

MAKE | BUILD | HACK | CREATE

132  
PAGES  
OF MAKING

# HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

October 2019

Issue #23

**HALLOWEEN**

**BUILDS**

**TEENSY 4.0**

The fastest microcontroller?

**LEARN TO BRAZE**

Molten metal joints

Oct.2019 Issue #23 £6

9 772515 514006

23

HA HA HA HA HA HA

MAKE SPOOKY, SCARY AND UNCANNY CREATIONS

IFTTT

Add your devices to the Internet of Things

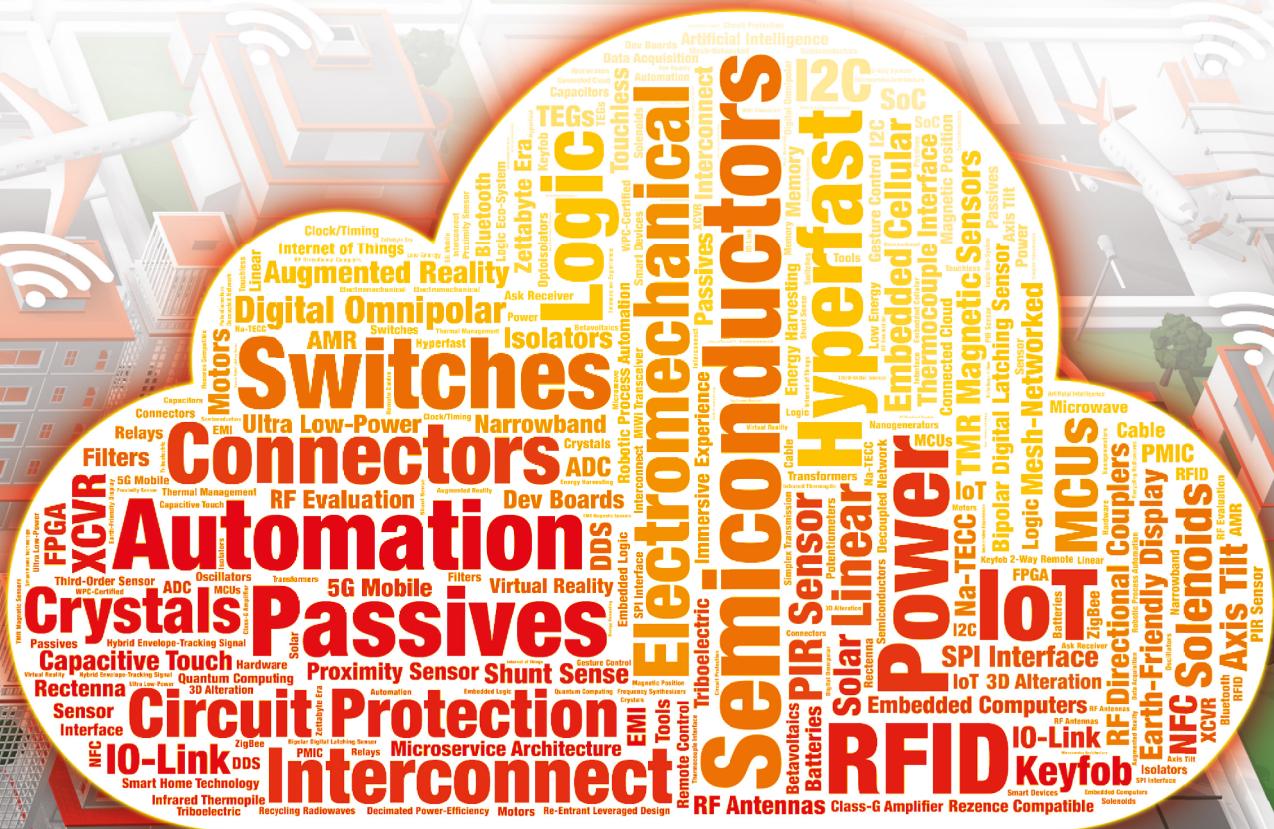
PRETTY PCBs DESIGN CIRCUITS IN INKSCAPE

NAIL ART BECOME A TEMPORARY CYBORG

SKULL SYNTH MAKE MUSIC ON A DEAD RACCOON

COSPLAY PROPS VULCANOLOGY 3D PRINTING

# HELPING DRIVE YOUR INNOVATION



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



0800 587 0991  
**DIGIKEY.CO.UK**

**Digi-Key®**  
ELECTRONICS

7.9 MILLION+ PRODUCTS ONLINE

800+ 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. © 2019 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

ECIA MEMBER  
Supporting The Authorized Channel



# Welcome to HackSpace magazine

This month we can dispense with practicalities. We can put aside function, and dispense entirely with features. October, particularly its final day, is all about looks – spooky, scary looks to be precise. It's a time to have fun with your skills – maybe this means an opportunity to try something new, maybe this means the chance to show off some new technique you've learned, or maybe it's simply a chance to stand a few hairs on end.

The beauty of making things for Halloween is that it's entirely about having fun – whatever that means to you. For some people, this means concentrating on electronics and building some intricate mechanisms, for others, it means big elaborate statement pieces, and there are those who might go in for really detailed, realistic builds. It doesn't matter, as long as it's what you're interested in.

If you're struggling for ideas, then we've got plenty to spare! Take a look at page 42 to dive into the spooky selection. If you prefer to hide from trick-or-treaters with the lights turned off, then fear not, we've also got a magazine jam-packed with all the usual hacking and making. Turn the page to get stuck in.

## BEN EVERARD

**Editor** @ben.everard@raspberrypi.org

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

get in touch at  
[hsmag.cc/hello](http://hsmag.cc/hello)

## GET IN TOUCH

✉ [hackspace@raspberrypi.org](mailto:hackspace@raspberrypi.org)  
✉ [hackspacemag](http://hackspacemag)  
✉ [hackspacemag](http://hackspacemag)

## ONLINE

✉ [hsmag.cc](http://hsmag.cc)

Available on the  
App Store

GET ON  
Google Play



## EDITORIAL

### Editor

Ben Everard  
✉ [ben.everard@raspberrypi.org](mailto:ben.everard@raspberrypi.org)

### Features Editor

Andrew Gregory  
✉ [andrew.gregory@raspberrypi.org](mailto:andrew.gregory@raspberrypi.org)

### Sub Editors

David Higgs, Nicola King

## DESIGN

### Critical Media

✉ [criticalmedia.co.uk](http://criticalmedia.co.uk)

### Head of Design

Lee Allen

### Designers

Sam Ribbits, Harriet Knight

### Photography

Brian O'Halloran

## CONTRIBUTORS

Lucy Rogers, Andrew Huang, Jo Hinchliffe, Emily Velasco, Mayank Sharma, Jo Hinchliffe, Andrew Lewis, Sophy Wong, Gareth Branwyn, Les Pounder, Marc de Vinck

## PUBLISHING

### Publishing Director

Russell Barnes  
✉ [russell@raspberrypi.org](mailto:russell@raspberrypi.org)

### Advertising

Charlie Milligan  
✉ [charlotte.milligan@raspberrypi.org](mailto: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  
✉ [hsmag.cc/subscribe](http://hsmag.cc/subscribe)

### Subscription queries

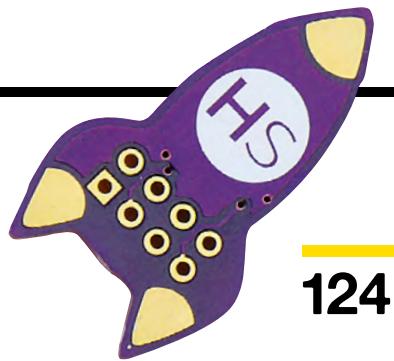
✉ [hackspace@subscriptionhelpline.co.uk](mailto:hackspace@subscriptionhelpline.co.uk)



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



124

06

## SPARK

- 06 Top Projects**  
You're a clever bunch, you lot
- 16 Objet 3d'art**  
Additive automotive manufacturing
- 20 Meet the Maker: Michael Dales**  
Meet the Leo Fender of the makerspace
- 24 Columns**  
Shoot down some pesky UFOs
- 26 Letters**  
Repair your stuff, repair the Earth. Simple
- 28 Kickstarting**  
Control paper aeroplanes from a smartphone
- 30 Rockets**  
Powered flight at the International Rocket Week
- 32 Space of the month**  
Radiona: a hub of learning in beautiful Zagreb
- 36 Forged in foam**  
Inspiring builds from a master of cosplay

41

## LENS

- 42 Halloween**  
Darkness falls across the land; the making hour is close at hand
- 58 How I Made: Rackety Raccoon**  
Incorporate a dead animal into a synthesizer
- 64 Volcanology**  
Affordable science with the team from Sheffield University
- 72 Improviser's toolbox Socks**  
Warm, comforting makes for cold toes

### Tutorial

School of making:  
Learn to braze



90

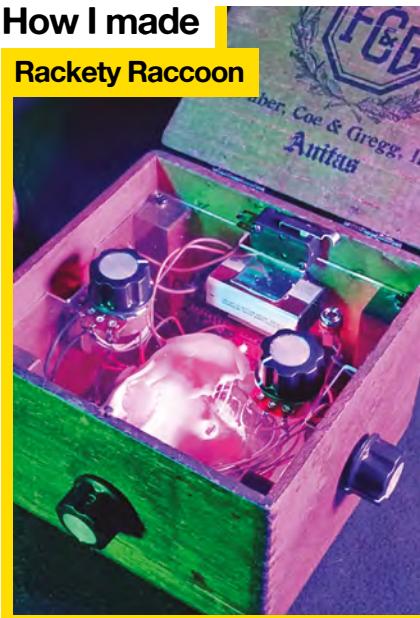
Stick things together with fire  
and hot metal

### Cover Feature



102



**How I made****Rackety Raccoon****58**Summon spooky sounds  
with a dead trash panda**36****Interview****Sheffield volcanologists****64**What links NASA, Raspberry Pi  
Camera Modules, and Jules Verne?**06****77****FORGE****78****SoM If This Then That**

Link smart devices through the Internet of Things

**82****SoM 3D printers**

Explore the anatomy of this makerspace workhorse

**84****Tutorial PCB art**

Link Inkscape to KiCad to create electronic art

**90****Tutorial Learn to braze**

Stick things together with rods of molten metal

**96****Tutorial NFC nails**

Prepare your cuticles for electronic augmentation

**100****Tutorial Servos**

Turn a cheap servos into a continuous motor

**102****Tutorial Cosplay wings**

Combine your skills to make a slick wearable project

**108****Tutorial Serial**

Transfer data over this ancient protocol

**111****FIELD TEST****112****Direct from Shenzhen ESP32-CAM**

A tiny WiFi-enabled camera module for £6.66

**116****Best of Breed**

Power solutions for Raspberry Pi

**122****Can I Hack It?**

Hot Wires – a tactile (and hackable) electronics kit for kids

**124****Review Teensy 4.0**

Fast, small, cheap – pick two. No, pick all of them!

**126****Review Adafruit MONSTER M4SK**

An oracular spectacular for Halloween

**128****Review Programino**

A more user-friendly way to program Arduino code

**129****Book Review Power Carving Manual**

Create high art with your rotary tool

**116**

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.

# Ball balancer

By Johan Link

 [hsmag.cc/snZxeW](http://hsmag.cc/snZxeW)

I'm an 18-year-old Swiss maker, self-taught on electronics and programming. When I started learning how to use OpenCV, I wanted to put my knowledge into practice, that's how I started this project. This robot is able to balance a ball on its flat surface. Three servo motors are used to tilt the plate to compensate the motions of the ball. A USB camera is perched above the platform, and a Python program analyses the images to detect the ball position, then the computer controls the servos via a custom PCB inside the robot. I quickly realised that I was not the only one who had the idea of this kind of robot, that's why I wanted to stand out by making a robot as elegant as possible. □

**Right**  Clean 3D printed parts gives this experiment a lovely finished look





Above □

Hipsters love 12-inch vinyl – now it's time for cassettes to get their moment in the sun



# Cassette art

By Chris Barton

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

**C**hris Barton is an artist with an interesting background in the film industry; creating animatronics for successful franchises including *Star Wars* and *Harry Potter*. Having developed an artistic appreciation for the once-ubiquitous music cassette, he now recreates them in every detail but 64 times larger as sculptural pieces of art.

The construction is meticulous, with hand-painted acrylic on canvas inlays, bespoke cases, and handmade cassettes, faithful in every detail to the original. They are created from a combination of acrylic and a laser-cut wood composite. Operating from his studio in Northamptonshire, around 22 titles have so far been bought by art collectors around the world, with new ones being regularly commissioned. More examples of Chris's work can be found on Instagram, Twitter, and Facebook using **@chrisbartonart** or on his website – **chrisbartonart.com** □



# Giant Game Boy

By Willie Colón

 @thegamegraveyard

**made the initial designs on Illustrator, and from there I made the front and back plates of the Game Boy.** I took my designs to a CNC router, got them cut out, then sanded, and finally, I stained them by hand. I used a cheap TV monitor with an HDMI to connect the Raspberry Pi, and used arcade buttons for the controls on the front plate to make it functional.

Thanks to the help of my good friend Isaac Gelman (another amazing maker), I was able to get some help on the finer details by laser cutting the D-pad and buttons.

The reason I built a giant Game Boy was to see if I could do it! There was a gamer-centric gallery happening where I work at Nickelodeon, and I was trying to figure out if it would be possible to take on such an ambitious project. I have also always been passionate about the original DMG, so it was a perfect fit. It is currently hanging in my living room for everyone to see and play, but when I'm home alone, I use a special Bluetooth Game Boy cartridge, and put it into one of my self-modded Game Boys to control the XL. I love gaming! Video games have always, and will continue to be a huge part of my life. Growing up in Chicago, I was always on the go with my Game Boy Color, so I wanted to take that iconic shape and build what is the Game Boy XL. □

**Right**   
When using emulators, make sure you get your games from a legal source







# Flatpack toaster

By Kasey Hou

 [kaseyhou.com](http://kaseyhou.com)

The project was Kasey's graduation project for her study at the University of Edinburgh. She started it with an investigation on E-waste. The project challenges how most products are made unrepairable by demonstrating how a toaster can be made repairable. People should have the right to repair broken products, rather than having to dispose of them. The DIY aspect of the project enhances the repair experience because people would feel more comfortable to repair the toaster if they learned to assemble it first. The chassis of the toaster is made of several CNC'd stainless steel plates. A mechanical timer, three heating elements, and a couple of wires were used to re-create the toasting mechanism. □



## Right ♦

The right to repair isn't just about legal rights – it needs to be built-in. Kasey's design uses no glue or plastic clips, so anyone can fix it





# RetroPie console

By Martin Mander

 [Kyliemander.com](https://twitter.com/Kyliemander.com)

T

**H**is portable retro gaming console is giving a new lease of life to a 1963 Sky Tourer car radio. Martin drilled out the top to make room for a joystick and six buttons, and cleared out the old electronics to make room for a Raspberry Pi 3 and a Pimoroni Picade board. To make the radio's fascia glow, he's built in a Pimoroni Blinkt! LED strip, used Sugru to neaten the build, and spray-painted the whole thing from the original drab grey to a nice bright red. □



Objet 3d'art

REGULAR

# Objet 3d'art

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





**L**ook more closely at this 1962 Volkswagen bus, made by Volkswagen and augmented by Autodesk: the wheel rims, steering wheel, wing mirror mounts, and other bits are 3D-printed metal. And, rather than being mocked up first in AutoCAD, they're designed automatically in software in a process called 'generative design'.

Generative design, like declarative programming, is design worked backwards. Instead of drawing up a plan, then building the project, you start with a set of constraints, and get the software to generate the form that you need.

"In the past, if you've designed, let's say, a vehicle rim, if you're going to design one of those, in the very far past, you had to basically sketch it by hand", says Erik Glaser, Principal Product Designer, Volkswagen Group of America. "You had a draughtsman who had to draw the whole thing out – very long process, very tedious. ↗



With generative design, instead of manually designing the actual shape, you give the software constraints. I need it to be this big; I need it to support this much weight in these kinds of directions. You feed it some physics information, basically."

Generative design is something that's only just making its way into mainstream manufacturing, but it's all around us in nature. That's why the forms produced here and in other generative designs look so organic. Termite mounds and beehives are great examples of common structures built to a requirement, or if you like to think bigger, H.R. Giger's designs for the *Alien* films.

Paul Sohi, Fusion 360 evangelist and iconic projects lead at Autodesk, says of the VW collaboration: "From a design point of view, generative design enables you to create really, really beautiful objects... On

top of that, nature is the best designer, and we end up with parts that look a lot like biomimicry. So we know that we're able to produce something that's beautiful and is not going to compromise the performance, either."

"If you speak to any designer or engineer, they'll tell you that producing stuff in the real world always takes longer than you think. One of the great things about using generative design here is we were able to go from ideas in our head to a fully manufactured vehicle in about six months, which is unprecedented and unheard of. Being able to do that while maintaining design and engineering is incredible, and it enables us to produce something that's really quite beautiful to look at." □

↗ [hsmag.cc/aIRKHB](http://hsmag.cc/aIRKHB)



# Meet The Maker: Michael Dales

The person behind the custom-built guitars



**W**

**hen the gods handed the first guitar down from Mount Olympus, humanity was given a mighty gift.** When you pick up a guitar, you're part of a global lineage that includes

Hendrix, Mick Ronson, Nancy Wilson, and any number of musicians from every continent.

We doff our caps therefore to Michael Dales, who for the last three years has been making guitars under the name Electric Flapjack. We spoke to Michael to find out how working in a makerspace has shaped his attitude to the noble craft of making guitars, his tools, his attitudes, and why it's good to share.

"The first guitar I made was just out of parts. I hadn't done any woodwork since high school, which was a long time for me. I took up the guitar again, but the guitar I had didn't really do it for me any more. I didn't want to get rid of it because it had sentimental value, so I bought parts and then kind of tweaked them. My brother, who is in a band, asked me to make him one – that was guitar number two – and I made the body for that one, which was a big leap forward.

"Since then, I've done a little bit more each time. Three years and nine guitars later, here I am.

