selecting two shapes, then clicking the Make a cut of two shapes button. The second shape you select will be removed from the first.

Use this subtraction technique to add holes to your part, using cylinder shapes with a diameter of three units. We are essentially creating a modified Lego Technic beam, where the centre of each hole's vertical position is 3.5 units from the bottom, or 2.5 units from the top. Horizontally, the holes are at multiples of five units from either end. If this is unclear, carefully measure a Lego Technic beam, noting that pretty much every measurement is a multiple (or occasionally a half-multiple) of the unit, 1.6 mm. →

AXLE HOLEY

Designing for Technic axles is really simple, because you're literally just making a hole for the axle to go through. If you also want your hole to be able to accommodate pegs (which are wider than axles in places), you will need the hole to be larger at the entrances – about 5.6 mm instead of 4.8 mm (3.5 Lego units instead of 3), to a depth of about 0.8 mm (0.5 units).

Left  If your hole is too small, use a 5mm drill bit to bore it out

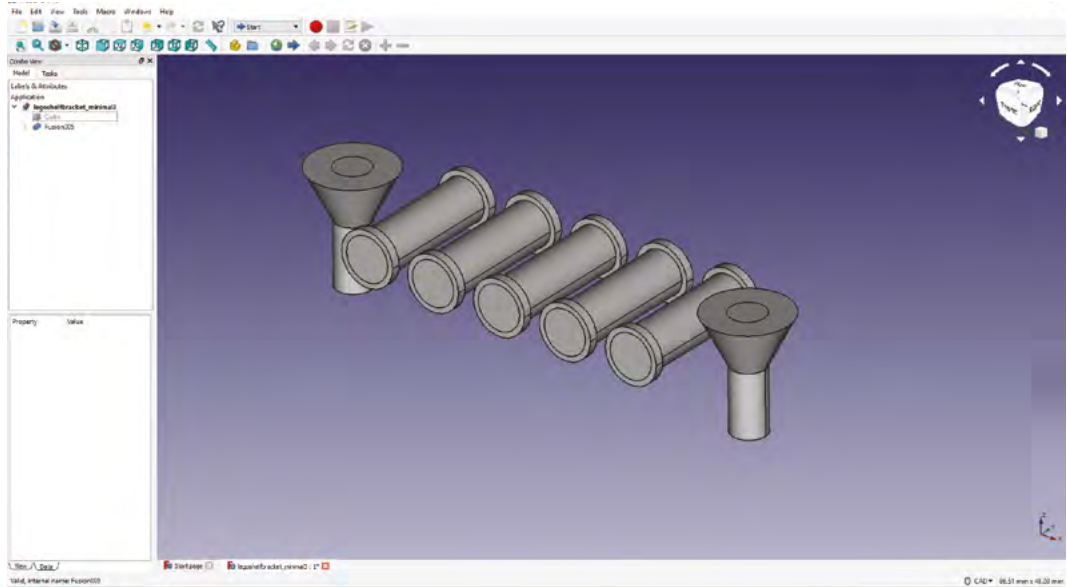
Below  The top of the mount should be flush with the surrounding Lego



TUTORIAL

QUICK TIP

If you ever need to buy one-off parts for a Lego project, bricklink.com is a good place to buy Lego in individual pieces rather than sets. It was originally an unofficial fan site, but is now owned by Lego.



Above

These shapes will be subtracted from a cuboid to create the final part

Right

Double pegs are useful for strong, rigid builds

Below

It only took 30 minutes to build this alternative design, where the helicopter model appears to be flying

Next, we want to add pilot holes for our screws (more cylinders), as well as countersinks for the screw heads, which are made using cone shapes. Measure your screws to determine the dimensions of the cone and cylinder objects. You may find it easier to create all the shapes you want to subtract, then fuse them together using the Union button, before subtracting them as one from the original basic cube solid.

MINOR ALTERATIONS

So far, our piece pretty much matches the dimensions of standard Technic beams, but there are now a few modifications to make. First, we need the

holes to be a little further from the wall (this will allow us to use angled pieces when building the shelf), so add another two Lego units (3.2 mm) to the bottom of the main cube solid. Second, you will need to reduce the size of the cube solid by at least 0.1 mm on the left, right, front, and back (i.e. 0.2 mm less width and depth) so that the fit is not too tight (Lego also do this).

Export your model as an STL file and print it. Once you've successfully printed one mount, check to see that it fits well with a Lego Technic piece. Try putting an axle or a peg in the hole, then modify or reprint






Left  The 3D-printed part is completely hidden in the finished design

Below  You can build your model before or after the mounts are attached to the wall

the part if necessary. Once you're happy, print the other matching part.

ON THE LEVEL

Next, we need to secure the mounts to the wall. Use a spirit level to check that each piece is properly aligned, both relative to itself and the other piece. If you want to be really accurate, you can build both mounts into a simple temporary Lego model to help get the spacing exactly right. If this is your first time mounting something to a wall, take your time! Our example is mounted on wood in a workshop, which only requires wood screws, but if you're doing this in a house, make sure you don't drill through an electrical cable, and use the appropriate wall plugs and matching screws.

With your mounts in place, you can build your shelf. We're not going to give you instructions for this part, because half the fun of Lego is figuring out the design. Bear in mind that the shelf will have to be fairly strong in order to support a model, particularly at the furthest point from the wall. Long Technic beams are good for adding rigidity (see the section about angled pieces), while flat 'plate' pieces are useful as the top surface, but don't feel you have to cover the entire top surface unless you want to – the main thing is that your shelf has enough structural integrity not to bend once you place a model on it. Add a few decorative pieces to finish, and you're done! 