"I'm a software engineer by trade, so I have no abilities in the real world – that's why the location from which I do all of this has been hugely important to me. It's a community workshop, and everyone focuses on the workshop side; there's lots of shiny toys and stuff – but equally as important is the community aspect. Because there are people around

who don't do exactly what you're trying to do, but these are the kind of people who will say 'let me help you'.

"There's all that experience you can tap into, so I wouldn't have been able to do this without a place like Makerspace.

## SOFTWARE TO SOFTWOOD

"[In my day job] I've done bits of hardware; I've done digital circuit designs, FPGA stuff. And I've done a lot more software in my career than hardware. That's also why I enjoy woodwork so much, because it exercises a completely different part of my brain. It's still problem-solving, but it's very different.

"You kind of find with places like this, that you have a general set of tools available to the populace, but [if you try] anything niche you end up by yourself. The makerspace isn't here to cater for luthier [guitar making] requirements. That comes from in part just getting into enabling me to have the confidence to try things.

"I watched a lot of YouTube videos, and the luthier community is very good at sharing; there are lots of great tutorials by professional luthiers on how they do stuff.

"I don't have all the luthier tools, but I have identified that there are some that are quite important; they'll save you a lot of time and effort. The Japanese saw for doing the fret slots is a good example of that. I have a specific requirement of that saw, because it had to be narrow enough that when I put a fret in it'll stick, so it had to be a narrow blade and quite sharp. That was one where it was working for money, which enabled me to do that. ➤



**Above** Two of Michael's takes on the classic Fender Telecaster design



**Above** It's nice having access to a laser cutter

"The first year was fine, but then everything went wrong. My first commission was late – I said it would take three months, but it took me a year. The CNC router stopped working, so I had to learn how to do it by hand. Like when any automation breaks down, it exposes whether you understood the thing you were automating or not.

"It's been an interesting journey. Thankfully the guy who commissioned the guitar was very accepting; he was happy to be part of that journey. I kept him involved as much as I could, and it worked out well.

### MAKERSPACE WORKING

"Being in a makerspace, you have tools available that traditional luthiers don't. And this is the counter to the fact that I don't have all the tools that a traditional luthier would have. So, for example, I have access to a laser cutter.

"One of the bits you really have to get right, otherwise it's not a musical instrument, is the fret spacing. Those have to be right, because otherwise, it's not going to sound correct. So, being a software engineer and having access to a laser cutter, I wrote a small tool that lets me type in '22 frets, 25 1/2 inch scale length', and it will etch a design.

"I still have to cut the slots by hand – the kerfing on the laser cutter means that you get a V-shape

rather than a straight slot – but I can use the tools around me to my advantage. The nice thing is that I know that other people use that tool. Even if they don't have a laser cutter, that's how a whole bunch of people generate their fretboards now, using that little tool that I made. It's nice to be able to give back to the luthier community, which has given me so much.

Being in a maker space, you have tools available that traditional luthiers don't

"It's fun being in an environment like a makerspace, where you don't have everything you need, but you've got more stuff. 'How can I solve my problem using the tools around me? If there are better tools, how do I find someone who can point me in the right direction?'

### ART & SCIENCE

"Necks are totally under-appreciated. When you look at a guitar, you tend to look at the body and the



pickups and the controls. Those are the eye-catching bits. But all the labour and effort is really in the neck – if you don't get that right, it's not a musical instrument. It's, at best, a bit of art.

"On the second guitar, I did the fretwork, which I shouldn't have done, because that made the project run over way more than it should have done. I broke my own rule and made too big a jump. I bought a neck without frets, and I ended up having to do the frets three times before I got it right.

"Guitars three and four were built together, and I'm about to replace the neck on guitar three. It was an OK first effort, but I'm a slightly better guitarist than I was two years ago, and it now annoys me.

"A guitar has to speak to you in many ways. We are emotional beings, and our relationship with our instruments is partly based on how we feel when we look at them. I care about getting the details right so that the person who gets it has that sense of pride. I want them to go on stage. What makes me happiest is when people are on stage playing a guitar that I built. That's the most awesome thing in the world. I went to one of my brother's gigs, and he was there on the stage playing a guitar that I'd made. In some small way I was responsible for the audience having a good night. That's an amazing sensation – nerve-racking as well, because if the guitar breaks, it's all my fault.

### THE QUEST CONTINUES

"Ultimately I'd love to do archtops. I think those jazz-style guitars are amazing. But it's just a journey. I started with a Fender Telecaster, because it's the simplest, dumbest guitar. It's the original solid-body guitar, right? It's a great starting place for someone who's got no experience."

"I'd always had a Telecaster so I knew what it should feel like. And on each one, I try to do a bit more. The builds I've got going at the moment, one of them is the first one where the body design is my own. And it's got some features from a Les Paul – it's got the electronics cavities in the rear. That, to me, is a very sensible design decision. Whereas the fact that you have to take the strings off to do anything with the electronics on a Fender design just strikes me as insane.

"What I want to get to is making guitars that people like, in the shapes that they like, and I bring along the engineering elements that make them a good instrument." □

**Above** Michael makes his guitars at Cambridge Makerspace, benefiting from the shared knowledge there as much as from the tools



# UFOs

Spring – er, autumn – clean your workshop shelves



**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, and is Maker-in-Chief for the Guild of Makers: [guildofmakers.org](http://guildofmakers.org)

**H**ow many UFOs can you see in your workshop? Unfinished objects (UFOs) were taking up too much of my mental and physical space, so I made time this summer to either finish them or bin them.

There are many reasons I have unfinished objects – maybe I got stuck, didn't have the right tool or part, or more pressing (paid) work got in the way. Or, more likely, when things are tidied away and out of sight, they lose their importance.

Rummaging through the unmarked brown cardboard boxes, I discovered that some projects still made me smile. For example, the one that hit me with a soapy, metallic, creamy sort of smell – the smell of brass polish. It reminds me of my childhood – polishing the brass items was my job, and I always had to have a shiny Girl Guide badge. In the box was an old carriage lamp. I'd wanted to make it into a table lamp. I'd taken it apart and cleaned it – hence the smell. But it still needed a new light fitting and switch installed, and a stand making for it. The wood for the stand was lurking in the workshop, hidden under a pile of other wood, the new light fitting and cable was

in the box – but I was missing a switch – so I ordered one off the internet, and then spent a happy few hours putting all the bits together – wondering why I had ever shelved the project.

Other projects I wondered why I had ever started, such as the electroluminescent wire bow-tie. Some projects had been superseded by technological advancements or by more affordable commercial products – like a video doorbell. Or my life had changed – I no longer own ducks, so the duck egg incubator project also got cancelled.

I reclaimed, or recycled, all the parts and crossed them off my mental list.

Some of my UFOs are books that I either haven't started or finished reading.

It's always a pleasure to give myself permission to read a book. I binge read, so once I'm in, I'm in for a good while. This summer I've enjoyed Adam Savage's *Every Tool's a Hammer* and Emilie Wapnick's *How to Be Everything* – as well as Ursula K. Le Guin's *Earthsea* series. I find fiction stories, especially science fiction or fantasy, always recharge my creativity.

Now I also have a lovely table lamp, more parts in my parts boxes, more shelf space in the workshop. And a slightly less nagging 'you should finish that before you start a new project' feeling. □

# The incredible shrinking chip package

Squeezing more sand into your computers



**Bunnie Huang**

 @bunniestudios

Andrew ‘Bunnie’ Huang is a hacker by night, entrepreneur by day, and writer by procrastination. He’s a co-founder of Chibitronics, troublemaker-at-large for the MIT Media Lab, and a mentor for HAX in Shenzhen.

The vast majority of chips today consist of a single fleck of silicon, usually in some sort of a package to protect the chip and make it easier to mount to a circuit board. In the 1980s, silicon was wire bonded into DIP lead frames; in the 1990s, the industry transitioned to surface-mount flat-packs and ball grid arrays; and in the new millennium, chip-scale packages rose to dominance. Regardless of how the chips came, the rule was typically one chip per package, with the main exceptions being flash memory chips which could be easily stacked using conventional wire bonding techniques, and a few mobile phone CPUs that could afford exotic processes.

This may soon no longer be the rule. In what appears to be a hedge against the end of Moore’s Law, foundries are investing heavily in next-generation packaging technology that allows a single ‘chip’ to contain multiple silicon die. AMD’s Ryzen ‘chiplet’ architecture and graphics chips using stacked HBM are just a hint of what’s to come. TSMC recently revealed a buffet of mouth-watering packaging options that will blur the distinction between individual silicon chips, and the package containing them. Three major trends – through-silicon vias, integrated fan-out, and chip-on-wafer-on-substrate – are allowing

designers to mix and match chips from processes optimised for logic, RF, analogue, and memory into a single, high-performance package no larger than your traditional single-chip package. The new packaging technology is also disruptive because it improves I/O density and on-chip performance. Increased I/O density means more I/O bandwidth, which is good news for today’s memory bandwidth-bound CPUs.

The on-chip performance advantage isn’t quite as obvious, but the performance of chips isn’t just constrained by how fast transistors can switch on and off. It’s also

constrained by how much power you can get into and out of the package. Desktop CPUs can draw over 100 amperes of current (the typical wall outlet in your house is limited to about 10–20 amperes, albeit at a much

## AMD’s Ryzen ‘chiplet’ architecture and graphics chips using stacked HBM are just a hint of what’s to come

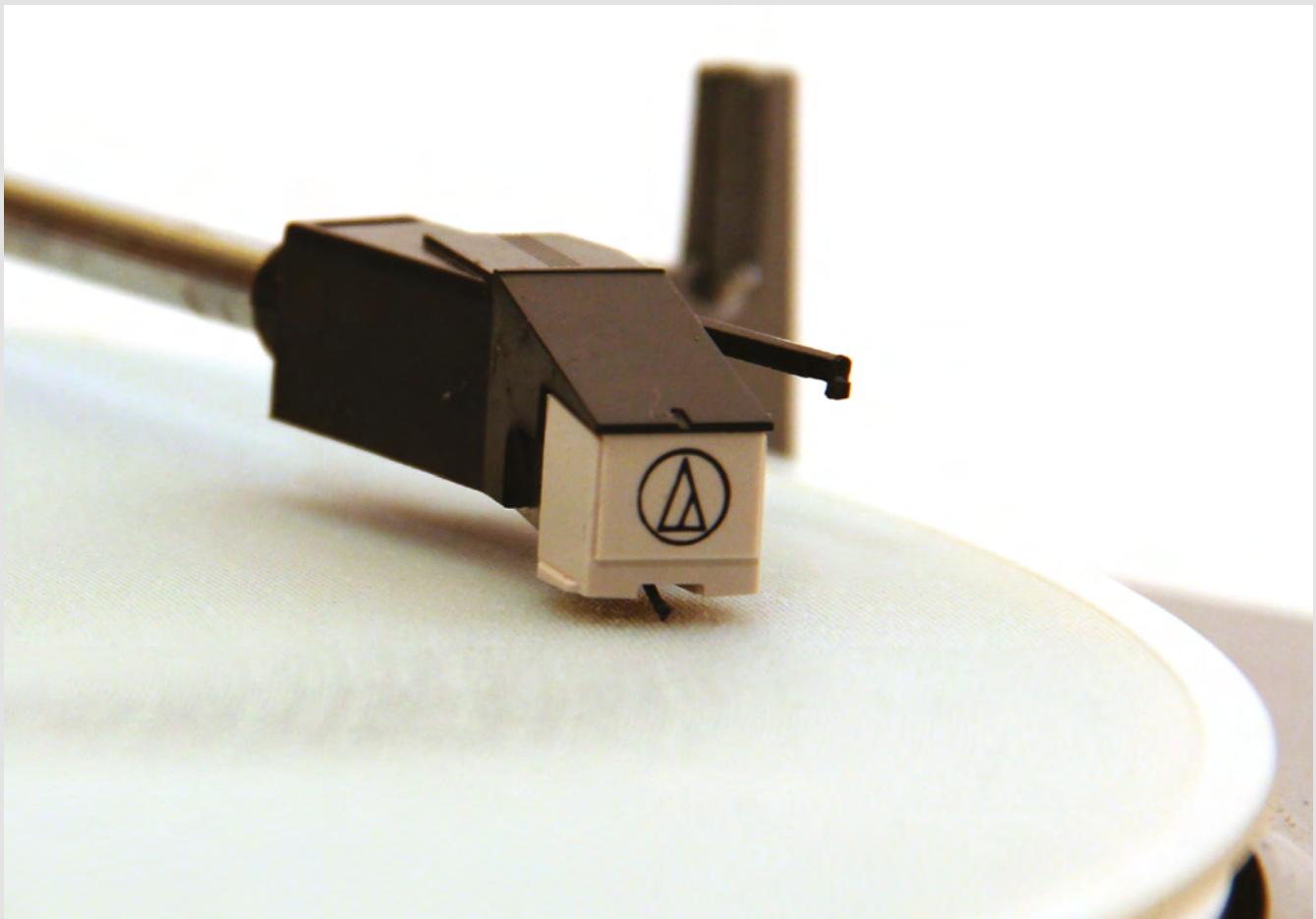
higher voltage). At these phenomenal ampacities, even tiny imperfections introduced by the package will make the CPU unstable. The latest round of packaging technology promises to push up the power ceiling by an amount that will equate to at least one generation of Moore’s Law improvement in performance.

The slowing of Moore’s Law may be bad for increased transistor densities on a single chip, but it also marks the beginning of a new race to improve the incredible shrinking chip package. □

# 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](http://hsmag.cc/hello)



### 3D PRINTING

The 3D-printed record seemed like the stuff of science fiction. I can almost imagine the record slowly rising out of the goop of UV-cured resin in the printer. Imagine showing that to Thomas Edison whose phonograph was a precursor to the vinyl record? It would

have blown his mind! Imagine what weird technologies await our children. I play with a 3D printer a little, but I'm not that confident with it. Can we have more 3D printer content, please?

Jane  
Brighton

**Ben Says:** I, for one, can't wait to see what the future holds. The ingenuity of humans knows no bounds. More 3D printing content you say? We might have a thing or two up our sleeves. Keep your eyes peeled over the next few months, and hopefully, you'll be pleasantly surprised.

## REPAIR

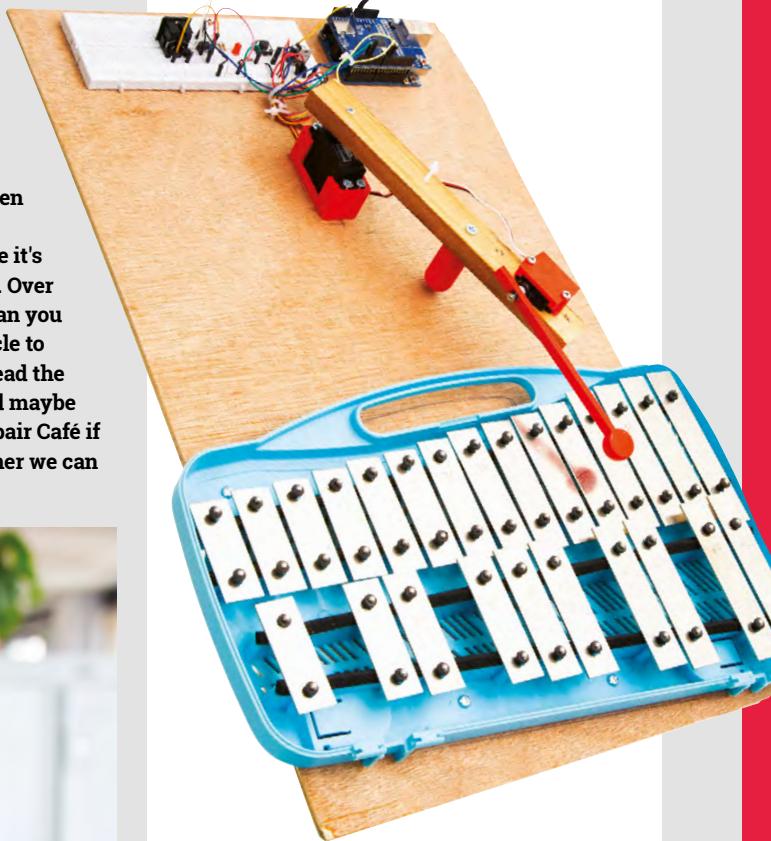
I really enjoyed reading about the Restart Project in HackSpace 23. It's frankly embarrassing how much stuff we throw away, especially as we claim to be a civilised society. Surely any civilised society has to take some responsibility for the planet they're living on, and not just keep dumping waste into the ground and carbon dioxide into the air? Let all of us makers make a stand and say enough! We're not going to buy new trinkets, but keep our old ones running.

**Roger**

Derbyshire



**Ben Says:** I heartily agree, Roger. There's no point in just waiting for someone else to take responsibility, because it's becoming clear that politicians and businessmen aren't going to solve this problem for us. It looks like it's up to us to save the planet. Over to you, dear reader, what can you repurpose, repair, or upcycle to keep it out of landfill? Spread the word, tell your friends, and maybe offer to help your local Repair Café if you have the skills. Together we can make a difference.



## MUSIC

Blimey, Matt Bradshaw's Glockenspiel looks ace! I enjoy music, but just never seemed to get the co-ordination to play an instrument (or perhaps I was just never dedicated enough to spend the time necessary to learn). I've spent the last 20 years as a programmer though, so hopefully I've got the technical skills to get a robot to work. Not sure a Glockenspiel is quite up my street though, but I'm going to get going on a robot drum kit. Thanks for the inspiration!

**Paul**

Minnesota

**Ben Says:** Fantastic news. Of course, not everyone wants to copy projects verbatim (in fact, we suspect very few people do), but the great thing about projects you make yourself is they can be endlessly tailored. Drum kit, glockenspiel, piano – it doesn't matter to the servos or linear actuators doing the playing.

# CROWDFUNDING NOW

## POWERUP 4.0

Give your paper planes an electric boost

From \$59 | [Kickstarter.com](#) | Delivery: April 2020

**W**e've looked at POWERUP's electrified paper planes before. In Issue 11, we crash-landed their DART in our neighbour's garden, and had great fun. In short, it's a motor-and-controller unit for paper planes that links to your phone via Bluetooth, and lets you zoom around your local park [Ed – Make sure you comply with local regulations and are respectful of other people in the vicinity].

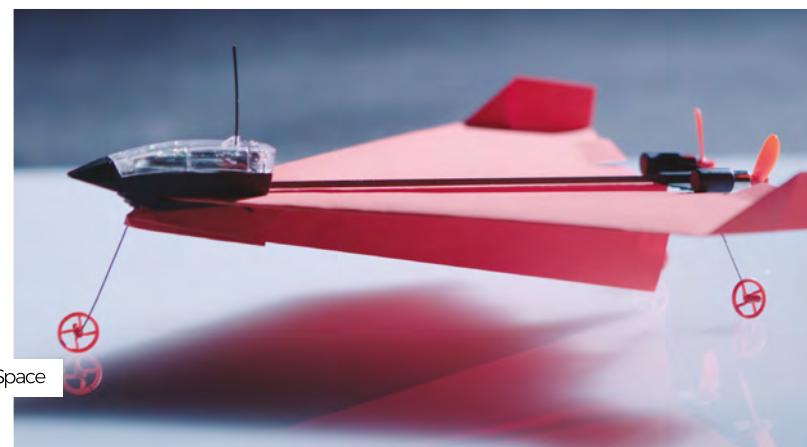
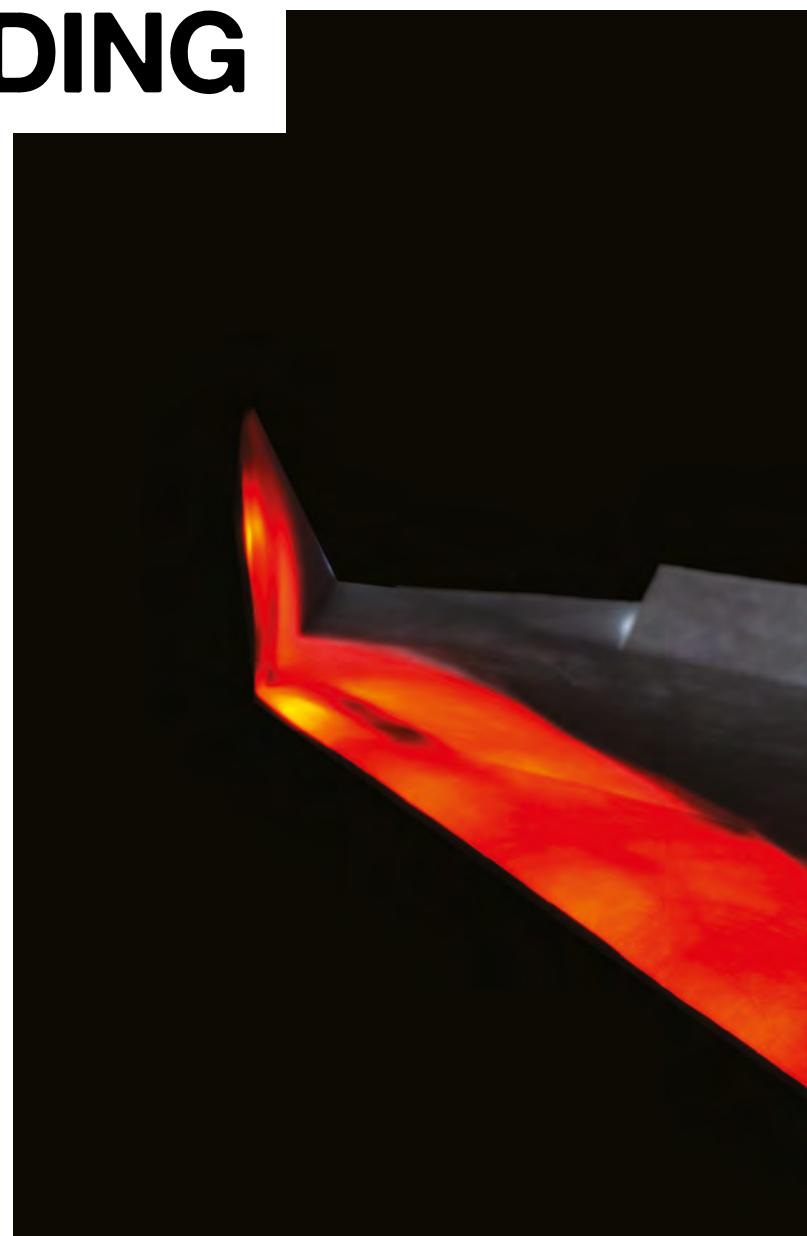
Well, they're back with the POWERUP 4.0, with a couple of new features: extra sensors and an extra motor. The extra sensors are in the form of a gyroscope and an accelerometer. These two give you far more control over the plane than on previous versions (hopefully this will mean that our neighbour's veg patch is spared a repeat of its aerial assault).

The extra motor gives more thrust, and that means that you can use a wider range of materials to make your planes from – unlike earlier models that required paper, you can now use foam, balsa wood, or almost anything else, provided the airframe is under 20 grams.

There are also some improvements to the robustness, with a carbon fibre frame and rubber bumper on the front.

The POWERUP 4.0 – like previous versions – isn't the most flexible way of making a plane. There's only one controller and, (while you can download flight data for later processing), no way of interacting with a flying plane from your own software. The unit is also fixed in one shape, so your plane has to fit around this.

However, what you lose in flexibility, you gain in ease of use. This is by far the easiest way of making a powered plane. It's fantastic fun to transform folded paper into a flying machine that you can control from your phone. If you develop your planes to the point that they hit the limits of the POWERUP 4.0, then you'll have to move on to more powerful hardware, and tackle the complexities that come with that. □



---

## 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.



The extra motor gives more thrust and that means that you can use a wider range of materials



# International Rocket Week 2019

The sky's the limit



**Above** A small scratch-built rocket called VR2 launching off the pad

**Right** A young rocketeer retrieves the Estes 'Baby Bertha' rocket he had built and flown

**Below** Peter Waddington's scratch-built, upscaled version of a smaller 'Estes Quinstar' spotted in the workshop with a HackSpace magazine sticker on it

**International Rocket Week (IRW) is a fantastic annual event that has just held its 34th year.** It's a week where the amateur rocketry community from the UK and beyond come together to build and fly an eclectic mixture of model, high-power, experimental, and research rockets. It's split between two sites. 'Base Camp' at Lapwing Lodge provides the accommodation and workshop areas which are a hive of activity from 9am until the wee hours of the morning, as people continue to tweak and build their projects. Fairlie Moor is the launch site, which is around 30 minutes drive away from Base Camp, and on decent weather days, this is the sanctioned flying site. With all launch equipment set up and a NOTAM (notice to airmen) submitted to air traffic control, this flying site can be busy! In one day alone, 55 rockets were launched, ranging from small model rockets to a huge M-class rocket that flew to 18,000 feet.

## DUE NORTH

The IRW is run by the brilliant Scottish Aeronautics and Rocketry Association (SARA), and is affiliated to the UK Rocketry Association (UKRA) who maintain the safety codes, the certification scheme

for high-power rocketry, and train range safety officers (RSOs). All flyers are also members of the British Model Flying Association, who provide the insurance (provided people are adhering to the UKRA safety code).

There are all kinds of competitions at IRW including landing a rocket, duration flights for boost gliders (a glider released from a rocket), helicopter recovery duration contests where they release rotating blades on the way back down, and a really interesting 'confectionery' competition where a



rocket either has to contain sweets, or be largely built from sweets or sweet containers!

### MEETING PEOPLE

The workshop areas are fabulous to work in and hang out with people helping each other and sharing skills. The range of tools that people bring to IRW is fascinating; people tend to have made all kinds of tooling and jigs to help with rocket builds, and it's really interesting to see and discuss. This year, as well as the usual hand tools, we saw 3D printers, a small band-saw, disc and belt sanders, and much more – it's very well equipped, and everyone shares what they have. People work on all manner of related projects, so it's not uncommon to see someone sewing a parachute or soldering avionics or spray-painting an airframe – this year, a team were working on a scratch-built, steerable GPS parawing to try and be able to return rockets close to the launch site. At Lapwing Lodge there's a friendly on-site vendor, 'Rockets and Things', with all manner of kits, spare parts, and materials to get you flying.

### RETURN TO EARTH

The lander competition was won in spectacular form this year by Sharon Calver, with her fabulous rocket that splits into two sections during flight and then deploys the landing gear. It was a complex build – following a spectacular failure the previous year, it had been meticulously rebuilt by Sharon. It landed a few hundred metres from the launch position with the three lander legs fully deployed in a textbook landing, earning her a well-deserved win.



**Left ♦**  
A collection of larger rockets in one of the workshop areas; there are so many rockets to look at and learn from at Base Camp

**Below ♦**  
Dr Jonathan Rhodes working on a section of his high-power rocket in one of the IRW workshop areas

It's not **uncommon** to see someone sewing a parachute or soldering avionics

In essence, IRW is a large group of rocket-obsessed makers who set up an impromptu hackspace with a focus on building all manner of flying things. It's a great event in two perfect locations – we hope to see you all there next year! Many thanks to the IRW team for hosting us. Check out their website for details of next year's IRW here [irw.rocketry.org.uk](http://irw.rocketry.org.uk) □



# Space of the month: Radiona



Radiona

- [↗ radiona.org](http://radiona.org)
- [ƒ Radiona.org](https://www.facebook.com/Radiona.org)
- [@RadionaOrg](https://twitter.com/RadionaOrg)
- [⌚ RadionaOrg](https://www.instagram.com/RadionaOrg)

**R**adiona.org is a Zagreb-based makerspace founded in 2011 in order to enhance the visibility of open-source culture, with the aim of connecting STEAM fields (Science/Technology/Engineering/

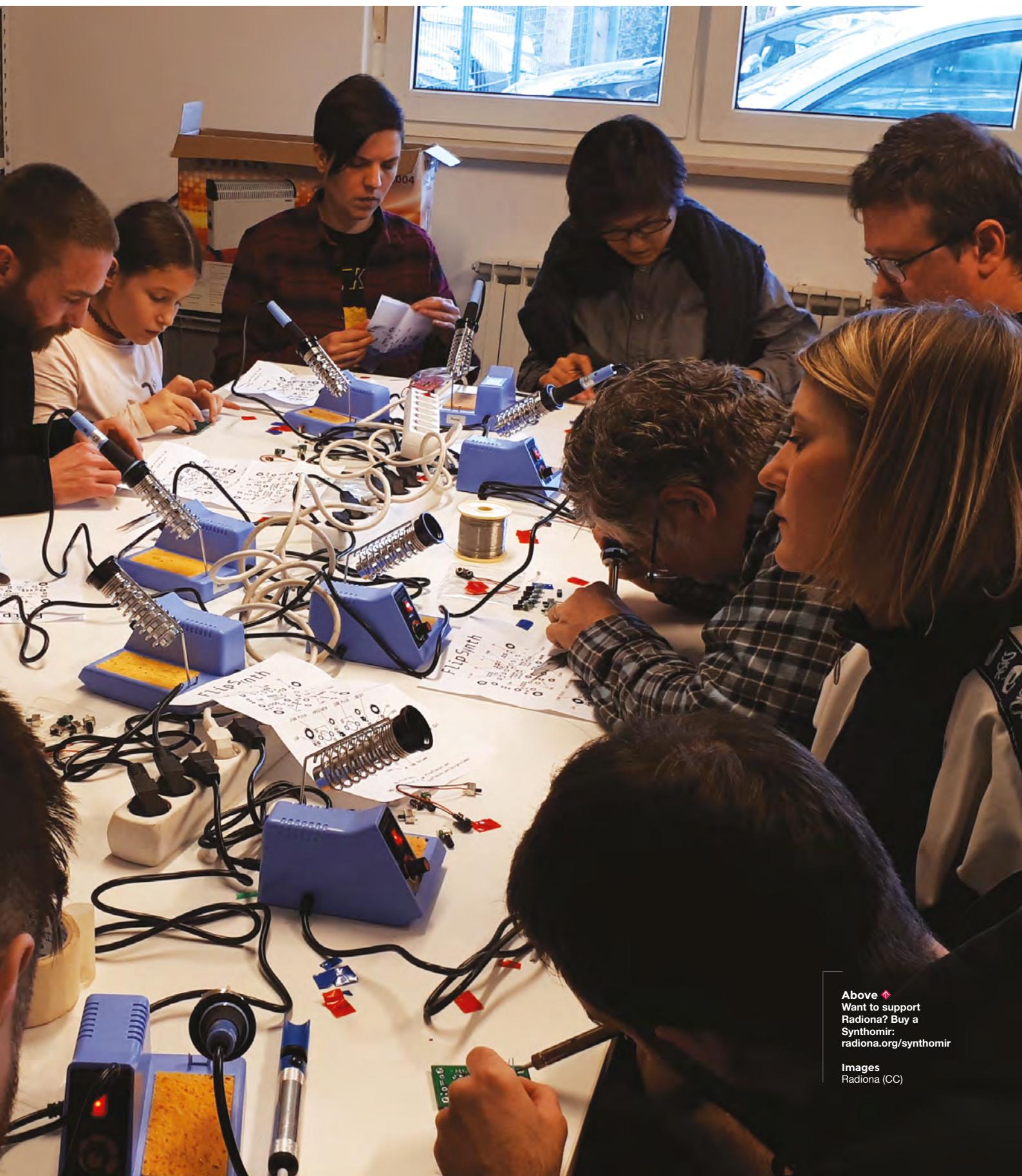
#### Art/Mathematics).

The lab focuses its activities on workshops, exhibitions, research projects, hardware production, projects with artists and designers, maker-in-residence programmes, international collaborations, maker/hacker camps and hackathons, social innovation projects, festivals, and many more. Its community is in the centre, as a main driving force for all activities that include production of new kits and technology solutions, artefacts for exhibitions and museums, education programmes with a cross-section of cutting-edge, open-source hardware and software lecturers, and a combination of high-tech and low-tech approaches to the maker/hacker culture.

Radiona has organised more than 180 international and domestic workshops, 26 international exhibitions, nine hackathons, two international festivals, numerous performances and lectures, and many more events. There are diverse sections and accompanying tools and machines for everyone interested in making things: microcontrollers and hands-on electronics, robotics and hacking, programming, biohacking, audio editing, restoration of vintage electronic organs and modular A/V synthesizers, FPGA, 3D printing, CNC, 3D scanning, e-textiles, interaction design, design thinking and gamification, IoT and smart city technology – LoRaVan, programming, flying objects, etc.

The lab is fostering innovative approaches based on hands-on practice and lifelong learning on sustainable foundations, inclusion, and participative practice driven by access to all philosophy. Radiona is taking part in several international projects with science centres and museums to foster the use of technological solutions with creativity, successfully creating solutions for exhibitions, or hackathons like Museomix. →



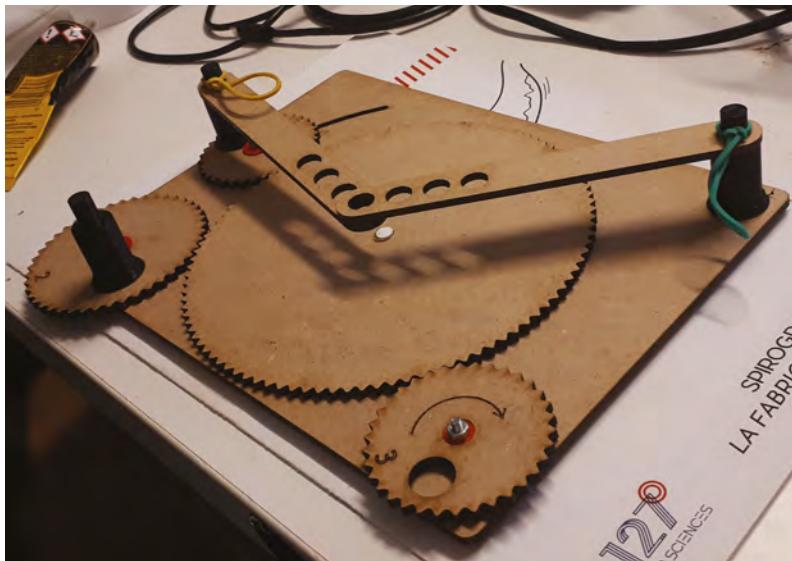


Above ↗  
Want to support  
Radiona? Buy a  
Synthomir:  
[radiona.org/synthomir](http://radiona.org/synthomir)

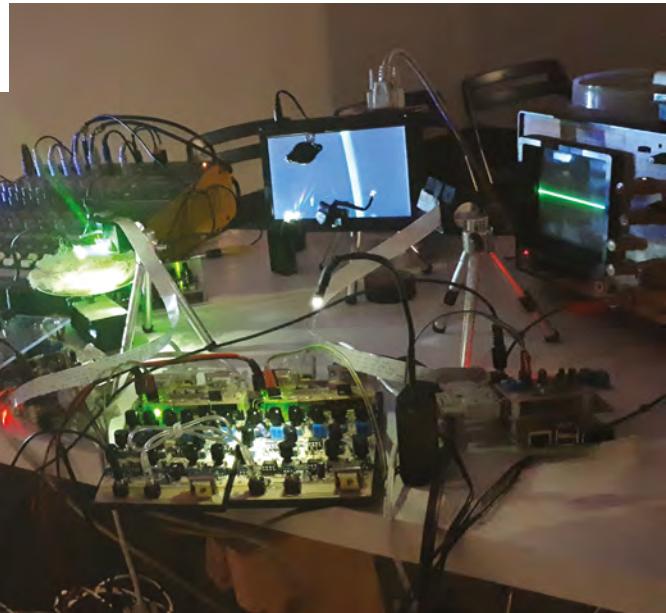
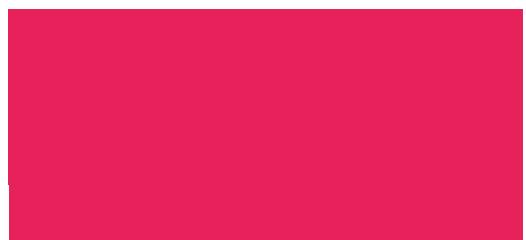
Images  
Radiona (CC)