STRONG BUILDS

A good way to strengthen your Lego builds is by using Technic beams at an angle. One of the most straightforward ways to do this is with 3:4:5 triangles, i.e. right-angle triangles which have a length of exactly three units on one side, four on another side, and five on the hypotenuse. In the case of Lego, this unit could be the distance between two holes (or studs). You can then scale this up, so a 6:8:10 triangle would also work, and is a good fit for Lego Technic because there are beams with seven holes, nine holes, and eleven holes (note that you need one extra hole for each side because the first one is effectively at zero).

QUICK TIP

There are many alternatives to FreeCAD for designing your parts – Tinkercad and SketchUp Free are two popular options, and both run in the browser, so you don't need to install anything.

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



SUBSCRIBE FROM JUST £5

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

Go ahead! Make something with your Raspberry Pi today

FREE PI ZERO W STARTER KIT*

With your 12-month subscription to the print magazine

magpi.cc/12months

* While stocks last



Buy online: store.rpiexpress.cc

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

REVIEWS

PG
106

T-WATCH 2020



An Arduino-compatible you
can wear on your wrist

PG
108

DIAL CALLIPERS

Make measurements
without batteries

PG
110



DREMEL 8220

A powerful rotary tool
without wires

PG
112

INKSCAPE 1.0

The open-source vector
editor grows up

PG
100

BEST OF BREED

Find the best dev board
for your AI project



PG
113

THE STORY OF TOOLS

Expert makers reveal their
favourite hardware

ONLY THE
BEST

Machine learning development boards

Get started building projects with built-in artificial intelligence

By Marc de Vinck

 @devinck

W

ho isn't at least mildly interested in artificial intelligence (AI)? But what exactly is AI? What about deep learning, machine learning, or edge computing?

There are so many terms and technologies out there that it's hard to keep track. For now, just know that artificial intelligence is intelligence that can be demonstrated by a machine rather than a human. For example, you can have your project identify a person or a specific object. It could also be something much more common, like speech recognition. Another important thing to know is that your next project can implement some kind of artificial intelligence or machine learning without breaking the bank.

The different learning architectures and systems related to AI are way beyond the scope of this Best of Breed. If you are interested in building a smart robotic platform, vision system, or really any electronics project that is capable of learning, understanding, and making decisions on its own, you can. Yes, it can get complicated in some cases, but it's achievable with so much content available online, and the help of the open-source community.

One thing to note: the DIY field of AI is still in its infancy. It changes and grows every day. It's exciting! Utilising software like TensorFlow, even a fairly inexperienced programmer can build a robot that uses a neural network model to discern a shirt from a dress, which is absolutely amazing. And that technology is thanks to all the efforts and sharing of the online AI community. Just keep in mind, you most likely won't be able to build a robot companion like you see in so many sci-fi movies. At least, not yet!

So, what can you do with the development boards in the roundup? There are a lot of examples out there, ranging from smart game-playing robots to interesting vision systems that can identify text or specific shapes. You could build an autonomous robot – not just a simple line-following robot, but one that can identify and follow a specific person or navigate a road and obey signs or other visual inputs. Or how about smart alarm system, or predictive robotic assistant? There are so many possibilities once you understand the process.

If you like to be on the cutting edge of technology, and want to get untethered from your desktop computer, then you should check out these development boards and start dreaming up how you can add some 'smarts' to your next project.

Coral Dev Board vs NVIDIA Jetson Nano Developer Kit

CORAL ♦ \$129.99 | coral.ai

NVIDIA ♦ \$99 | developer.nvidia.com

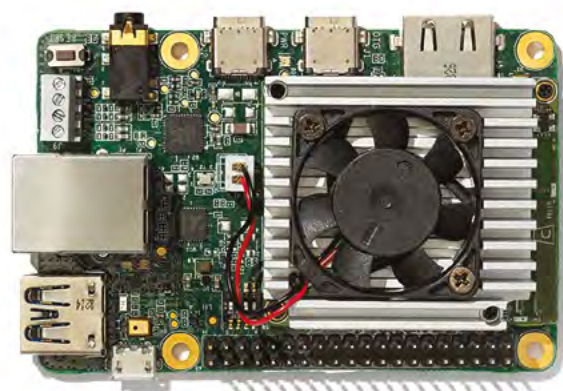
Brought to you by Google Research, the Coral Dev Board packs a lot of processing power and cutting-edge technology into a small footprint.

It's an ideal board for local deployment of a trained machine learning model, and inferencing. Or, to put it in more simplistic terms, you can run TensorFlow Lite tools on the Coral Dev Board to quickly analyse data and make decisions without having to send it back to a centralised computer.

At its core is an Integrated GC7000 Lite Graphics GPU, and an i.MX 8M SoC by NXP. Couple those components with the integrated TPU coprocessor which can perform four trillion operations per second (TOPS) – yes, four trillion! – and you have an amazing machine learning platform.

You would think all that processing power would use a lot of electricity, but surprisingly it only sips a mere 0.5 watts per TOPS.

The whole Coral ecosystem is designed to scale from prototype to manufacturable product. Start with this dev board, prove your concept, then you can move on to your own hardware design that implements the available system-on-module versions which include the CPU, GPU, TPU, and more.



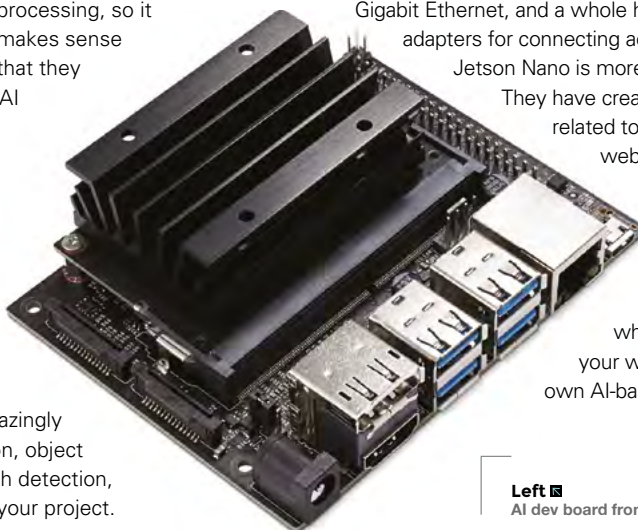
Above ♦
Quickly develop local machine learning

Who doesn't recognise the brand **NVIDIA**? They are well-known for their high-end graphics processing, so it makes sense that they

would also be in the AI development board arena. The NVIDIA Jetson Nano Developer Kit is a powerful computer that allows you to run multiple neural networks in parallel. What does that mean? It means you can do blazingly fast image recognition, object detection, and speech detection, and all locally within your project.