## Space of the month

### REGULAR



The production of artworks and designing exhibitions are one of the strongest frames of the lab, with three to four yearly international exhibitions in galleries and museums across Croatia. Among many projects developed through the years, recently the lab has gained a lot of international recognition with the ULX3S powerful ECP5 board for open-source FPGA development, and has covered a big part of Zagreb with LoRaWAN using Raspberry Pi inside the gateways. One of the lab members also made an hgTerm Raspberry Pi mini-computer that gained a lot of interest on Hackaday, an online platform where Radiona members regularly take part in competitions. The education programme of the lab is one of the most recognised areas of the lab, with an impressive list of guest appearances from international media artists, makers, and hackers. If you are planning to visit Zagreb, be sure to check Radiona and its community of makers, or follow all activities via social networks: Twitter – [twitter.com/RadionaOrg](https://twitter.com/RadionaOrg), Facebook – [facebook.com/Radiona.org](https://facebook.com/Radiona.org), GitHub – [github.com/RadionaOrg](https://github.com/RadionaOrg), or [radiona.org](http://radiona.org). □



### CONTACT US

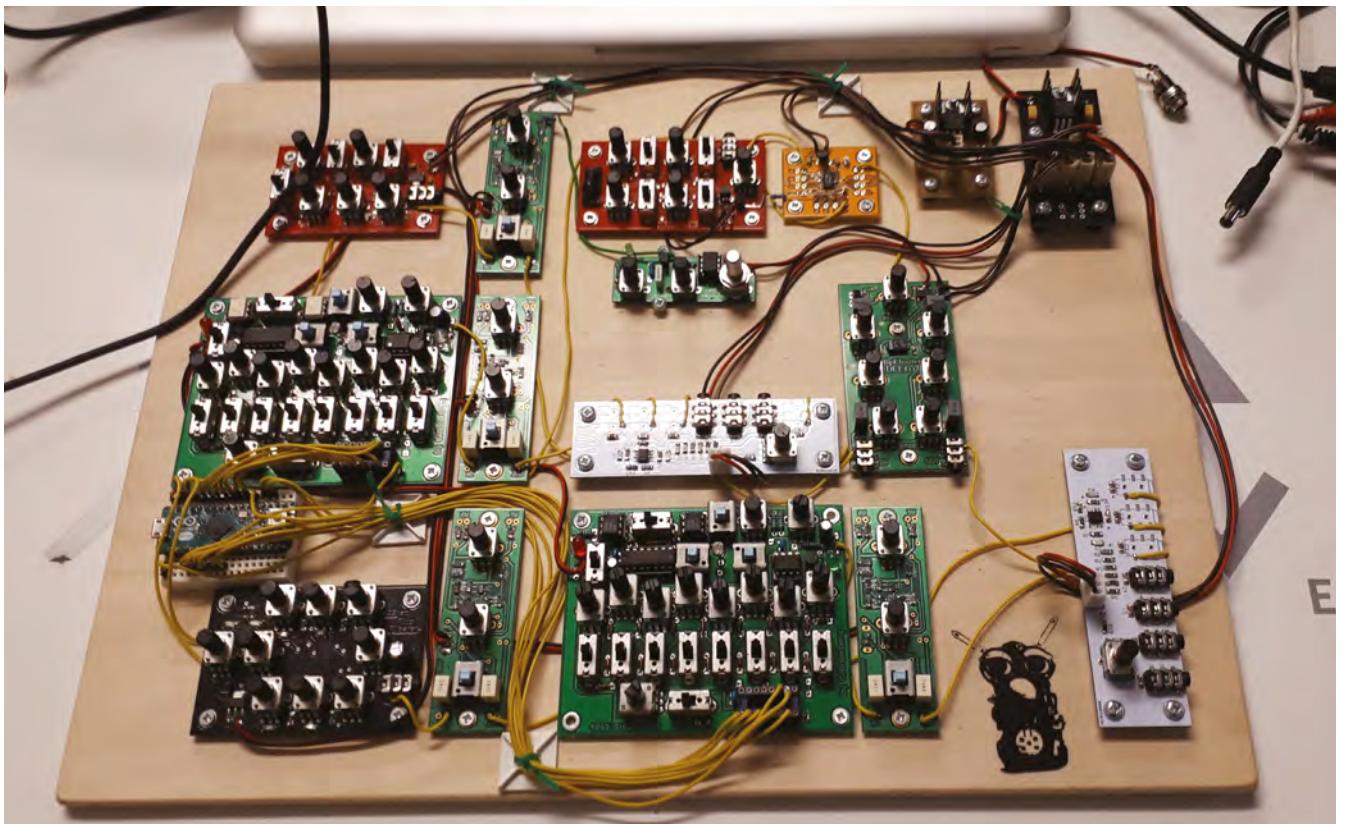
We'd love you to get in touch to showcase your makerspace and the things you're making. Drop us a line on Twitter @HackSpaceMag, or email us at [hackspace@raspberrypi.org](mailto:hackspace@raspberrypi.org) with an outline of what makes your hackspace special, and we'll take it from there.

**Below ♦**

Radiona attracts a string of international makers, among them Marc Dusseiller from Switzerland, who led a workshop making this 8-bit synth

**Left ♦**

This image was taken at an event ominously called "Brain Hack"



# Forged in foam

How a Halloween costume started a career



Krista Peryer

@missladygeek\_

Krista is a cosplayer and craftswoman with a list of ongoing projects longer than Daenerys' titles on Game of Thrones.

**C**osplay, the word we know to be defined as dressing up in a costume to portray a character from a comic, video game, anime, manga, book, TV show, or movie, is still a relatively new term used in the broader scope of time when relating to just how long people have been costuming as their favourite literary and film characters. In a short history, the first time the term 'cosplay' was used was in 1984 when the Japanese reporter, Nobuyuki Takahashi, frankenworded 'costume' and 'play' to explain his experience while attending Worldcon in Los Angeles. Takahashi thought that translating 'masquerade', the traditionally oft-used word to describe costuming, felt too old-fashioned to translate to his modern Japanese audience, and thus, 'cosplaying' was born.

Although the term 'cosplay' didn't make its debut until the 1980s, cosplay has existed since the early 1900s. The first documented instance was in 1908, when Mr and Mrs William Fell of Cincinnati, Ohio, attended a masquerade ball dressed as Mr Skygack and Miss Pickles, Martians from a newspaper comic

He and his father worked together to create the costume, effectively igniting an enthusiastic passion for cosplaying

that first ran in the Chicago Day Book. Forrest J. Ackerman, a sci-fi writer and fan, became one of the first sci-fi cosplayers when he was the first person to attend a sci-fi convention, New York's first World Science Fiction Convention, Worldcon, wearing what was dubbed a 'futuristic costume' (another fun frankenword!) complete with a cape and tights, in 1939. The outfit, designed by his girlfriend Myrtle R. Douglas (better known as Morojo), sparked interest in fan costuming, or cosplay as it is now known, and



Above ⤵  
Joining the collective

sci-fi conventions began to look like masquerade balls, complete with costume contests for 'best costume'. Ackerman continued to support and cosplay his beloved sci-fi, and eventually became a key figure in establishing sci-fi fandom. In the years since, many more sci-fi conventions have developed worldwide, with legions of fans attending dressed in cosplays made from a variety of materials to showcase fandom of their favourite characters from popular modern and classic sci-fi such as *Star Trek*, *Doctor Who*, and *Star Wars*. Bruce Holt is one of them.

## PLAYING WITH COSTUMES

A cosplayer, master craftsman, and teacher specialising in intricate EVA foam cosplay creations, Holt remembers the first time he heard the term 'cosplay'. It was in the 2000s, and he was in Tulsa at Hard Rock Casino for a costume contest. Wearing a *Star Trek* Borg costume that he'd made out of plastic and leather, another attendee approached him to ask if he'd made his cosplay out of foam. Holt had not only never heard the term 'cosplay' yet, (even though he'd already been hired to guest host panels on costumes and makeup at this point in his costuming career), he'd never even considered using foam



**Left ♦**  
Warning: this shield  
will not protect you

**Below ♦**  
Two Wonder Woman  
breast plates ready  
for cosplay



as a material for costume creation. Although the Borg costume was what had initially put cosplaying regularly, especially competitively, on the map for Holt, he had been making costumes since childhood.

### SMALL BEGINNINGS

Growing up in southern Utah, his parents divorced when he was nine years old. His father subsequently moved to California, got remarried to a seamstress, and together Holt's father and his wife created costumes for local school productions and events, such as plays and cheerleading. Holt mentions that he'd never had much interest in material costuming, until one year while spending summer with his father at the age of 14, he decided to ask his father for help making an elaborate Halloween costume. They worked together to create the costume, effectively igniting an enthusiastic passion for cosplaying that set Holt on a fast track that eventually became the cosplaying business he now owns and operates – Crazy Costumes.

### LEARNING THE TRADE

That first costume he and his father made together was 'a reverse man', as Holt recalls. 'Reverse Man' was such a huge hit at Halloween that year; each year after he strove to make more elaborate costumes out of any materials he could get his hands on, which at

the time consisted of leather, plastic, and fabric. He joined his school's theatre group and participated every year to increase his ability as a maker. As he grew older and received more and more positive feedback for his makes, he began dreaming of moving to California permanently to make costumes all the time. However, in 1979, after meeting his wife of 40 years now, Joyce, he was set on a whole new, wonderfully different trajectory when they married and settled in Cassville, Missouri. →



**Left ♦**  
One of Bruce Holt's  
early costumes

## REGULAR



Despite moving halfway across the country, Holt continued his love for costuming, travelling to sci-fi conventions wearing the *Star Trek* Borg costume that he'd continued to wear throughout the 2000s, and attending Halloween parties. It wasn't until some time in the 1980s, after being approached at a Halloween party while wearing the Borg costume, that Holt realised there was money to be made in costuming, opening up an incredible door of new

opportunities. Holt was told that his Borg costume was so incredible that he should consider entering into a local bar's Halloween costume contest the next year, and they give away cash prizes for contest winners. Holt entered the contest the next year with his Borg costume, and won. He states that when he realised there were prizes to be won, money to be won, he began to get more creative with his craft. One of his bigger and more elaborate cosplays was a *Star Trek* Klingon. Holt had honed his skills quite a bit more by that point, and the facial prosthetics were praised as looking 'movie quality'.

### BECOMING A CRAFTSMAN

Since then, Holt has gone from a novice costumer making costumes of plastic, leather, and fabric, to a master EVA foam craftsman and cosplayer with numerous awards, conventions, cosplays, and happy students to show for it. Holt states that while he loves cosplaying, it's actually his love of teaching that keeps him going and wanting to learn more so he can share his love and knowledge for cosplaying with communities throughout the world. Holt travels frequently, attending three to five conventions a year to showcase his makes, act as a guest panellist, and hosts workshops to teach EVA foam cosplay creation. Creations for his workshop business called 'Forged in Foam', range from making a shield over two days (three hours per day) to something as big as making a *Starship Troopers* costume over a

**Right ↗**  
Bruce has come a long way

**Below ♦**  
Protecting humanity from bugs in style





weekend like he recently did in 2018 with 18 students at Visioncon in Branson, MO. Holt recalls that this was an extremely exciting workshop for students because Casper Van Dien of *Starship Troopers* was a celebrity guest on the roster at

**When he realised there  
were prizes to be won,  
money to be won, he began  
to get more creative with  
his craft**

Visioncon that year. After completion of the workshop, the students were surprised with the opportunity to do a photo shoot with Van Dien wearing their costumes – Van Dien also wore a special costume designed by Holt himself.

### FINAL FORM?

Some of his present, much more evolved, fantastical EVA foam cosplays now consist of Morphysses from World of Warcraft, Wonder Woman, Odin, and

Lady Sif, as well as taking commission work for almost anything your cosplaying heart desires. It's easy to tell, by the way he lights up when talking about cosplaying, that while it's fun for him, and he'll always love cosplaying, his true passion lies in educating and growing us all to be makers of our own. □

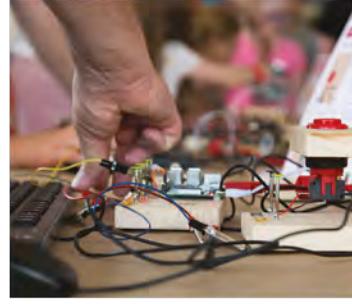
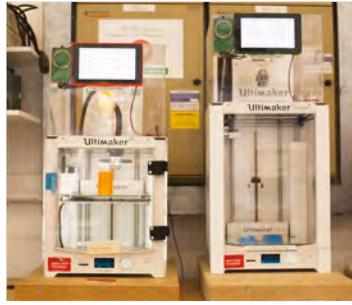


**Above** ♦  
Sif, Goddess of  
the Earth

**Below** ♦  
Odin is often thought  
to be Norse, but was  
worshipped by many  
Germanic peoples

**Left** ♦  
Inspiration can come  
from games





# 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](https://rpf.io/makerspace)



Raspberry Pi

The Raspberry Pi Foundation. UK registered charity 1129409

#RPiLearn

# LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG  
58



## HOW I MADE A SKULL SYNTHESIZER

What would you make  
with a dead raccoon?

PG  
64

## INTERVIEW: THE VOLCANOLOGISTS

We meet the people hacking  
Raspberry Pi cameras for science



PG  
72

## IMPROVISER'S TOOLBOX: SOCKS

Clear space in your drawers  
for a new season's woolies



PG  
42

# HALLOWEEN BUILDS

SPOOKY, SCARY, AND UNCANNY  
CREATIONS TO MAKE  
DR FRANKENSTEIN PROUD

BY  
MAYANK  
SHARMA

S

ufficiently advanced technology – as Arthur C. Clarke once said – is indistinguishable from magic. On the most magical night of the year, sufficiently advanced technology has to stand in for those of us not naturally imbued with magical tendencies. The technology doesn't have to be particularly advanced (though it can be), it just has to be deployed in unusual and unfamiliar ways, and this is what we as makers are particularly adept at.

There's a huge range of hardware options for creating unique looks and embedding light, sounds, and movement in your costumes. You can use almost any platform and fabrication techniques you feel comfortable with to create your own outfit. This is a chance to have fun, push the limits of your knowledge, and show off your skills to friends and family. There are opportunities at every skill level, from adding a few lights or 3D-printed parts, to huge custom-built costumes – you just have to decide what fits your time, budget, and skill level, then get making.

When this magazine first appears on the shelves, there'll be 42 days to get your outfit together. If you've already started, then great. Perhaps the next few pages will give you some ideas on how to enhance your get-up. If you're still planning, don't worry; there's still time. Let's take a look at what some other makers have made to give you some inspiration. Get making and get spooky! →



# LET THERE BE LIGHT

GLOW, BLINK, AND ILLUMINATE YOUR WAY TO THE PARTY

**Erin says you can optionally add a small cabochon over the HalloWing's screen to give the animation some dimension and shape**

**L**ights are an important part of the Halloween festivities. The annual holiday can be traced back to the ancient Celtic festival of Samhain, where people would light bonfires and don costumes to ward off evil spirits and ghosts.

Erin St. Blaine is a fashion and LED artist and has combined the two to create glowing mirror masks ([hsmag.cc/wWAoSh](http://hsmag.cc/wWAoSh)). The tutorial covers two designs, but Erin says the technique can be used on masks of any shape and size. The main

electronics behind the mask are the Adafruit HalloWing and NeoPixel LED strips. The mask shines with the help of a flexible one-way mirror film that you can buy online. Erin also tells you how to make your own to help keep the costs down. The best thing about the project is that you can do it without any soldering. Erin used a needle and thread to stitch the NeoPixels around the outer edge of the mask. The HalloWing adds another unique dimension to the mask, but you can skip it to simplify the build.

If you want a mask that's a bit more sinister, Noe Ruiz has just the thing for you. His sci-fi inspired 3D-printed gas mask ([hsmag.cc/YLpHdz](http://hsmag.cc/YLpHdz)) will surely turn heads on Halloween. The project has three main components: a mask, a respirator, and the goggles. As you would have guessed, the respirator glows thanks to a couple of NeoPixel LED rings that are wired to a Gemma microcontroller. Noe recommends using silicone-coated wires that are more flexible than standard ones, and a good choice for all types of wearables. The other components of the mask are 3D-printed, and if you use Noe's design, you can just snap the electronics in place. He's also made some changes to the standard costume goggles to match the rest of the mask. Finally, an EL wire that's strung through the various components of the mask gives it a more ominous appearance.

If a mask isn't your thing, Nathan Dunbar has got another type of headgear. He's replicated the two helmets worn by Daft Punk ([hsmag.cc/0UuwLM](http://hsmag.cc/0UuwLM)).





If you only have time to build one, Nathan suggests you replicate the silver Thomas helmet, which is easier because it doesn't require vacuum forming

A major part of the build involves crafting the helmets using a combination of 3D printing, vacuum forming, and sanding. Nathan has covered this aspect in great detail, and also goes out of his way to explain the principles behind the vacuum forming process and how you can build your own vacuum former.

The two helmets are visualised differently and therefore use different electronics. The silver Thomas helmet with a scrolling marquee LED uses an Adafruit Pro Trinket, and is controlled from a smartphone over Bluetooth. The scrolling effect is created using four MAX7219 dot-matrix modules linked together. The helmet also has LEDs around the ears and are controlled by another Trinket. Similarly, the other gold Guy helmet uses several LEDs controlled via the Parallax Propeller microcontroller. The project is one of the more difficult ones to replicate, but three people have already done so on Instructables.

### RUN THE GAUNTLET

We aren't sure if Michael Darby is a fan of French electronic music, but by the time he came out of the theatre after watching *Infinity War* he knew he had to make the Infinity Gauntlet ([hsmag.cc/Xxqf0N](https://hsmag.cc/Xxqf0N)). Initially, Michael just wanted the gauntlet to be an aesthetic only project that you could wear on Halloween. But then he decided to use the glove to wreak Thanos-like havoc by eradicating half the artificial life he'd created in an earlier project ([hsmag.cc/KCgJnh](https://hsmag.cc/KCgJnh)).