The NVIDIA Jetson Nano Developer Kit features a 128-core Maxwell GPU, a quad-core ARM Cortex-A57, 4GB memory, high-end video encoders and decoders, Gigabit Ethernet, and a whole host of plugs and adapters for connecting accessories. But the Jetson Nano is more than just hardware.

They have created a robust ecosystem related to AI. Head over to their website to not only grab the required resources, but also check out their free Deep Learning Institute Course, which will get you well on your way to developing your own AI-based project. >



Left ▣
AI dev board from a name we all recognise

VERDICT

Coral Dev Board

A good entry point, with potential to scale.

10 /10

NVIDIA Jetson Nano Developer Kit

An amazing piece of architecture at a great price.

10 /10

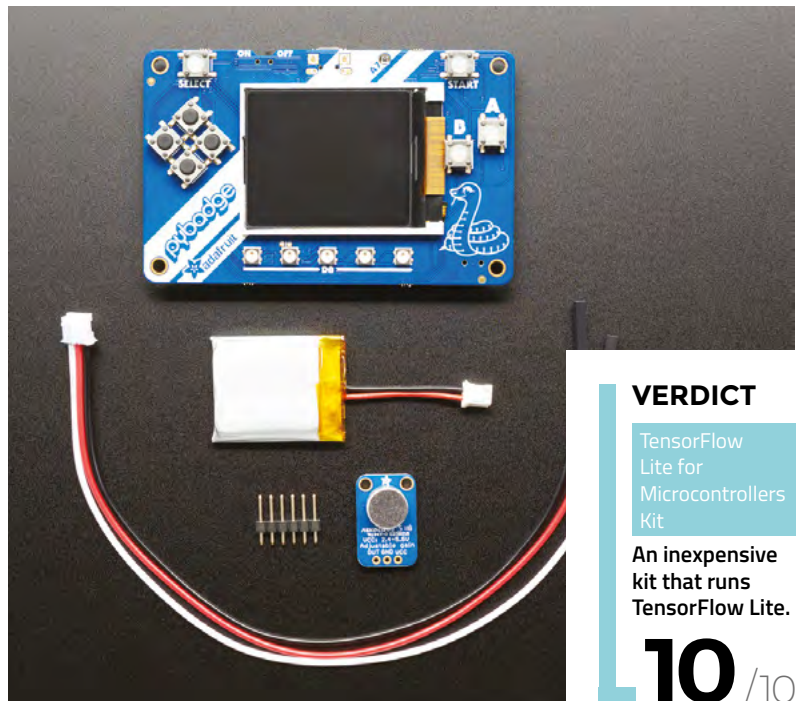
BEST OF BREED

TensorFlow Lite for Microcontrollers Kit

ADAFRUIT ◆ \$44.95 | adafruit.com

If jumping into the world of machine learning all seems too complicated and expensive, but you still want to get your feet wet, then the TensorFlow Lite for Microcontrollers Kit from Adafruit might just be perfect for you. The kit contains the Adafruit PyBadge with ATSAM51 ARM Cortex-M4F processor, a built-in display, multiple buttons for feedback, and an integrated speaker. The kit also includes an electret microphone, LiPo battery, and all the necessary wires to hook everything up. They took all the worry out of what you need to get started with machine learning, and combined it into a simple and affordable kit.

So, what can you do with this kit? A lot actually! You can use it to run TensorFlow Lite and, with the included microphone, create a project that understands, and reacts to, simple voice commands. You can also try out gesture sensing or several other beginner-friendly projects that are all documented on their website. Just keep in mind that you will need to do a little bit of soldering, but even a beginner shouldn't have any problems getting up and running quickly with this low-cost, yet powerful, kit.



VERDICT

TensorFlow Lite for Microcontrollers Kit

An inexpensive kit that runs TensorFlow Lite.

10 /10

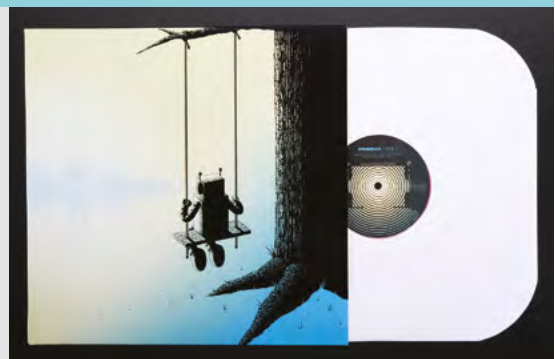
Right

An affordable and approachable way to get started with AI

FREQUENCY

ADAFRUIT ◆ \$19.95 | adafruit.com

AI projects take a lot of hard work. And all that hard work can get mundane at times. That is, unless you have some inspiring music playing in your studio. Check out Adafruit's first vinyl record, *Frequency* by Bartlebeats. When everyone is going digital, this physical vinyl record really stands out. You'll need a record player for this product, but if you are into AI and electronics, maybe you could make one of those too!



Grove AI HAT for Edge Computing

GROVE ◆ \$28.90 | seeedstudio.com



Right ■
Plug and play AI add-on for your Raspberry Pi

The Grove AI HAT for Edge Computing is an add-on for your Raspberry Pi that integrates a Sipeed MAIX module. This adds a dual-core 64-bit CPU along with a slew of other AI-related features to your Raspberry Pi.