He started with a pretty standard Thanos Gauntlet that's available in most costume shops, which he then activated using an Arduino Pro Mini, some different coloured LEDs, a couple of buttons, and a bunch of other electronics. The finished product lights up just like the real thing, and unfortunately for the artificial life forms in his other project, does pretty much the same damage.

Costume designer Sarah James has another idea of lighting up the night. The key ingredient in her Light Up Costume ([hsmag.cc/aazOsx](https://hsmag.cc/aazOsx)) is electroluminescent (EL) wire. She used green wires of different thickness; a 4 mm thick wire for the body and a 2.2 mm thick wire for the gloves. In her Instructables, she shares her

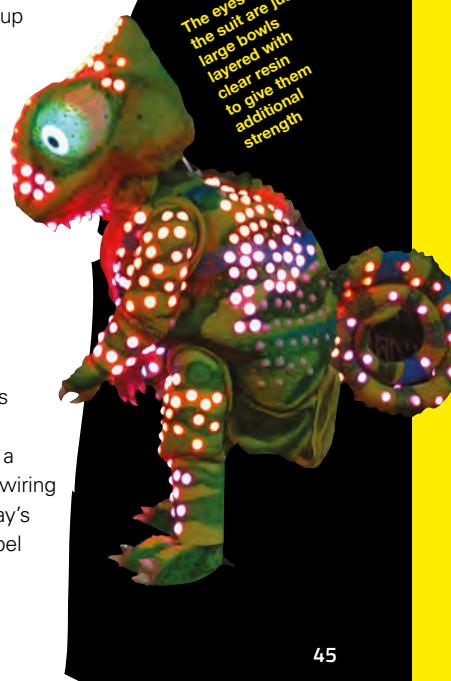
Michael didn't tuck the components inside the glove, and we feel this adds another dimension of wickedness to the gauntlet

## HE'S REPLICATED THE TWO HELMETS WORN BY DAFT PUNK

design process, which involved creating a mock-up with rope, before hand-sewing the EL wire.

If you want something more extravagant, here's an impressive colour-sensing chameleon costume ([hsmag.cc/PNGwHi](https://hsmag.cc/PNGwHi)). It was made with about 600 LEDs, and required over two dozen lithium-ion battery packs. The suit has a colour sensor in one of the hands. When it touches any coloured surface, an Arduino changes all the LEDs in the suit to match that colour.

The suit is fabricated with foam by Klay Jones who does this for a living (Partybot Designs). Besides detailed instructions, his Instructable has lots of useful information and advice for anyone interested in this fabricating foam. The project is a major undertaking, from constructing the suit to wiring and powering the electronics. But one look at Klay's videos of the finished product is enough to compel anyone to put in the effort. →



# AUTOMATIC MOVEMENTS

MAKE YOUR OUTFIT WIGGLE, POP, SCROLL, AND DANCE

**I**ights aren't the only thing you can use to dazzle a Halloween evening. There's nothing quite like an unexpected movement to make the hair stand on end. In that regard, Adafruit's aptly named HalloWing board is just the thing you need.

Mark Stevenson has used one to create an Alastor 'Mad-Eye' Moody costume for his son ([hsmag.cc/bgAaFm](http://hsmag.cc/bgAaFm)). A majority of his Instructable is about perfecting Alastor's get-up by creating the various elements of the costume, from the duster coat to the wooden staff. To create the mad eye, he's pointed to Adafruit's Bionic Eye tutorial ([hsmag.cc/PUCYUS](http://hsmag.cc/PUCYUS)) while detailing what he did differently.

Following the tutorial, he 3D-printed the eyeball, socket, and motor mount for the eye and went one step ahead and 3D-printed the goggles that he adapted from a model over at Thingiverse. He then sewed a black elastic to the side of the goggles and loosely sewed the wires along the inside of the strap to make sure the wires don't hang, but also don't restrict the elastic from stretching. He then focussed on making sure the contraption is comfortable enough for his son to wear all night.

If you're looking to add some thrills to your Halloween costume, Dano Wall, the Deputy Director of Manufacturing at Adafruit has just the thing. He's taken a standard foam reindeer mask and turned it spooky with animated eyes and a glowing nose ([hsmag.cc/oiYcMp](http://hsmag.cc/oiYcMp)). As you can probably

guess, he's used the HalloWing as well. Instead of one though, he's used a couple that were first synced together to mimic real eyes. Dano points to another Adafruit tutorial ([hsmag.cc/CDOZqe](http://hsmag.cc/CDOZqe)) that shows you how to wire up multiple boards. Just remember to string the boards together with a long enough length of wire to position them inside the eye sockets in your mask. He complements the eyes with an LED on the nose. Follow Dano's advice to make sure all the wires and electronics are secured properly to the mask, and also kept away from your face.

If you really want to turn heads, follow ace costume designer Andrew, who's an engineer by day and

Andrew first downloaded photos of the Ninja and then scaled them for his height using a formula that he's shared in his Instructables

cosplayer by night. His Raiden Cyborg Ninja costume ([hsmag.cc/uNeaHC](https://hsmag.cc/uNeaHC)) uses a motorised face shield that looks awesome. The major part of his Instructable focuses on fabricating and sculpting the costume from EVA foam using patterns crafted from cardstock. He created the face shield using two high-torque servos and a Pololu Maestro Servo controller. The microcontroller is programmed to move the two servos

## THE BOOTS HAVE INFRARED SENSORS THAT LIGHT THEM UP

simultaneously at the push of a button. He also created a sword and then wired some LEDs in the sheath diffused with a blue Mylar folder to get the right effect.

Andrew's also created a Stealth Iron Man costume that has a mechanised helmet ([hsmag.cc/dysXII](https://hsmag.cc/dysXII)). Again, the majority of his Instructable is about crafting the costume to perfection. Just like before, the helmet is powered by two servos glued to the back that are operated by the Pololu Maestro Servo controller. LEDs in the gloves, eyes, and chest piece complete the look.

The LEDs in the eyes are wired to a reed switch that switches them on and off, depending on whether the face mask is open or closed



Anatoli even has a custom suitcase to store his costume that's branded with a Predator icon and an Instructables logo

Jerome Kelty, a bicycle designer turned jeweller also loves creating elaborate costumes. One of his most extravagant ones was put together when his friend approached him to animate his fibreglass Iron Man MkIII costume ([hsmag.cc/stqMPa](https://hsmag.cc/stqMPa)). He decided to replicate as many functions of the suit as depicted in the movie.

Jerome's project uses an extremely large number of electronics. For instance, in addition to 20 servos, the suit uses four Arduino Pro Mini boards. There are two on the hands, one on the helmet, and the fourth on the boots. A couple of digital mini servos and XBee radios control the helmet. Two RFID tags in the left glove trigger the helmet, hip pods, and back flaps. Similarly, the RFID tags in the right glove trigger the forearm missile and shoulder rockets. The boots have infrared sensors that light them up and make mechanical movement sounds when the boot is lifted from the ground. Replicating the build is no easy task, but Jerome's Instructable has extremely detailed and illustrated instructions.

Just like Jerome, Anatoli Arkhipenko also loves to animate ready-made costumes. However, his choice is sure to freak out everyone at a Halloween party. That's because Anatoli's Predator costume doesn't only sound like one, but is complete with a head-tracking plasma gun and laser sights as well ([hsmag.cc/Zlvkre](https://hsmag.cc/Zlvkre)). Anatoli has combined elements of a ready-made costume with some 3D-printed components such as the faceplate and gauntlets. The motion, light, and sound effects are controlled by a Wii Nunchuk controller, while a Teensy 3.5 microcontroller is used for the head tracking. It took Anatoli over a year to complete the build working only on the weekends, but you can replicate his costume in a fraction of that time thanks to his easy-to-follow instructions. →

Jerome has created an animtronics forum for makers to help each other as they animate their builds



# AUDIO EFFECTS

**CRASH! BANG! WALLOP!**



The mask has a rotary selector switch that you can use to switch to another sound effect from the collection you've uploaded to the mask

# W

**hat's Halloween without sounds?** Scary or not, sounds can make inanimate objects come to life. If you can't afford to invest the time it takes to build Jerome's or Andrew's Iron Man Suit, how about just the gauntlet? Richard

Albritton's superhero glove ([hsmag.cc/HYkZde](https://hsmag.cc/HYkZde)) reacts to your movements and lights up and plays sounds accordingly.

The 3D-printed glove, which he's sanded and painted over to camouflage its origins, is powered by

an Adafruit Feather 32u4 Basic Proto microcontroller. It uses an accelerometer to trigger the sounds from the Audio FX Mini Sound Board and make the NeoPixel Ring and Jewel light up. Assembling the electronics took him about an hour and a half, and besides textual instructions, there's also a video of the process, which is easier to follow for the visually inclined. Richard's also shared a bunch of sounds for you to download into the Sound Board. Besides a slide switch to power-up the glove, there's also a button to disable the motion control and the associated sound effects to ensure your friendly wave to a friend doesn't 'blast' away others.



## WHAT'S HALLOWEEN WITHOUT MONSTERS?

The Audio FX Sound Board is a wonderful device to give a voice to your wearables. Prolific maker John Edgar Park has taken apart a talking Chewbacca Mask to increase its audio repertoire ([hsmag.cc/JSmgNe](https://hsmag.cc/JSmgNe)). The crux of his build involves replacing the mask's original sound-board that had limited audio capabilities with an Audio FX Sound Board that can playback your favourite sounds. He's kept the mask's existing switches and speakers, and has complemented them



with an amplifier, rechargeable batteries, and a power switch, in addition to the Audio FX Sound Board.

A major part of the build involves tearing down the original electronics from the mask. Needless to say, but the teardown instructions will only work with the kind of mask that John is using, however, you can follow them in principal to get an idea of what's involved. The second part of the build process is where you put together the new circuit, which involves soldering the amplifiers and other components to the Audio FX Sound Board, and is a lot more involved and time-consuming. Again, there's a video of the process that'll be easier to follow than the illustrated guide.

### TALKING HEADS

Once you're done vocalising Han Solo's friend, you can do so with yours as well. The intrepid Becky Stern uses the Audio FX Sound Board to build a talking dog collar ([hsmag.cc/gYoMwj](http://hsmag.cc/gYoMwj)). Her project uses the Bluetooth-enabled Bluefruit LE microcontroller to play sounds through a mono amp connected to a mini metal speaker embedded in her dog's collar. The build is inspired by the talking golden retriever named Dug from the movie *Up*, who vocalises his thoughts via a special collar.

Becky's replica uses some 3D-printed bits that also conceal the speakers, a couple of LEDs, alkaline batteries, and other electronics. Her video makes the process look easy and, once you've assembled all the components, all it really requires is a good amount of soldering longer wires. Once you've tested them, you can glue the bits to the circuit, and you're done. The



## WEARABLE CELL PHONE

Halloween isn't only about scaring people, is it? Over the years, Halloween has become an outlet for a maker's creativity and ingenuity. And maker extraordinaire Bryan Cera has the perfect wearable to impress everyone at the party: a fully functional mobile phone that you put on your hand like a glove ([hsmag.cc/oOoWZR](http://hsmag.cc/oOoWZR)). Bryan says his project is "the literalisation of Sherry Turkle's notion of technology as a 'phantom limb', in how we augment ourselves through an ambivalent reliance on it, as well as a celebration of the freedom we seek in our devices."

The first part of the build involves fabricating the custom glove. Instead of using fabric, Bryan decided to 3D-print one to give the project a more robotic look and feel. The next step involved ripping apart the Burg Watch Phone that has small tactile buttons that are well suited for the project. In addition to the buttons, he also reused the battery, microphone, and speakers from the phone. His Instructable has detailed instructions and videos of the different parts of the build process.

collar is controlled via the Bluefruit LE app that sends instructions to the collar over Bluetooth to trigger the sound samples stored on the Audio FX Sound Board and make your furry friend come to life.

Talking of furry creatures, what's Halloween without monsters? Adafruit's Creative Engineer, Phillip Burgess brings together two of his favourite things, electronics and Halloween, to bring a demon to life ([hsmag.cc/WSeiRW](http://hsmag.cc/WSeiRW)). The main aspect of the costume is the monster mask, which combines a couple of Phillip's earlier tutorials.

He starts by first using the Adafruit Wave Shield to create a voice changer ([hsmag.cc/BcigOE](http://hsmag.cc/BcigOE)). He mounts it inside a mask from a craft store that he had to modify significantly in order to wear it. Instead of following his lead, Phillip suggests you mount the voice changer inside a mask from a costume store that's actually designed to be worn.

The monster's animated eyes ([hsmag.cc/mcPeaC](http://hsmag.cc/mcPeaC)) are another significant aspect of the project. Instead of directly attaching the matrix backpacks to the mask, he mounts them on a flat acrylic plate that's attached to the mask via elastic bands. Make sure you place the LEDs slightly above your eyes to be able to see. The mask is camouflaged under a see-through sheer fabric, and Phillip also shares a couple of ideas to hide the speakers as well.



Becky advises prototyping the project on a solderless breadboard, which will be of great help, especially for inexperienced makers like the feature's author



# ULTIMATE ACCESSORIES

PROPS TO SPOOK,  
SCARE, AND DELIGHT

You can use the Eye of Newt as a creepy prop, or add it to your Halloween costume to raise the chills



**H**alloween decorations are an important aspect of the celebrations. With a little electronics though you can take their spookiness to unprecedented levels. Erin St. Blaine has taken a leaf out of Shakespeare's *Macbeth* to create a 'real' eye of newt ([hsmag.cc/0YalQG](http://hsmag.cc/0YalQG)).

The project runs off the Teensy microcontroller, which controls the OLED display breakout board. Erin grabbed the images of the nevnt eyeball from the web – she then photoshopped to her preferences. She's shared them, along with the code for the project and detailed assembly instructions. The project uses a photocell to make the pupils dilate in low light, which is a nice touch. Erin's also created a fabric case for the prop, which hides all the electronics and displays the eye through a cabochon. Before you set out, remember that the project might look easy but involves a lot of soldering.

Sometimes projects come about accidentally. It wasn't until after John Thurmond had bought an animatronic triceratops skull on impulse, did it occur to him that it wasn't particularly interesting. John, who is a founder member of the Guild of Makers, then did what any maker would do; he turned it into a talking skull that greets people ([hsmag.cc/OLGWBz](http://hsmag.cc/OLGWBz)).

John ripped open the skull and replaced most of its



If you lack the skills to build your own, Otto also sells disassembled ready-to-fly kits for about £1500

## FLYING WITCH

Not content with scaring only a room full of people, Otto Dieffenbach instead terrorises the whole neighbourhood with his remote-controlled flying witch ([hsmag.cc/eaUnWo](http://hsmag.cc/eaUnWo)).

The flying witch is mainly made of Depron foam pieces, along with some plywood. Otto has cleverly incorporated the essential elements of flight (lift and steering) in the design of the witch. For instance, the cape controls the lift, while the broom does the steering. The build uses an electric motor and a 10-inch propeller for propulsion. As you can imagine, the build is fairly complicated and involves a fair bit of woodworking, but it is well worth the effort. Make sure you follow the instructions to the letter, or else you'll end up throwing away its centre of gravity. Otto has created more than one flying witch, and besides detailed instructions, he has also shared the changes and improvements he's made to the build process.



Using an Echo Dot can be a security issue. Since we don't want random people to talk to Alexa, always remember to mute its microphone.

internals with an Amazon Echo, an Arduino, and a speaker. He also placed an Adafruit Motor Shield inside the skull and connected it to the skull motor. The contraption was programmed to turn the motor and move the jaws of the skull. The whole animation, audio, and jaw movement were triggered by the skull's original motion sensor.

### MAKE THEM JUMP

On similar lines, the Haunted Jack in the Box ([hsmag.cc/potCxt](http://hsmag.cc/potCxt)) also took shape as its maker went about building it. Sean Hodgins tells us that he knew what he wanted to build, but "just didn't know exactly how it was going to get there". Sean used his background in automotive technology and AI to put a Raspberry Pi to good use and automatically cranks the popular prop making it a lot scarier.

First, Sean 3D-printed a shell for the contraption. He then paired a Raspberry Pi Camera Module to a Raspberry Pi 3 that runs a face-detection script to fire up a servo motor that cranks the music box and fires up Jack. You can find the Python script, along with the complete schematics for the project, in Sean's Instructables page. He warns us that anyone looking to replicate his build must be prepared to deal with a lot of wires strewn about the toy.

From one Halloween staple to another. Jack-O-Lanterns are the most common props you'll find on Halloween, and while you can be creative with

# BODY BOARDS

If you're reading this magazine, chances are you've dabbled with microcontrollers to do interesting things. Wearable microcontrollers are specially designed to be concealed in clothing and can be sewn into costumes. Many of them are also washable, so you don't have to remove them before chucking the costume in the washing machine.

The LilyPad Arduino was the first board that featured sew-through contacts for stitching soft circuits. It's still popular and currently available in several versions and can be easily extended with sewable LEDs, sensors, and more. LilyPad inspired the Adafruit FLORA, which has a lot more sewable modules. If you don't require too many swappable modules, SquareWear might have everything you need for your wearable project.

Adafruit's Gemma is another popular microcontroller that is ideal for tiny wearables that don't require a lot of I/O. If you need more I/O without compromising on the size, there's the TinyLily Mini, which lacks some of the conveniences you get with other boards. There's also the Teensy, which isn't designed for wearable as such but is popular for them because of its minuscule size. Despite its size, this Arduino IDE-compatible microcontroller packs in quite a punch. Several projects in our feature use the Adafruit HalloWing, which was designed specifically for powering Halloween projects. It features a TFT display, light and motion sensors, a speaker driver, and more so that you can roll it into your Halloween costume or prop with ease.

them, making them throw flames will definitely turn heads. Markus Haack's fire-breathing Jack-O-Lantern ([hsmag.cc/YfSczk](http://hsmag.cc/YfSczk)) is IoT-enabled using MQTT.

Before you begin, please understand this build is extremely dangerous because it deals with fire. Please replicate this build after taking appropriate protections. We'd also like to repeat Markus's warning: make sure you place the pumpkin at a safe location to not harm any person or property. We'll add our own warning: compressed flammable gasses and candles are a hazardous combination. Take responsibility for your own safety and only work with them if you're confident of the safety of your approach.