And although you can run TensorFlow on your Raspberry Pi without the extra horsepower, this HAT will speed everything up.

The Grove AI HAT also allows you to add a camera and LCD as needed for your project, and it has a built-in microphone and three-axis accelerometer which are useful for AI-based robotics. There is a wide variety of breakout boards and shields available from

Seeed Studio, all based on the Sipeed module, including versions for Arduino. Keep in mind that the documentation for this is light, so we can only recommend it for someone who really knows their code and electronics. If that's you, this could be a good choice for adding some extra processing power to your Raspberry Pi-based AI project. Check their website to find the best version for your project and the available resources.

VERDICT

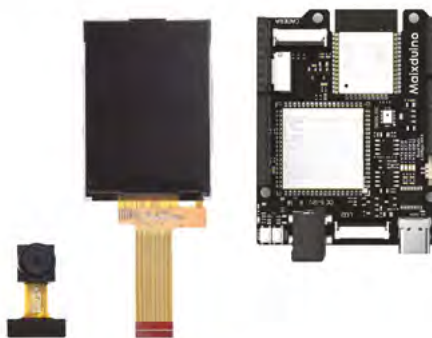
Grove AI HAT for Edge Computing

Affordable AI for Raspberry Pi.

7/10

Sipeed Maixduino Kit for RISC-V AI + IoT

SIPEED ◆ \$23.90 | seeedstudio.com



Left ◆
A lot of technology at a very affordable price

If you are willing to do a little extra research, and would like to save money, take a look at the Sipeed Maixduino for RISC-V AI + IoT combo pack from Seeed Studio. This is very similar to the other Sipeed boards, but this one is in an Arduino-friendly form factor. The board includes the Sipeed M1 module, a MEMS microphone, Audio DAC, ESP32 module with WiFi and Bluetooth, TF card slot, and more.

You can also easily add the included 2.4-inch TFT LCD display and camera with the provided components and connectors. This is a fairly inexpensive way to get

started with machine learning and vision systems, all in the ever-popular Arduino form factor. Keep in mind that, unlike most other Arduino boards and shields, this one should be reserved for someone with a little more advanced knowledge of programming and electronics. →

VERDICT

Sipeed Maixduino Kit for RISC-V AI + IoT

Inexpensive, but lacks in documentation.

8/10

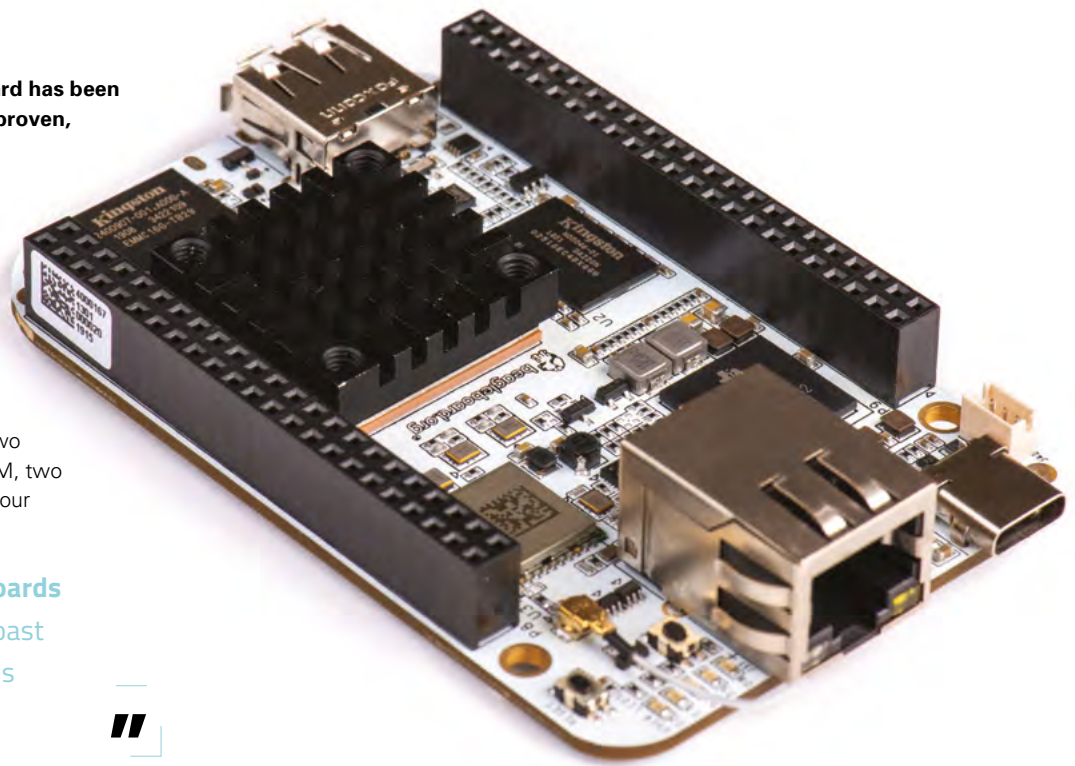
BEST OF BREED

BeagleBone AI

BEAGLEBONE ♦ \$127 | beagleboard.org/ai

The BeagleBone AI board has been designed around the proven, and very popular, BeagleBone Black form factor, and built around the open-source Linux community.

This variation of the BeagleBone board is based on the Texas Instruments Sitara AM5729, and the board packs a lot of power, including notable features like dual ARM Cortex-A15 microprocessors, two floating-point DSPs, 2.5MB of L3 RAM, two dual ARM Cortex-M4 coprocessors, four



“ Not a lot of other boards in this class can boast all those features and more **”**

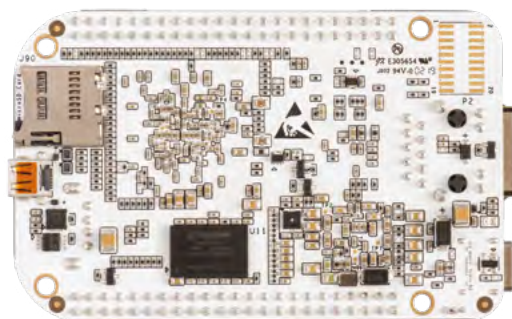
Embedded Vision Engines, a 2D-graphics accelerator, and a dual-core 3D GPU. Not a lot of other boards in this class can boast all those features and more. Be sure to check out the extensive documentation, open-source design files, and quick-start guide on the product page. □

Above ▣ An AI dev board from a long-time favourite of the DIY community

VERDICT

BeagleBone AI
A board from a company with a long history of making quality products.

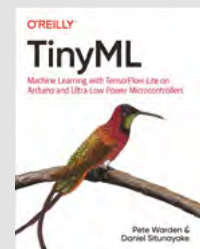
9/10



TINYML

O'REILLY ♦ \$32.99 | oreilly.com

If you aren't sure where to start, and want to keep it simple in terms of electronics, take a look at *TinyML* by Pete Warden and Daniel Situnayake. This book shows you how to implement TensorFlow Lite on an Arduino. The authors show you how to build speech recognition, train models, and implement image processing. It's a good overview of basic AI on a microcontroller.



GET STARTED

WITH



ARDUINO

Robots, musical instruments,
smart displays and more



Inside:

- Build a four-legged walking robot
- Create a Tetris-inspired clock
- Grow veg with hydroponics
- And much more!



AVAILABLE
NOW

hsmag.cc/store

plus all good newsagents and:

WHSmith

BARNES & NOBLE



Available on the
App Store



GET IT ON
Google Play

FROM THE MAKERS OF **HackSpace** MAGAZINE

LILYGO TTGO T-Watch-2020

Strap a programmable microcontroller to your wrist

LILYGO ◆ \$28.50 | [tindie.com](https://www.tindie.com)

By Ben Everard

🐦 @ben_everard



Above ◆
After six hours use, the watch still has two-thirds of its battery remaining (and it was only charged to 90% at the start)

W

atches are surprisingly complicated devices. They need to pack some complex electronics, a one-handed user interface, and a battery into a package small enough

to unobtrusively sit on your wrist. Makers and hackers have long been making watches – there are some you can buy as a kit, and there are some that can be re-flashed with other firmware, but this is the most technologically advanced watch designed to be hacked that we’re aware of. It packs an ESP32 microcontroller (with WiFi and Bluetooth), a 240x240 TFT touchscreen, a three-axis

accelerometer, a buzzer, a vibration motor, and a 380mAh battery into a unit that’s 48mm × 42mm × 13mm. While this is a little chunky for a watch, it’s not excessively so, and it is small enough to be worn comfortably. All this comes for \$28.50, plus delivery (\$5.50 to the UK). This puts it as one of the more competitively priced smartwatches, but can it live up to its promise?

The build quality is good for the price. The glass front and metal sides feel sturdy, though we haven’t had it long enough yet to really vouch for its toughness. The back panel, which is clip-on plastic, exposes the electronics and removable battery. While this makes it easy to get into, it does leave us some concerns about how vulnerable this watch will be to water.

SOFTWARE WOES

The biggest disappointment in this watch is the stock firmware. When you first turn on the device, you’re greeted with a touch interface that looks good, but is ultimately pretty useless. It can connect to WiFi (if you’re patient enough to use the atrocious password entry system), and get time from NTP, but there’s no way of changing the time zone, so you’re stuck with China Standard Time (CST) if you use this. The complex interface chews through the battery far too fast to be useful, unless you enjoy plugging your phone into a USB cable every few hours. Fortunately, this device is designed to be reprogrammed, so the real test is how easy it is to code it into something more useful.

There is an Arduino library for accessing the phone’s hardware at [hsmag.cc/SPIWgx](https://github.com/hackspace/HackSpace). Unfortunately, the documentation for this is almost non-existent at the time of writing, however, there are some example projects to get you started. These aren’t commented well, but if you’re an

experienced programmer with some familiarity with the Arduino and ESP32 platform, you should be able to figure out enough to start building something.

A challenge with working on this device is the limited battery life. If you want to use the device as a watch, you'll probably want to go at least a day between charges – this is only possible if you avoid wasting power. To test out how functional this was as a watch, we put together a minimum viable smartwatch project that just displays the time and counts steps using short clicks on the power button to turn the screen on and off. You can see the code here: hsmag.cc/SaK7MMN. It should give you a good base for building your own simple watch projects.

You'll need to familiarise yourself with using interrupts and messages because these are key to how the library is structured. They let you keep an eye on what's happening in the device without having to check the status of everything. For example, the power button has a short-tap option which is handled by the AXP power management unit. You can tell if there's been a short tap with:

```
if (ttgo->power->isPEKShortPressIRQ()) { ... }
```

It comfortably lasts a day in our testing, though this will depend on how often you check the time and step count. There's plenty of space left on the screen for adding additional information read-outs (such as displaying information grabbed from the internet). The display supports Adafruit's GFX library, which



```

BMA423_StepCount | Arduino 1.8.5
File Edit Sketch Tools Help
BMA423_StepCount config.h
void loop()
{
  int per = ttgo->power->getBattPercentage();
  if (irq) {
    irq = 0;
    bool rlat;
    do {
      rlat = ttgo->bma->readInterrupt();
    } while (!rlat);
    ttgo->power->readIRQ();
  }
  if (ttgo->bma->isStepCounter()) {
    Serial.println(ttgo->bma->getCounter());
    ttgo->tft->setTextColor(random(0xFFFE), TFT_BLACK);
    sprintf(buf, "%d", ttgo->bma->getCounter());
    ttgo->tft->drawString(buf, 22, 60, 4);
  }
}

```

makes it easy to draw graphics. While we've used a simple `drawString()` method for putting text on the display, you can get far more creative if you want to. Alternatively, you can use the Light and Versatile Graphics Library (LVGL) to create an interface.