The project basically involves triggering a can of WD-40 to spray its contents across a burning candle inside the pumpkin, which erupts outside giving the impression that the pumpkin is throwing flames. Needless to say, the first part of the project involves building the flamethrower. Markus has built one atop a wooden platform. The key element of the platform is a servo-controlled piece of wire over the nozzle of the WD-40 can. The servos, in turn, are triggered by PIR motion sensors that detect when someone is approaching the prop. The project is simple to wire and is run via the minuscule ESP8266 board. →

Markus uses the Homie project to add MQTT support to his build in order to make the pumpkin spit fire from a mobile phone



Sean encourages anyone who wants to replicate his build to come up with some different 'scary' sequences



# EMBEDDING EYES

**WINDOWS TO THE (ELECTRONIC) SOUL**

In the front cover of this month's HackSpace magazine, you'll see one of our favourite Halloween effects, **Uncanny Eyes**. These are eyes displayed on small screens that can be embedded in any projects and casually look around, blink, and generally freak people out.

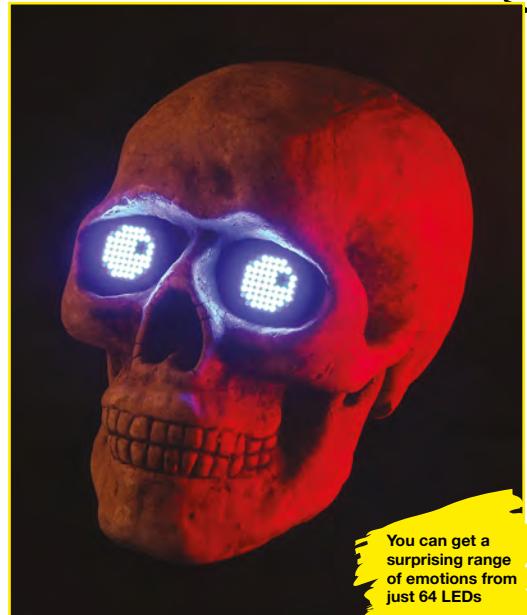
Although we've become accustomed to brilliantly realistic graphics in games, these graphics require huge amounts of processing power to create. As most people don't want a dedicated graphics card in their Halloween costume, we need a powerful microcontroller and some clever code. You just simply can't get this level of processing on an Arduino Uno or similar AVR board.

## WE NEED A POWERFUL MICROCONTROLLER AND... CODE

Powerful microcontrollers and coding tricks necessary to make these eyes work came together thanks to Phil Burgess (known to the Twitterverse as PaintYourDragon), Creative Engineer at Adafruit, and the Teensy 3.1/3.2 microcontrollers. You can still find the guide for doing this (from 2015) here: [hsmag.cc/HjGMUs](http://hsmag.cc/HjGMUs).

While this works, there are easier ways of achieving this effect these days. You can get the effect out the box and ready to include in your Halloween build using a HalloWing from \$34.95 (for a single eye) or MONSTER M4SK \$44.95 (for two eyes). The effect can be powered by Arm Cortex-M0+ cores, (running 128×128 pixel displays) or more powerful M4 cores (running 240×240 pixel displays). It's the quality of image that's different between the two display resolutions, not the physical size.

Of course, while microcontrollers might still need all their processing might to generate eyes like this,



more general-purpose computers find it easier. The Raspberry Pi has always had a powerful graphics processing unit (GPU), so even the first version – which had limited processing power – could achieve quite complex graphical processing. There is still quite a bit of CPU processing to be done, so if you want to achieve this look with a Raspberry Pi, we'd recommend the Raspberry Pi 2 or greater. There are instructions here: [hsmag.cc/boepBv](http://hsmag.cc/boepBv). At the time of writing, these don't work with Raspberry Pi 4, but we have a pre-release version that does work on the latest hardware (and eagle-eyed readers will spot it on this month's cover), and this should make it through to the website soon.

One advantage of the Raspberry Pi version is that, while it can use the TFT (or OLED) screens like the microcontroller versions, it can also use HDMI screens. This obviously includes monitors and TVs, but there's also a wide selection of smallish HDMI screens that are great for embedding. 3.5, 4, and

5-inch screens are widely available for quite reasonable prices, and these can be used if you want a larger eye in your build.

#### LOTS OF LEDs

Realistic eyes are a great effect on some projects, but they're not the only option. Smallish LED matrices, such as 8x8 grids, can produce eye effects that while maybe not realistic, have a cyberpunk charm that's right for more robotic projects.

Of course, you don't have to create your eyes using light; you can create a physical set of eyes and move them around. Careful use of ping-pong balls and servos can work, but the devil, as always, is in the details, and mounting everything can be a bit tricky. Fortunately, if you've got access to a 3D printer, Thingiverse user [jim2386](#) has done the hard work for you, and you can print out your very own set of animatronic eyes based on his designs at [hsmag.cc/AGuRIL](http://hsmag.cc/AGuRIL).



That's the eyes sorted,  
now you just need to  
decorate the rest of  
your pumpkin



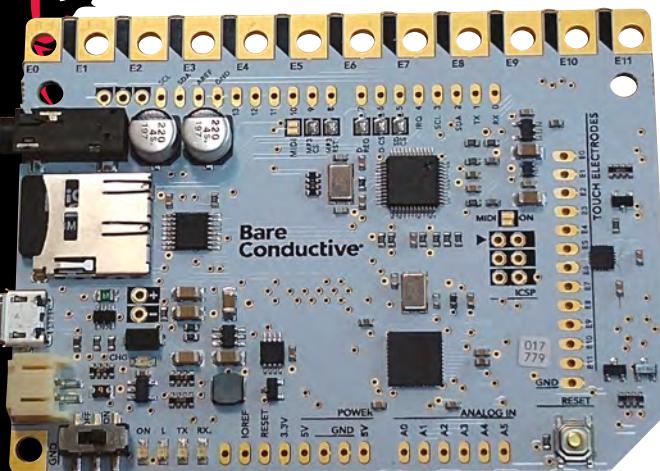
# SOUND EFFECTS

ADD SPOOKY, SCARY, AND CREEPY SOUNDS TO YOUR PROPS

**I**here are countless ways of adding sound effects to the project, but one of our favourites is using a **Bare Conductive Touch Board**. This is a great choice for two reasons. Firstly, it's incredibly easy, and secondly, sounds can be triggered with touch input, which makes it perfect for reactive sounds. This board is designed to work with conductive ink – and it does this really well – but it works equally well with anything conductive. The pads are large enough for crocodile clips.

Your board should come pre-programmed with the Touch MP3 sketch, but if it doesn't (or you've

Touch the pad and the sound is played – audio effects the easy way

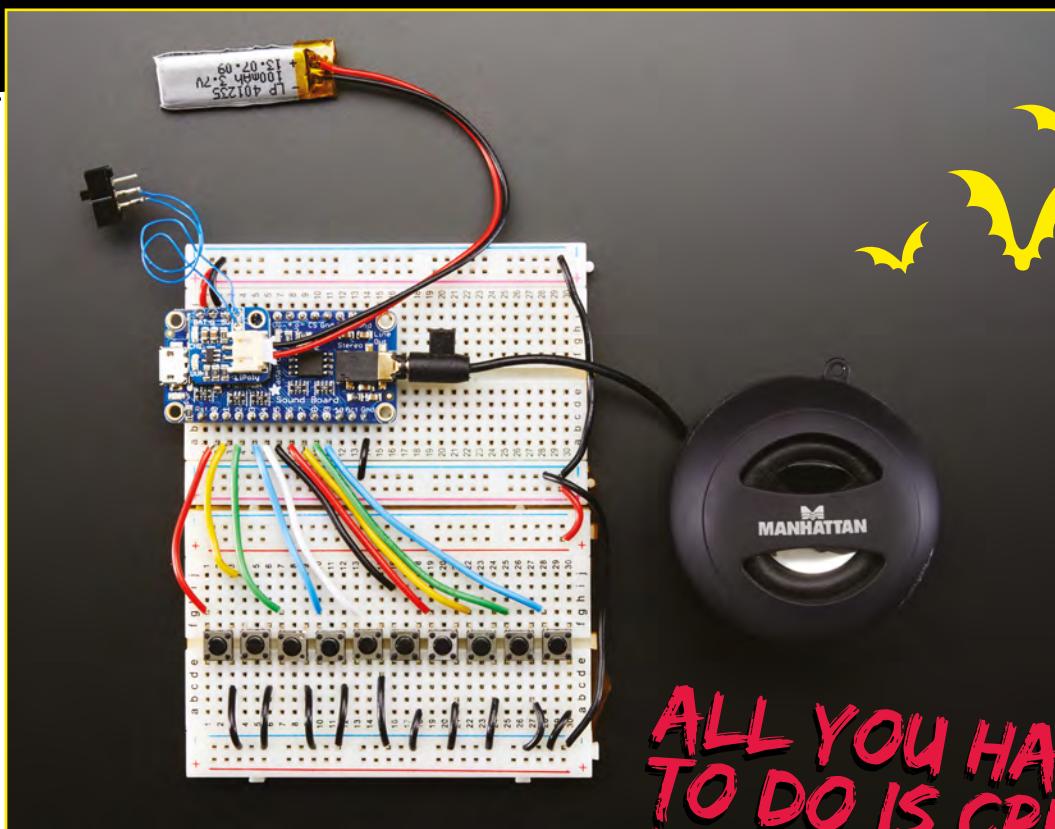


previously pre-programmed it to do something different) you can download the code from [hsmag.cc/UICkAv](http://hsmag.cc/UICkAv). You can use this in the Arduino IDE (see here for how to set it up for the Touch Board: [hsmag.cc/bGGeqn](http://hsmag.cc/bGGeqn)).

Once the code is uploaded, you're ready to go. There are three touch-sensitive pads down one side of the board labelled E0 to E11. Each of these triggers an MP3 file on the microSD card to be played. E0 will trigger file **TRACK000.mp3**, E1 will trigger **TRACK001.mp3**, etc. All you have to do is create MP3 files with the sounds you want and load them onto the microSD card.

The Touch Board includes almost everything you need for your sound effects, but it doesn't have a speaker. Any speaker that plugs into the headphone port should work. You can get small portable speakers that are great for little sounds, or you can hook it up to a large hi-fi or PA system if you really want to shock people.

The other thing you'll need is a way of triggering the sounds. The pins are touch-sensitive, and this transmits along wires, so if you attach a crocodile clip to the pin, touching the other end of the wire will trigger the sound. If you attach the other end of that crocodile clip to something conductive, then touching that will trigger the touch sensor. There's no easy way to calculate exactly how long this chain of conductive things can be before it stops being touch-sensitive, but the shorter the better. You might need to turn the board off and on again after attaching wires to recalibrate the touch sensors.



The Adafruit Audio FX Sound Boards are easy to use, but require buttons to trigger sounds rather than touch events

## ALL YOU HAVE TO DO IS CREATE MP3 FILES

You can make the final prop for the user to touch out of anything conductive. Large-headed pins or nails are a good option, as is tinfoil (provided it doesn't need to be too rugged). Conductive thread or cloth can make fabric touch-sensitive (but make sure it's not against your skin as it'll always be triggered). There are a few varieties of conductive paint and ink that also work. This is where you need to get creative for the maximum effect.

### ALTERNATIVE APPROACHES

The Bare Conductive Touch Board is a fantastic tool for adding sound to projects, but at £53, it's not the cheapest option. Adafruit has a few options under the name Audio FX, and SparkFun has a similar range under the name Soundie. They have different options in terms of amount of memory, method for connecting to speakers, and types of audio supported, so take a look through and see which one fits your needs from the diminutive Audio FX Mini Sound Board – WAV/OGG Trigger – 2MB Flash for \$19.95 that will need an additional amplifier to connect to speakers. Other options include audio-out jacks, the ability to play MP3 files, and more memory. All of these, however, play audio on a trigger voltage and are not touch-sensitive. This means that you'll need buttons to trigger the sound. Whether or not this will work

for you depends on how you want your effects to work.

You don't have to use hardware designed for sound effects. Many general-purpose microcontrollers include the hardware necessary for playing sounds, including the Circuit Playground Express (CPX). There are details of how to do this here: [hsmag.cc/ZkJzbC](http://hsmag.cc/ZkJzbC). There's a small built-in speaker on the CPX, but you can also attach an external speaker with a bit of wiring.

At the very cheapest end of the spectrum, you can get electronics designed for toys and greetings cards from a direct-from-China website. For around £1 you can usually find a module that plays predefined sounds (set at the factory), whereas for around £2 to £3 you can usually get something that will play a pre-recorded sound (recorded via a microphone connected to the board). The sound quality is usually terrible (as you've probably heard on cheap toys and greetings cards), and they can be a bit tricky to get started, but if you're on a tight budget – especially if you plan on making several objects with integrated sound effects – they can be worth investigating. □

# 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](http://hsmag.cc/subscribe)

# SUBSCRIBE TODAY

**Subscribe for 12 months**

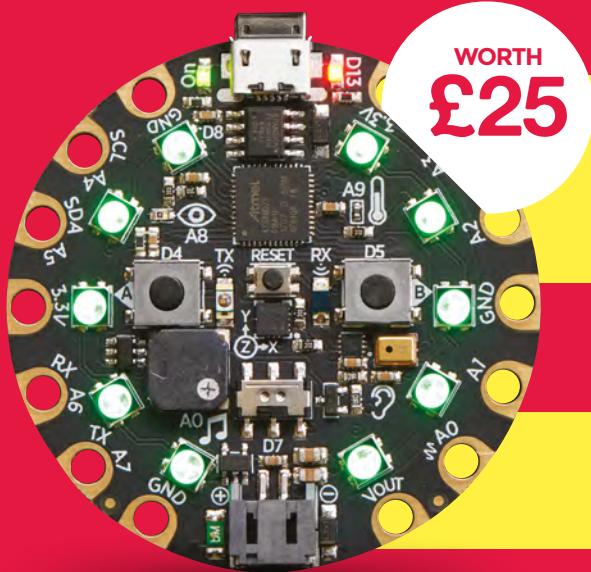
**£55** (UK)    **£90** (USA)

**£80** (EU)    **£95** (Rest of World)

Free Raspberry Pi 3A+ with 12-month upfront subscription only  
(no Raspberry Pi 3A+ with rolling monthly subscription)

**Rolling monthly subscription**

- ▶ **Low monthly cost** (from £5)
- ▶ **Cancel at any time**
- ▶ **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](http://hsmag.cc/subscribe)



# How I Made RACKETY RACCOON SYNTHESIZER

Combine the Mozzi library for Arduino and a dead animal to conjure magic

By Emily Velasco

**t's not every day I get to build a synthesizer that doubles as a cabinet of curiosities for a raccoon skull, but then, it's not every day I get gifted a raccoon carcass.**

The roots for this project, which I'm calling the Racketey Raccoon Synthesizer, stretch back to last August when an ex of mine showed up at my front door with a gleam in his eye and a beaming smile on his face. He meant to get us back together, and he thought his ticket back in was the enormous dead raccoon he had brought me.



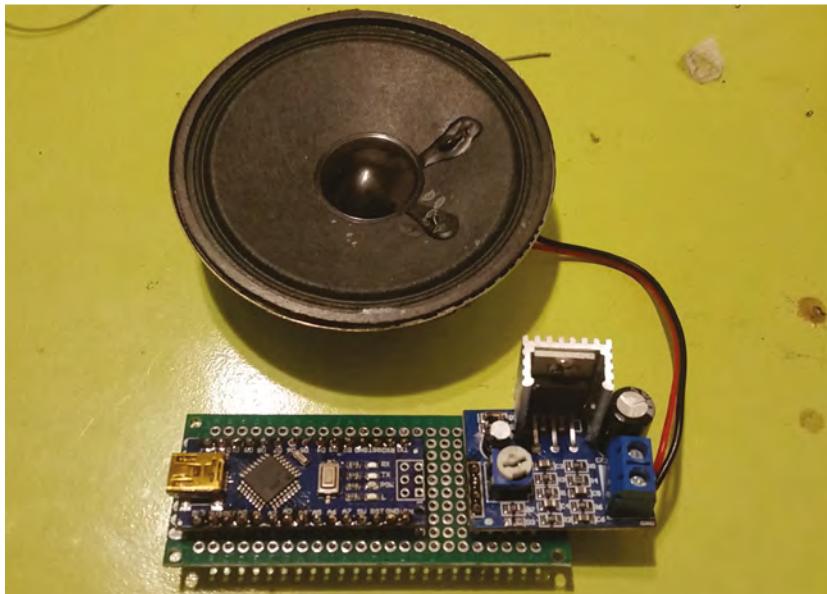
Above ◀  
Why make a box  
when you can  
re-use one?



Ultimately, things didn't work out between him and me, but the raccoon corpse stayed and over the course of nearly a year, I turned it from a slightly car-crunched dead animal into a set of clean, white bones I could use in my projects.

## THE SYNTHESIZER

Independently of my bone-related ventures, I've recently begun exploring Arduino-based synthesizers. Usually, I prefer working with analogue circuitry, but a few months back, I had a synthesizer project in mind that was better suited to a digital solution. After a bit of research, I stumbled on the Mozzi audio synthesis library for Arduino, and I was impressed. I've seen plenty of Arduinos synthesize sounds before, but they were always simple, square-wave tones – the kind you hear in 8-bit video games. By comparison, the sounds coming out of an Arduino running Mozzi were positively lush. There were the expected bleeps and bloops, but also warbling, vibrato, echoes, and tones that softly faded in and out. So I downloaded Mozzi and got to tinkering.



**Left ◇**  
A bit of protoboard joins  
the Arduino and amplifier



**Above ◇**  
The cigar box brings character as  
well as providing a secure enclosure

## DOWN THE RABBIT HOLE

Deciding where to start with Mozzi was my first challenge because it can do so many things, and the sketches that come preloaded with it sound amazing right out of the box. There's an eerie theremin sketch, a sketch that varies tones based on feedback from temperature and light sensors, and even a sketch that will play back short samples of recorded audio.