AUDIO OUT

You don't just have to get output via the screen. There's also a buzzer and vibration motor for keeping track of what's going on when you don't have time to look at the screen.

If you're looking for a more time-related project, you can also access the real-time clock which includes features such as alarm interrupts.

For the price, this is a fantastic wearable computer for anyone willing to put in the effort to bend it to their needs. It can certainly be used as the base of a smartwatch, but it can potentially be much more. Just a few projects that we'd be interested to see are:

- Interacting with other devices over WiFi or Bluetooth, and therefore be a portable controller for almost any of your projects
- The perfectly customisable accessory to a cosplay outfit
- A games controller powered by the accelerometer sending commands over Bluetooth

These are just our ideas though. You may well have some more thoughts.

This is a powerful tool for making some really exciting projects. We're looking forward to developing more projects with it over the coming months, and also seeing what uses other people find for it. □

Above ♦ Despite the poor documentation, we were able to build a simple interface for the watch quickly by copying code from the examples

Below □ The default firmware looks pretty, but is almost useless

VERDICT

A fun platform for wearable computing, but lacking in documentation.

9/10

Moore and Wright Dial Calliper

Ditch the batteries!

MOORE AND WRIGHT ♦ £51 | machine-dro.co.uk

By Jo Hinchliffe

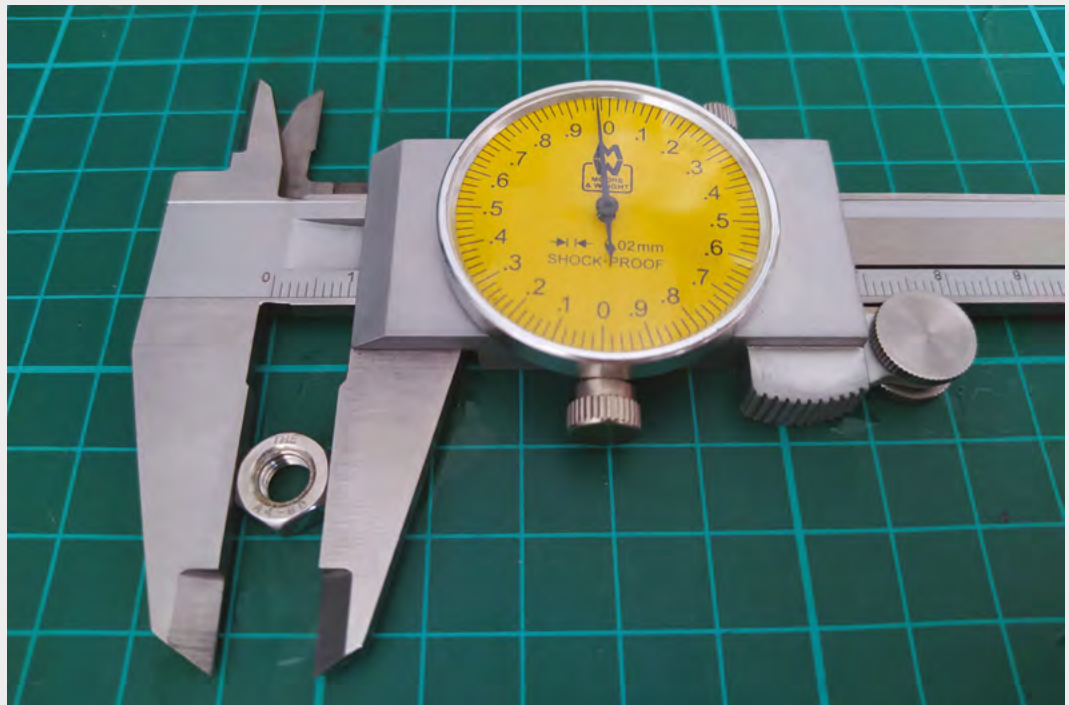
[@concreted0g](https://twitter.com/concreted0g)

While we love the ubiquity of cheap and affordable digital vernier callipers available online, the ones at the cheaper end of the market do sometimes have their issues.

We've lost track of how many cheap digital callipers we have had that have accidentally turned on in their case and then had no battery power when needed,

or indeed that pair of callipers ordered online that, on arrival, turned out to be made of plastic! At the other end of the scale (pun intended), original analogue-style vernier callipers with a vernier scale can often be confusing for those who don't know how to read the system, and therefore have become a less common sight on the workbench.

Sitting in the middle of these extremes are mid-priced dial callipers, such as this set of Moore and



Right ♦ Using the dial callipers to measure a nut, the combination of reading the ruler scale and then the dial becomes second nature very quickly



Left ♦ Dial callipers have the advantage of never running out of batteries, but they are harder to read than digital callipers. They are, however, easier to read than classic vernier scale callipers

Wright 150mm metric callipers available from the Machine DRO company, which is a part of the larger Allendale Group. Moore and Wright are an iconic tool brand in engineering and machining circles – tools from their extensive back catalogue, which dates back to their formation in 1906, are still collected and revered and used in many tool rooms.

These dial callipers arrived promptly and well-packed; the callipers themselves were in a fitted plastic case, inside a cardboard sleeve, inside a cardboard box. Once unwrapped and in hand, it was immediately obvious that they're a quality item. Sliding the calliper open and closed felt incredibly smooth, and puts to shame many of the £10 internet-ordered digital callipers we've had over the years! We went with the yellow-faced, high-contrast dial, which is easy to read and has prominent markings.

As is common to most callipers, these have a locking bolt to hold the jaws of the calliper in a certain position, but they also have another screw mechanism under the dial bezel to be able to zero the dial graduations. It's a simple procedure to zero the dial: loosen the bolt, ensure the calliper jaws are clean, close the jaws, rotate the bezel so that the dial's zero mark aligns with the needle, and then re-clamp the bezel bolt. This method of zeroing also means you can zero the dial at a particular reference point if needed.

“ **The callipers are made of good-quality steel, and the jaws are hardened and ground to a high-precision finish** ”

The callipers are made of good-quality steel, and the jaws are hardened and ground to a high-precision finish. There are the usual internal and external jaws, and also the depth rod for measuring heights and depths. Measuring on these callipers is very straightforward. The nearest millimetre graduation is read off the ruler scale marked along the calliper, and the dial is marked with 100 graduations, each

representing 0.02 mm. This means that a single rotation of the dial results in 2mm of travel. The dial is nicely marked with every 1/10th of a millimetre labelled 0.1, 0.2, 0.3, etc., and restarts at 0 for the second millimetre of

travel. It becomes second nature to read this dial to 0.02mm very quickly and, while accuracy may vary, it's possible to predict roughly the third decimal place as to how far between 0.02mm markings you are. If you need accuracy to thousandths of a millimetre though, you probably need to move beyond callipers! As mentioned, the set we have reviewed are the 150mm option, but a 300mm version is also available.

At a little over £40 plus VAT, we feel these callipers are accurate, well-made, and extremely nice to use, with the added bonus that they will never have run out of batteries when you need them most. Couple that with a company that offers great customer service and prompt delivery. □

VERDICT
Accurate and easy to use.

10/10

Dremel 8220

A rotary tool that's not tied down

DREMEL ♦ £165 | dremel.com

By Ben Everard

 @ben_everard

To many people, the name **Dremel is synonymous with rotary tools.** These spinning machines are so useful, it can be a little hard to pinpoint what they're used for – they've got so many different workshop applications that people use them in many different ways. They can be used to cut and grind metal, stone, wood, and plastic; they can polish almost anything; they can tidy up the rough edges left behind by 3D-printed supports; with an adapter, they can be turned into a drill-press, router, and even a CNC milling machine.


Dremel sent us the 8220-5/65 kit to review, which comes with five attachments and 65 accessories in a carry case which retails for £165, but you can also get a more stripped-down kit which comes with just the basics for £99. There's no doubt that this pricing puts it at the top end of rotary tools, so let's see if it's worth it.

The main feature of this tool is that it's a cordless rotary tool with the same power as the higher-end, mains-powered versions. That alone will justify the price for some people. You can smoothly adjust the speed from 5000 to 35,000 rpm (remember

though that you shouldn't exceed the rated speed of the accessories you're using). This oomph comes from the 12V motor, and makes a big difference if you're working with harder materials, and this is the first cordless rotary tool for which metalwork is really an option. Others we've tried are so slow as to be frustrating.

It's not just about raw power, though. There are plenty of touches that make this a nice tool to use. The rubberised surround makes it comfortable to use, and the light attachment makes it easy to see what you're working on. Users of rotary tools will be familiar with the rounded square collet that you need to unscrew to change tools. Usually you get a spanner to undo the collet and... well, if you're anything like us, those spanners have a habit of disappearing. With



Above  The Dremel 8220 packs a powerful 12V motor, while still being portable

the 8220, there's a collar that unscrews from behind the collet and performs the job of this spanner – no frustrating searches between tool changes!

ACCESSORISE

The five accessories that came with our kit show off the potential of the tool. You get: a line and circle cutter, a tile cutter (like a sort of a very basic router), a detailer's grip, a shaping platform, and a transparent guard. Of these, the detailer's grip and transparent guard are probably of most use to makers. The grip helps you hold the tool steady for fine work (though if this is how you want to use your rotary tool the majority of the time, you may be better suited to using either a lighter tool or a flex shaft), while the transparent guard makes high-speed grinding a little safer.

Alongside these, you get a range of the usual rotary tool bits for sanding, cutting, grinding, and polishing. The Dremel accessories are noticeably higher quality than the cheap ones available from a wide range of other manufacturers. This makes them nicer to use as you spend more time doing what you want, rather than replacing worn-out tools.

You also get two of the SpeedClic cut-off wheels. These are 38mm abrasive cutting wheels with a novel attachment mechanism where you push and twist them into place, rather than using a screw. They're much quicker to use, and we've found them a bit less prone to shattering as there's a little bit of give in the mounting, should you put pressure on the side of the disc. These aren't cheap, but they are quicker and easier to use than traditional wheels. Whether they're worth it depends on where you come down on the price vs convenience debate.



Above ♦ The collet remover is integrated into the tool itself – no more lost spanners!

These accessories and attachments are all solid and well-made. If you're new to rotary tools, it may be a good investment to get your toolkit fully stocked, but for many people, it'll make more sense to start with a minimal kit and build up the pieces as and when you need them.

This is the best rotary tool we've used due to the power, portability, and build quality, but that comes with a price. Whether or not it's worth it will depend on your use-cases. If you don't need to work with harder materials, you may find that the premium price for a powerful tool isn't worth it, or if you're only an occasional user, the price for going battery-powered may be hard to justify. However, if, like us, you're regular rotary tool users, even with the premium price, this is an attractive tool. □



Right □ Cut, polish, grind, and sand with the included bits

VERDICT

A powerful, well-thought-out portable tool.