It was all very exciting to play with, but my enthusiasm for writing my own Mozzi sketch soon found itself tempered by two

Then I started to get myself in over my head when I decided it might be fun to build a synthesizer that played back samples of raccoon sounds to go with the skull I harvested from the raccoon. In theory, this should be easy with Mozzi because it comes with a pair of Python scripts that are supposed to do most of the heavy lifting for the user. One script compresses audio samples so they're small enough to fit into an Arduino's meagre flash memory. The other takes the samples and converts them into a format that can be played back by Mozzi.

I spent many hours sitting cross-legged on the shag rug in my den, tweaking existing sketches to see what results I could get

things: 1) I'm a terrible coder, and 2) The documentation for Mozzi leaves a lot to be desired.

Nonetheless, I spent many hours sitting cross-legged on the shag rug in my den, tweaking existing sketches to see what results I could get. At one point, I had a sketch that varied the playback speed of a sample with the turn of a potentiometer. At another point, I had a sketch that modulated a chorus of heavenly tones based on the position of a joystick.

In practice, neither was easy to use. The file conversion script kept spitting out an error I didn't understand, and it took me several days and some insights from a friend to figure out that "too many initializers" means "your audio file is too big." The compression script took me on an even more meandering path that first led me to try installing a series of Python libraries that the script needed. When those attempts were unsuccessful, as they required a version of Python that was incompatible with my →



**Above ◇**  
Pink LEDs, because why not?

# How I Made: Racket Raccoon Synthesizer

## FEATURE

**Right** ♦  
Potentiometers  
in position  
ready to make  
some noise



### Mozzi

The Mozzi library is designed to let you create sound on Arduino boards that don't have digital-to-analogue converters (DACs). These can't play true audio, but can approximate it using clever trickery and a digital output pin. It requires minimal circuitry, and you can get up and running quickly with just a board, a pair of headphones, and a way of connecting the two. Find out more at [sensorium.github.io/Mozzi](https://sensorium.github.io/Mozzi)

MacBook, I even went as far as reviving an old laptop as an Ubuntu machine so I would have free rein over what I did with Python.

After a couple weeks of being lost in my own personal Python hell, I shoved everything aside and started over again. My original goal was to build a synthesizer, not a box of raccoon sounds, and I had gotten very off-track. I started looking around online for someone who had already built a synthesizer using Mozzi, and I found a nice Instructable from user Dorahan, who built an Arduino synthesizer with five potentiometers for controlling pitch, vibrato intensity, vibrato rate, and the range of

adjustment for vibrato intensity and rate. I liked what I heard from the short video he included in his Instructable, so I downloaded the sketch and started gathering parts.

### THE AESTHETIC

Since I knew for sure that I wanted the synth to feature a skull, I let that drive some of my decisions about the design. Drawing from the cabinet of curiosities aesthetic, I went in search of a wooden enclosure. A really excellent source of small, wooden project boxes is cigar shops. Despite our living in a modern era where nearly everything comes encased in plastic blister packs, cigars still often come in wooden boxes of surprisingly high quality. It used to be that you could get these boxes from a cigar shop for free, but now the shops usually charge a few bucks apiece. Still, even at \$5 a box, they're a lot cheaper than metal project enclosures, and they add some old-school DIY electronics classiness (think Apple I prototypes).



**Left** ♦  
The speaker mount



**Left** Clamping and gluing the supports for the acrylic

For this project, I picked up a few cigar boxes at a local tobacco shop, but then I stumbled across a stunner of a cigar box at one of my favourite thrift shops. It was clearly a few decades old, worn enough to have a lot of character, and it had a hinged lid, which is rare in wooden cigar boxes. I paid \$2.50 for it.

The other aesthetic I was dead-set on with this project was lighting it with pink LEDs. Several years ago, I saw a video of a drone synthesizer somewhere online. Its case was made of acrylic, and it was lit internally by pink LEDs. I had never seen pink LEDs before, and I marked them down in my brain as something to use in a project someday. I found a hundred of them for \$6 on Amazon and placed an order.

I rounded out my supplies for the project with a sheet of acrylic, some potentiometers and knobs, a tiny mono amplifier that runs on 6–12 volts, and assorted bits from my collection of junk, including a small speaker and a limit switch.

### THE GUTS

The first thing I needed to do was assemble the guts of the synth. I didn't want to sacrifice an entire Arduino Nano to the project, so I opted to use a knock-off Nano, which resulted in some complications down the line. One of the things that drive the price of an Arduino is the use of an FTDI chip for

Unfortunately, I installed a copy of the buggy driver on my computer, and I lost a day of progress while fixing the problems it caused.

After getting the driver issues sorted out, I slightly modified the Mozzi sketch I downloaded from Instructables and loaded it onto the knock-off Nano. Then I soldered the Nano to a small piece of perfboard to

give myself more room to wire things up. I selected five of the analogue input pins (A0–A4) and wired each to the wiper leg of a potentiometer. Each potentiometer was also connected to ground and

the 5V pin of the Nano.

I soldered the small amplifier to the perfboard as well, and then wired its input pin to digital pin 9 of the Nano, which is where Mozzi outputs audio. To power the amplifier, I connected its supply voltage pin to Vin on the Nano, and then tied their grounds together. Hooking up a speaker was as simple as sticking its leads into the screw terminals on the amplifier and tightening them down. I soldered a pair of the pink LEDs to current-limiting resistors →

I didn't want to sacrifice an entire Arduino Nano to the project, so I opted to use a knock-off Nano

USB-to-serial communications. The knock-off Nanos cost less, in part because they use a cheaper alternative, the CH340 chip. The CH340 chip behaves in basically the same way as an FTDI chip, but with one crucial disadvantage for me – Mac computers don't come with a CH340 driver. This should be a simple fix: just download and install the appropriate driver, but one must be careful: there are versions of the CH340 driver floating around the internet that have a bug that causes a kernel panic in Macs.

# How I Made: Racketey Raccoon Synthesizer

## FEATURE

**Right** ♦  
The electronics fitted snugly  
in the side of the box

**Below** ♦  
The finished instrument



and long leads that connected to the positive and ground rails of the combined Nano-amplifier board.

All that was left was to supply the combined parts with 9 volts to see if it all worked. I soldered a 9-volt battery connector

fit inside the cigar box. This was one of the simpler parts of the build, just requiring me to design the panel with CAD software and then cut it on a laser cutter. I gave it mounting holes for two of the potentiometers, and some cut-outs to allow

With the most tedious part – the electronics – out of the way, it was time to really have some fun

and a limit switch to the board, and plugged a battery in.

Now, I'm not the most skilled person when it comes to soldering, but in this case, I did everything right, and I had functioning synthesizer guts sitting on my workbench.

### FABRICATION

With the most tedious part – the electronics – out of the way, it was time to really have some fun.

Since I wanted the raccoon skull to be visible to the user, but protected from damage, I fabricated an acrylic panel to

sound from the synthesizer to project out at the user.

The panel was mounted inside the box by way of stand-offs I fabricated from a piece of scrap cherry I had in my workshop. After cutting/sanding the stand-offs, I gave them a coating of tung oil to condition the wood, and then rubbed them with mahogany gel stain to give them a similar colour as the mahogany box. The stand-offs were daubed with wood glue, and clamped into the corners of the box until the glue had set.

As much as I hated to modify the cigar box because of its advanced age, I wasn't going



to be able to get away with leaving it in original condition. So, I compromised and allowed myself to modify it, while trying to keep such modifications to a minimum. That meant drilling holes for mounting potentiometers and other hardware was OK, but cutting a big opening for the speaker was not. Instead, I chose to mount the speaker face-down to the bottom inside of the box. This reduced the volume of the output, but it also used the wood of the box as a resonator, softening up some of the harsher sounds from the synth. Again, this required some CAD work. I whipped up a design for a speaker-mounting flange and 3D-printed it.

#### OUR LOYAL FRIEND, HOT GLUE

Now I had all the parts needed for final assembly, which made liberal use of one of my favourite tools, the hot glue gun. The speaker and its flange were glued to the bottom of the box, and the Nano-amplifier board got glued to one of the inside walls.

I slipped three of the potentiometers through the holes I drilled in the walls of the box and secured them with their nuts.

The pink LEDs were glued to the inside of the skull, so they would light it from within. By the way, bones make gorgeous diffusers for LEDs. The skull got glued down on top of the speaker, mostly hiding it.

To complete the synthesizer, I attached a holder for a 9-volt battery to the inside of the box and then attached the limit switch so that when the lid of the box is shut, power to the synthesizer would be turned off.

#### SO, UM, WHY?

When I show this synthesizer off to someone new, the first question out of their mouth is usually, "Is that a real skull inside?" When I affirm that it is indeed a real skull, their next question is often, "Why?"

I can answer a lot of questions about my projects, but explaining why I made a synthesizer with a roadkill raccoon skull inside is not easily explained. The honest answer is I don't really know. It just seemed like a cool thing to do, and now I'm probably the only person on the planet with a pink-lighted, electronically screaming raccoon skull in a box. □

**Above** ♦  
Candles are a  
recommended extra

#### Animal Bones

Some of the very earliest tools were made out of animal bones, and they've been a staple material for makers for millennia. It's only in the past hundred years or so that they've started to be seen as an unusual material.

However, making anything with biological materials carries hygiene risks that you need to understand and mitigate. Working with animal parts isn't necessarily dangerous, but can be. Make sure you know what to do before someone turns up on your doorstep with a dead raccoon.

HackSpace magazine meets...

# The Volcanologists

Raspberry Pis go in where others fear to tread

**S**cience is, sad to say, expensive. You need a ton of gear, a lab, a travel budget... and if the powers that be decide you can't have a grant, you get no funding and the research just doesn't happen. That is, unless you're a maker – then you can just build your own instruments.

We travelled to the University of Sheffield's Geography department to find out how one small team of volcanologists are using the tools you have in your local makerspace to deepen our understanding of planet Earth. With 3D-printed parts, self-taught Python skills, and cheap Raspberry Pi kit, they're putting knowledge in the hands of local communities and massively increasing our understanding of what's going on under the Earth's crust. Strap in for some science... →

**Right ↗**  
The Sheffield  
volcanologists (from  
left): Tom Pering,  
Andrew McGonigle  
Tom Wilkes,  
Tehnuka Ilanko





**HackSpace** You're monitoring volcanic activity, with a Raspberry Pi Camera Module. How on earth does that work? We would have thought you'd be taking samples of gases, or using a seismograph to measure vibrations under the earth. How does taking photographs help you discover what's going on in the centre of a volcano?

**Andrew McGonigle** One of the main components of volcanic gas release is sulphur dioxide [SO<sub>2</sub>], which absorbs really strongly in the UV range at a wavelength of about 300 nanometres. The gases come out of the top, so if you can develop a UV imaging camera and point it at the volcano, some pixels are going to be darker, because of that absorption of the UV light. You can see that absorption, and you can process that to work out the amount of gas coming out per second.

What you end up with is something that looks like the output of a thermal camera, but showing gases escaping, rather than heat. Scientific-grade UV cameras are really expensive. It was almost an accident that we started working with Raspberry Pi technology, but a good accident, in terms of costs.

### HS How expensive is 'really expensive'?

**Tom Pering** The UV cameras traditionally used in volcanology are meant for things like astronomy, and they're adapted for our purposes: that's why they're so expensive. The commercial systems that we used to use, and that our colleagues in Italy use, cost somewhere in the region of £10K to £14K, that level. Using Raspberry Pi, we can make one for a couple of hundred quid, which is absolutely amazing. Even the design and development stage, we probably only spent four or five thousand pounds. We undercut the commercial system immediately.

Part of the reason we started looking into this is that we couldn't buy the commercial system: we couldn't afford

it. We couldn't get the grant that we asked for, so we decided that we'd make one ourselves.

We started looking at hacking the Raspberry Pi Camera Module, and it's surprisingly sensitive to UV, once you take the Bayer layer off. There are certain ways that you do that to make it as good as possible, and it's actually better than some systems you can buy for £10K.

**Tom Wilkes** I guess the idea came from astronomy, but it's actually just a smartphone sensor so it's modified gear that's actually just used for basically anything you could think of. People use it for surveillance, wildlife studies, so it's just your bog-standard camera sensor that you'd find in a smartphone. What's great about it is how cheap it is; that's

Using the  
**Raspberry Pi**  
**Camera Module** is  
actually better than  
some the systems  
you can buy for £10K

the main thing. Because so many people use these cameras, they have such wide applications, they can be made for next to nothing, and so that's the advantage of this.

You can use either type of Raspberry Pi Camera Module; once we remove the lens system, they're exactly the same sensor (the NoIR just has the near-infrared filter removed from the lens system).

**AMcG** The key issue is that we strip the sensor down so that it can see ultraviolet light, and then we rebuild the system with new lenses that are UV transmissive. There's a thing called the Bayer filter on the sensor that makes it see red, green, and blue pixels to get colour images, but for what we do, we need to see ultraviolet light, and that

filter absorbs all the UV. What we did was we worked out – after a lot of trial and error – a way of getting that filter off without killing the sensor.

It just so happened that the Raspberry Pi version 1 camera had a sensor where we could, relatively easily, get it off. The version 2 cameras, we think we've done it with hydrofluoric acid, but that's not really ideal. For which reason, when the version 1 cameras were discontinued, we bought in 400 of them, which are sitting in my office, so we're future-proofed.

### HS And what version of the Raspberry Pi are you using at the moment?

**AMcG** We're using Raspberry Pi 3 at the moment, because we need the built-in wireless capabilities. We connect from a laptop to the camera using the wireless that's built-in. We don't need to move to Raspberry Pi 4; I don't think this application needs [the extra power].

### HS What was the eureka moment that led you to build a camera?

**TP** It was just a random idea. We needed the cameras, but we couldn't afford them. I'd done some work previously on low-cost measurements; Tom came along and gave it a go, and it turned out to work really well. We tested it at Drax power station [near Selby] to begin with, then went to Etna to test it, alongside the commercial system.

**TW** We've got two versions: one in the bigger box that uses a USB power pack to power both Raspberry Pis, and there's this one that runs off mobile phone batteries. I had this running for six/seven hours off 3.7 V batteries. That's another advantage that we don't think about so much: the fact that these things use a lot less power than expensive UV cameras. That's really important when you're going somewhere as remote as Papua New Guinea, where power's an issue. A 20 amp power bank will last all day. →



**Right** ▶

This July, one of the cameras was lost during an explosion on Stromboli, in Italy

**Right** ▶

The cameras create a wireless hotspot that the accompanying laptop connects to – the team are working on updating the software so that the cameras don't need a laptop in order to operate





**AMcG** It was an educated guess as well. The architecture of the sensors has changed a lot in smartphones over the last couple of years.

On a previous design, the front-illuminated CMOS, you'd have the active area, which detects incoming photons, then you'd have all the wiring at the top. So the light gets blocked by all the wiring. But the back-illuminated CMOS, which was developed by NASA, all the wiring's on the back. That technology has only been applied at the smartphone camera level relatively recently.

We knew that this would provide far greater possibilities in terms of getting UV sensitivity. We knew that people had done this sort of thing previously, but with SLR cameras, removing the Bayer filter and getting UV sensitivity. To our knowledge, no-one had tried it with smartphone cameras. But, based on the physics and the architecture of the sensor, we had an idea that this would be a sensible thing to try. And if it didn't work, it didn't matter; we'd just spend a few hundred pounds destroying sensors.

Our end-user market is volcanologists who live in parts of the world where there isn't a whole lot of money, so part of the motivation is to develop low-cost gear. It wasn't just a case of our wanting this for our research purposes; it was also a case of making things accessible for others. We're now involved in a project where we're starting to get the bits of kit out into the field to end-users. Tom's just been in northern Chile with guys who are using some of our gear. We're hoping to set something up in Peru.

**HS** Can you give the equipment to people in the field and have them start using it straight away, or does it take a bit of training?

**TW** The more detail you go into with the cameras and the physics, the more complicated retrievals of SO<sub>2</sub> you can get, but a lot of volcanologists know a lot about rocks, but don't necessarily know a lot about gas measurements, so they want to be able to use this technique without needing to know all this detailed theory. So the idea is that this can be relatively simple.

A lot of the commercial systems don't come with code anyway, so people have got to write it. There are so many volcanologists groups writing their own code to do the same things. Recently a paper released some Python code that's open source that can do all of this. My thought is that it's a great piece of code, but the problem is it's still not very useful

because it still doesn't have a GUI attached.

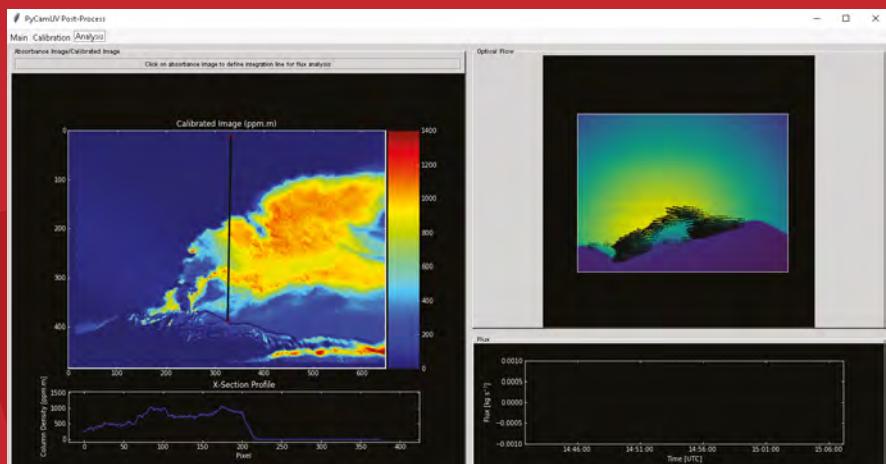
Maybe more commercial systems assume too much knowledge from other volcanologists, either in computer

There are so many volcanologists groups writing their own code to do the same things

programming or in gas monitoring. We try to simplify everything so the end-user can point and click. SO<sub>2</sub> absorbs UV at a certain wavelength. So we take two images: one wavelength where the SO<sub>2</sub> absorbs, then we do another wavelength where it doesn't, and we basically contrast them. We essentially divide one image by the other.