9/10

Inkscape 1.0

Open-source vector graphics grows up

INKSCAPE TEAM ♦ Free | inkscape.org

By Ben Everard

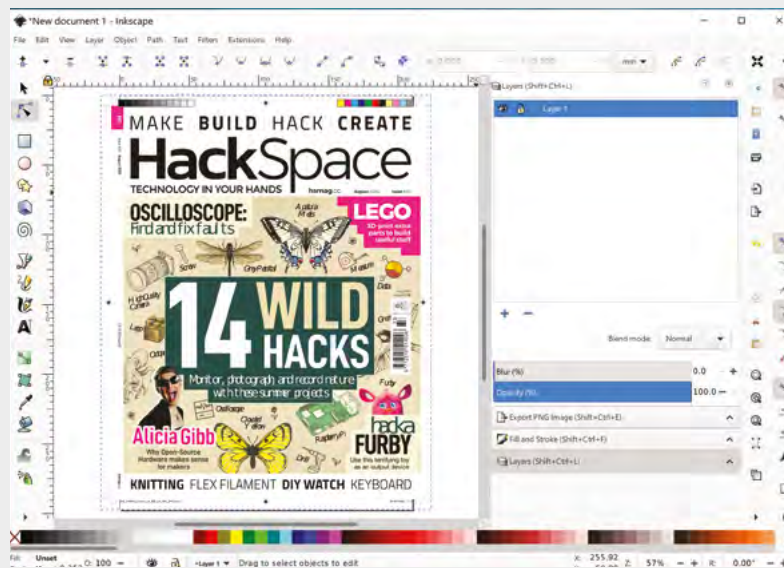
🐦 @ben_everard

Inkscape is one of the staple bits of maker software as it's powerful, open-source, and relatively easy to get started with. As general-purpose vector drawing software, Inkscape is useful for a wide range of people, such as artists and diagram makers. Because it's vector-based, it stores the lines and shapes in the drawing, not just the colour values of pixels (as a raster editor does). This means that tools such as laser cutters can import the vectors and convert the lines and shapes in your image to cuts, or other tool movements.

In a hackspace or other multi-user workshop, the open-source nature of Inkscape is a useful feature, because it means that lots of members can use the same software without having to worry about licenses or payments.

Inkscape has been around since 2003, yet the application didn't reach version 1.0 until May 2020. It's this major milestone release that we're looking at here.

If you've used Inkscape before, the new version will look familiar, but a little more modern. This is thanks to the upgrade to GTK3. This upgrade in graphics toolkits brings with it a host of improvements for modern systems, most noticeably support for high DPI screens and a native OS X version. The former of



these is a huge boon for this reviewer, as previous versions had really struggled on his machine.

There have been far too many improvements to go through them all here (see the release notes at hsmag.cc/zGvUvi for a full list). The area that stands out to us as useful for makers is the changes to Live



If you've used Inkscape before, the new version

will look familiar, but a little more modern

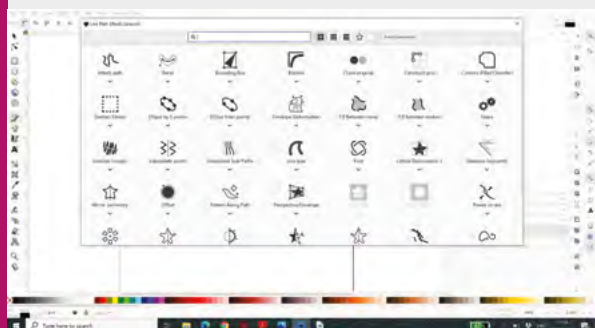


Above ♦

The new Live Path Effects make it easier to design and manipulate vector objects

Below ▣

The new interface is laid out in the same way, but looks so much better on high-resolution screens



Path Effects (LPE). These allow you to make reversible changes to a vector path. The whole area of LPEs has had an overhaul, but a couple of new options may be of particular interest: fillet/chamfer and measure segment.

The one note of caution is that the extensions API has changed, and while many of the extensions have been ported, not all of them have.

If you particularly rely on any extension as part of your workflow, Inkscape was already a great choice as a vector image editor, and the new improvements in this version make it even better. □

VERDICT

The premier open-source vector editor gets even better.

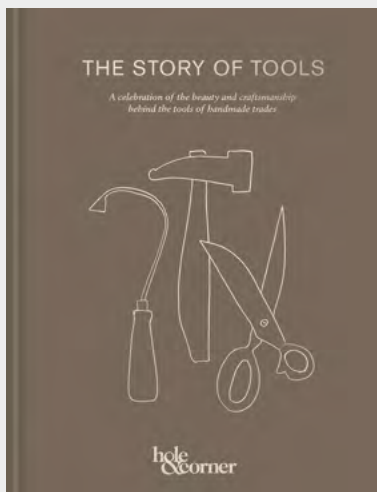
9/10

The Story of Tools

VARIOUS ♦ £20 | Pavilion

By Ben Everard

 @ben_everard



There's great pleasure in using the right tool. The one that fits just perfectly in your hand and feels like an extension of yourself. In this book, makers from a wide variety of specialities talk about the tools that are just right for them. It reveals the deeply personal nature of tools: many are either custom-made, modified, or so worn from years of work that they bear little resemblance to their original forms.

In revealing why these makers use the tools they do, we get an insight into what it means to be an expert craftsperson. This book also made us think about our own relationship with tools. There's a temptation to quickly stock our workshop with tools that are easily and cheaply available, but where will these tools be in ten years, or 100 years? Will we pass them on to our children, or will they just clog up landfill? Do they merely function, or are they a joy to use? Does this matter? Should we understand our tools enough to make them ourselves, or is this a distraction from the projects we want to work on? The book doesn't provide answers to these questions. There aren't really answers to them – at least not in general terms – but that doesn't mean we shouldn't be asking them, and thinking about what we're putting in our hands.

If you enjoy making, then you probably enjoy a good tool, and if you enjoy a good tool, then you'll enjoy this book. □

VERDICT
Beautiful
and thought-
provoking.

9/10

next month

issue

#34

ON SALE
20 AUGUST

Get started with

Feather

ALSO

- Recycle fabric
- Electronic music
- MicroPython
- 3D printing
- **And much more**

DON'T MISS OUT

hsmag.cc/subscribe





"Open-sourcing your hardware gives it the freedom to grow in ways that you would never have thought of."



Alicia Gibb

Executive director of the
Open Source Hardware Association
(OSHWA)