**HS** Apart from Raspberry Pi hardware, how else are you keeping the cost down?

**TW** The lens caps and mountings are 3D-printed, it's relatively simple, and there's Blu Tack used at the moment to help things fit together. This is graphite-reinforced nylon, printed using selective laser sintering. When I first tried, I did try with typical fused filament, but found that, because of the thread on it, I couldn't print with the detail we needed. There's →



**Above ♦**  
Sulphur dioxide concentrations show up in real time on the captured images

**Below ♦**  
The kit comprises two Raspberry Pis, two camera modules, 3D printed parts and a UV-transmissive lens



a bed of powder and a laser, and where the laser hits, the powder solidifies. It's to an accuracy of something like 0.1 of a millimetre.

The great thing about having a 3D-printed part is that we can make this bespoke. We can change the filter as and when we need to look for other gases.

And it makes good sense. We can actually build scientific-grade instruments using stuff that high-school students know how to do. How cool is that?

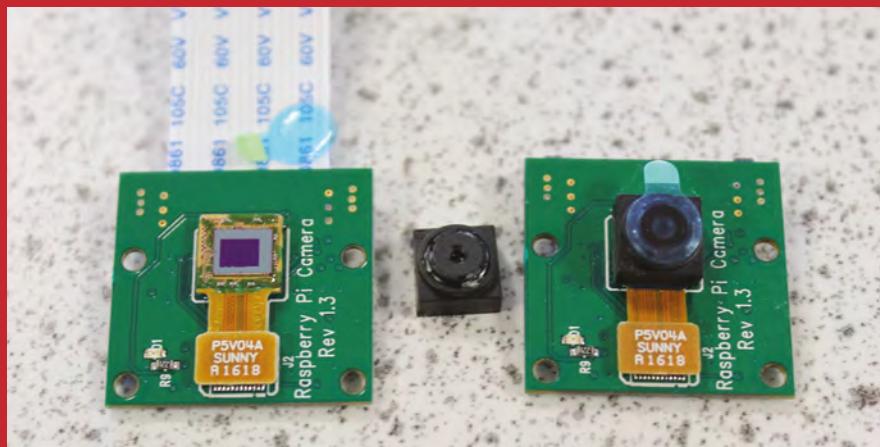
**AMcG** The reason for using that approach is that you can measure the gas release very quickly, at the frame rate of the camera, once a second or so. And that enables you to look at really quick processes going on in the volcano. So the time resolution is really important – you're basically taking the volcano's pulse. Previous techniques could only measure a few times a day, or a few times an hour, so now we're getting a much better handle on what the volcano's doing. The reason for that is that volcanic activities are driven by gas; the gas is pressurised, and that causes explosions. Often those gas release patterns change very quickly depending on bubble bursts. That's why people got into cameras. Previously there were spectroscopic techniques whereby, in 10 or 15 minutes, you could work out how much gas was coming out. That's a useful workhorse in measuring gas release in volcanoes across the world.

But if you want to understand something like 'how does an explosion work', an explosion on Stromboli would last ten seconds. Taking a measurement every ten minutes would be useless.

But the two techniques are quite complementary. If you use spectroscopy along with UV monitoring, you can get a better calibration for your remote sensing.

**HS** Haven't you also had interest in this kit from NASA?

**AMcG** The whole business model of NASA has changed. Rather than landing single objects that cost a billion dollars, like an Apollo lander, they're now more interested

**Left ◆**

On the right, the Raspberry Pi Camera module in its natural state; on the left, the module with its lens and Bayer filter removed to expose the sensor

in lower-cost units, but maybe sending multiple rover units to the moon. They reached out to us to say that they had an application that requires a very low weight UV sensitive detector; would you like to partner with us?

We developed some sensors for them. With Raspberry Pi sensors, they've developed prototypes for spectrometers to go to the moon.

There's a limit to what I can reveal, but the whole unit that they've built is about  $150 \times 100 \times 100$  mm in size, and it's got to come in under 60 grams, so it needs to be really lightweight. I suspect it isn't 3D-printed, like ours. But our contribution has been to deliver

the sensor module. With the project, they've got to technology readiness level 4, so it's already reasonably mature in terms of where it's got, and it's passed a whole series of tests. But the amazing

The quantum efficiency of the Raspberry Pi camera version 1 is **absolutely unbelievable – it's comparable or better than some of the commercial systems**

thing about it is that NASA asked us to look at the quantum efficiency of the sensor. If you get 100 photons falling on the detector, how many electrons are generated? With these kinds of

**Left ◆**

Tom's written a GUI for the software, so you don't need to be an expert in Python programming on Linux filesystems to study volcanoes

sensors, in the visible range they're really sensitive, the quantum efficiency could be 70–80%, which is incredible, but you'd expect that efficiency to drop in the UV, mostly because of all these filters, but the quantum efficiency of the Raspberry Pi version 1 camera at 310 nanometres is 40%, so it's absolutely unbelievable.

It's comparable or better than some of the commercial systems. I've seen one quoted at 8% at 310 nm. Which is why this has worked so well. There are other issues to do with noise as well. That is just absolutely amazing. It's why it's worked. It shouldn't have worked, but it did.

**HS | What's next?**

**TP** We've bought in some stuff from Pimoroni. The Enviro+ has all sorts of interesting applications. There's a mini gas sensor on it, which I'm not sure will be useful for us, but the thing that will be useful is that particulate matter sensor that you can add on to it. That's looking at the size of particles in air pollution, but I don't see any reason why we can't use it to detect ash particles on a drone flying through a plume.

A lot of work I've been doing recently has been on using mini cameras, flying through gas plumes at the same time with mini gas samplers.

The other massive advantage of all this, from a volcano perspective, is that things get blown up. Just this summer there was an explosion and loads of kit got wiped out. If that kit costs thousands of dollars less to replace, it just makes sense. □



# SOCKS



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.

Don't work your socks off. Instead, take them off and put them to creative uses

**T**hey might seem like a modern-day invention, but socks are one of the oldest types of clothing that are still in vogue. In fact, you can use them as a barometer to track the changing fashion and taste

**through history.** Like many man-made marvels that have survived through the ages, the secret behind the timeless existence of one of the simplest pieces of clothing lies in how it serves our need for both fashion and function. Many historians believe that even the early cavemen covered their feet in the same fashion as we do today, albeit with a rudimentary version of the modern-day pair of socks. For a long time and across civilisations, the socks were some sort of leather fashioned from an animal's skin, tied around the ankles to hold them in place. The earliest instance of a knitted pair was discovered in a tomb in Egypt dating back to 1000 CE. In a major fashion faux pas, historians say that it was very likely these knits were paired with the popular footwear of the era: sandals.

For a long time, stockings were a privilege of the rich and the affluent. Knitted and woven socks became a status symbol of the nobility throughout much of Europe, and their manufacturing process was a closely guarded secret. That was until William Lee, an English reverend

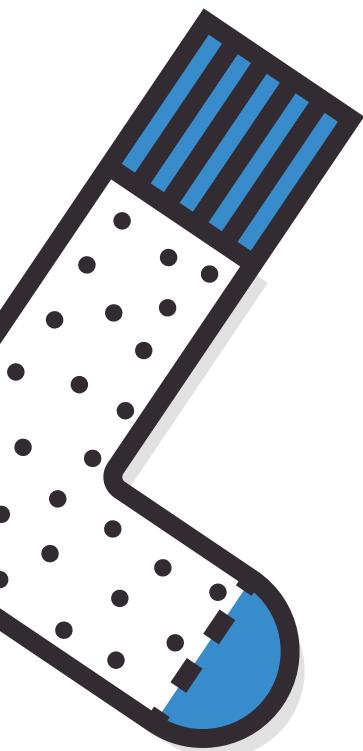
from Nottingham, invented the knitting loom in 1589. He was, however, denied a patent by Queen Elizabeth I, who wasn't impressed with the quality of the knit. That's when France's King Henry IV came to the reverend's rescue and decided to fund his venture, only after Lee moved the manufacturing unit to France.

**"THE EARLIEST INSTANCE OF A KNITTED PAIR WAS DISCOVERED IN A TOMB IN EGYPT DATING BACK TO 1000 CE"**

By the time the Industrial Revolution took over the manufacturing process, socks became easier and cheaper to manufacture, which made them accessible all over Europe. Through the ages, the sock took on many different lengths and patterns, influenced primarily by the prevalent fabric of choice. Cotton became the favourite in the 17th century, and continued for several hundred years. The next major milestone in the evolution of the sock came with the advent of nylon in the 20th century. Nylon gave them much needed elasticity, and quickly became a hit with the masses. The palette of fabrics and designs have increased manifold since.

Although they are designed to be worn on the lower limbs, these ingenious makers can't wait to get their hands on them.

# SOCK DOLLS



**O**ne of the most popular uses for old socks is to convert them into soft toys. The process isn't expensive; all it essentially requires is an odd sock and some skills with thread and needles. Singapore-based artist Lee Wai Han is very passionate about transforming socks into cute dolls. She has been doing it for a long time, and has an impressive portfolio over at DeviantArt ([hsmag.cc/REtshT](http://hsmag.cc/REtshT)). She created the owls with an ankle-length sock that she first turned inside out, before sewing it close to the top and snipping away the rest. She then snipped the toes and filled the sock with cotton, while shaping it and then sewed it shut. This created the basic head of the owl, and she glued eyes, nose, and a mouth to complete the look.

While the exact steps for creating the dolls vary, the general principle remains the same. Also, you can create the dolls with any socks, but by carefully selecting the socks, Lee is able to use the patterns on them to accentuate the design of the doll. You can obviously use a plain white one and just glue or sew on the elements once the basic structure of the doll's head is ready. □

**Right ♦**  
Here's another tutorial ([hsmag.cc/CEQERC](http://hsmag.cc/CEQERC)) where Lee uses lace on the socks to fashion a dress for the bride



**Project Maker**  
**LEE WAI HAN**

**Project Link**  
[hsmag.cc/JZIONo](http://hsmag.cc/JZIONo)

**"SHE CREATED THE OWLS WITH ANKLE-LENGTH SOCKS THAT SHE FIRST TURNED INSIDE OUT"**



# UPCYCLE OLD SOCKS



**Project Maker**  
**CORINA YEN**

**Project Link**  
[hsmag.cc/fbiUNt](http://hsmag.cc/fbiUNt)

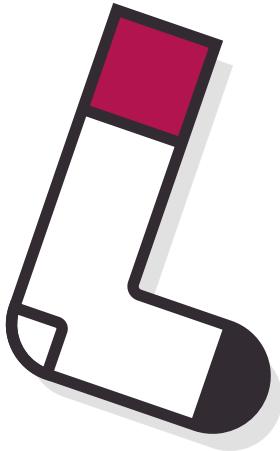


hen you're done making sock puppets, follow Corina Yen's lead and turn old socks into something more useful. In her Instructable, she shows you how to repurpose a

single sock into three useful products. The first is a wristband with a pocket that she calls the sporty sock. It is made from the elastic top and half of the sole portion of the sock. You can use it as an exercising accessory to secure your keys and other useful knick-knacks. Follow Corina's Instructable to sew the pocket that's cut from the sole portion to the tubular top. The heel portion of the sock, which still has some of the elastic bit from the sock's tube, is refashioned as a pull-on coaster or a cup

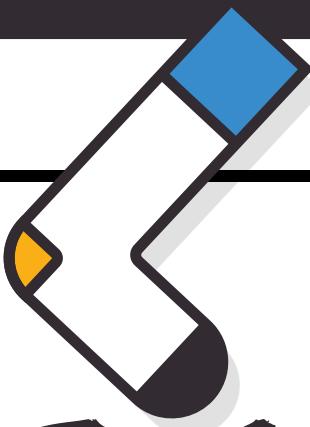
identifier. She calls it the coasty sock, and it'll again need some sewing before it can be used as such. The leftover bit is put to use as a key chain pouch that insulates the keys from scratching the phone, or anything else in your pocket or wallet. Corina has illustrated the process for crafting all three products, including the sewing bits that can be done without a sewing machine, which makes this build accessible to everyone. □

**"YOU CAN USE IT AS AN EXERCISING ACCESSORY TO SECURE YOUR KEYS"**



**Right ♦**  
As if snipping one sock into three different products wasn't enough, Corina used the leftovers to create a small hand puppet





# PHONE CASE



Project Maker  
**KERRY MICHELLE**

Project Link  
[hsmag.cc/fksskQ](http://hsmag.cc/fksskQ)



**W**hen Kerry, a self-professed fan of Pac-Man, was gifted a pair of Pac-Man socks, she couldn't bear the thought of wearing them as such.

She had been saving them for a creative useful hack for sometime, when she decided to use them to create a cover for her new mobile phone. Like all the other builds, this one will require a bit of needlework as well. Kerry first measured her phone against the sock. She then stitched the sock as per the phone's length, and then snipped away the remaining unwanted portion of the sock, saving it for another future hack. Replicating Kerry's hack isn't time-consuming and only requires a simple stitch. You can enhance your sock phone case further by adding a couple of buttons to keep the cords for the earphones neatly organised ([hsmag.cc/wwwYVz](http://hsmag.cc/wwwYVz)). □



**Right ♦**  
Although she only had to stitch one side of the sock, you might have to stitch the sides as well, to make sure your phone is a snug fit

# HEAT PACK SOCKS

Project Maker  
**JUANITA**

Project Link  
[hsmag.cc/IFYVpY](http://hsmag.cc/IFYVpY)

**Right ♦**  
Unlike the other hacks, this one requires a fair bit of stitching to create the pockets



**D**o you have poor circulation in your feet and are your toes always freezing?

As a mum and an adventurer, Juanita understands our predicament and the need to keep our feet warm. Her hack repurposes two pairs of old unused socks into one pair of insulated socks that'll do a lot of good on cold rainy nights. First, turn one pair of socks inside out and wear them. Then wear the other pair over them the right way out, so that the inside of both the pair on either leg are facing each other. Then refer to Juanita's Instructable and use chalk to draw lines on the sock, around the base of the foot and running up. Then, remove the sock carefully and use a backstitch to stitch along the lines. When you're done, you'll end up with a series of pockets in the socks. Fill these with rice and stitch some more, and you're done. Now, microwave the socks for 30 seconds, and put them on. □

# 3 ISSUES from £5

on a quarterly subscription



Visit: [hsmag.cc/345](https://hsmag.cc/345)

# 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  
**84**

### ARTISTIC PCBs



Use Inkscape to create art  
for KiCad PCB designs

PG  
**90**

### LEARN TO BRAZE

Use fire to join bits  
of metal together

PG  
**96**

### ELECTRONIC NAILS

Embedded electronics  
on your hands

PG  
**100**

### HACK SERVOS

Convert these cheap devices  
into continuous motors

PG  
**78**

### SCHOOL OF MAKING

Start your journey to craftsmanship  
with these essential skills

**78 If This Then That**

**82 At a Glance: 3D printers**

PG  
**102**

### COSPLAY WINGS

Add movement and flair to your  
outfit with pop-out wings

PG  
**108**

### SERIAL

Communicate with your devices  
the old-fashioned way

# Send data with If This Then That

Control almost anything with CircuitPython



**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.

If This Then That (IFTTT) is a really simple bit of glue that lets you link a condition with an action, both of which are taken from web services. All IFTTT applets are in the format If 'condition' then 'action'. This simple form is surprisingly powerful because it's linked to a huge number of internet services and web-connected devices. The sheer range of services that can link together, and the fact that everything can be configured by clicking on things in a website, make it a great platform for basic Internet of Things devices and experiments.

If you're using commercial devices and services, such as smart bulbs, weather forecasts, and smart hubs, you can do everything without having to touch a line of code, but you can also use IFTTT with your own devices. There's a service called Maker Webhooks that lets you send data into the service by firing it at a URL. We've created a simple wrapper library that lets you send data to this using CircuitPython. Let's take a look at how to do this.

There are three parts to this project: the input, which will be our CircuitPython device (you can use any device that has WiFi), the IFTTT applet that

The screenshot shows the IFTTT mobile application interface. At the top, there's a navigation bar with the IFTTT logo, a search icon, and a user profile icon with the text "Get more". Below the navigation bar, there's a large blue button with a white circular icon containing three dots. The main content area has a title "Connect Webhooks" and a subtitle "Step 1 of 6". A descriptive text block below the title reads: "Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform." At the bottom of the screen is a large black button with the word "Connect" in white.

**Right ♦**  
Webhooks let you trigger IFTTT applets from any device with an internet connection

The IFTTT home screen features a search bar at the top with the text "Search". Below it is a large yellow banner with a white robot head icon and the text "Spotlight on Litter-Robot".

**Right ↗**  
It's well worth searching through the different services to IFTTT, as there are many that you might not suspect, and there are some great examples to whet your appetite

## SENDING EVENTS

We've used the IFTTT library in this tutorial to make it easy to send events via webhooks, but it's actually fairly easy to send webhooks directly. To understand what we're doing, you need to understand that there are different types of requests that you can send to a web server. Probably the two most popular are GET and POST. The former is what you send when requesting a normal webpage; the latter is usually what happens when submitting a form on a webpage. We can use either in our webhooks, but in order to send data, we have to use a POST.

shunts the data to the right place, and the output. We'll use the IFTTT app on a phone as the output – this lets us send notifications to our phone from a web-connected CircuitPython device.

## GETTING INPUT

Let's start at the beginning of our data flow – the CircuitPython device that will send the data to IFTTT. We'll detect this based on a touch – press the touchscreen, and it will send data to IFTTT. This could easily be used to create a notifier where people can tap a button on a touchpad to get your attention. It could also be used with other sensors to alert you to a particular condition, such as using a soil moisture sensor to let you know when your plants are running dry. The important bit in all of this is the link to your phone via IFTTT, so let's focus on that.

The main loop that does the work is as follows:

```
while True:
    p=ts.touch_point
    if p:
        print("touched")
        ifttt.send_message(wifi, secrets,
                           "circuitpylink", debug=True, value1="touchy touchy")
```

The IFTTT mobile app interface shows a "Webhooks" screen. A blue card displays a recipe titled "If Maker Event "circuitpylink", then Send a notification from the IFTTT app" by beneverard1. The card includes a toggle switch and notification icons.

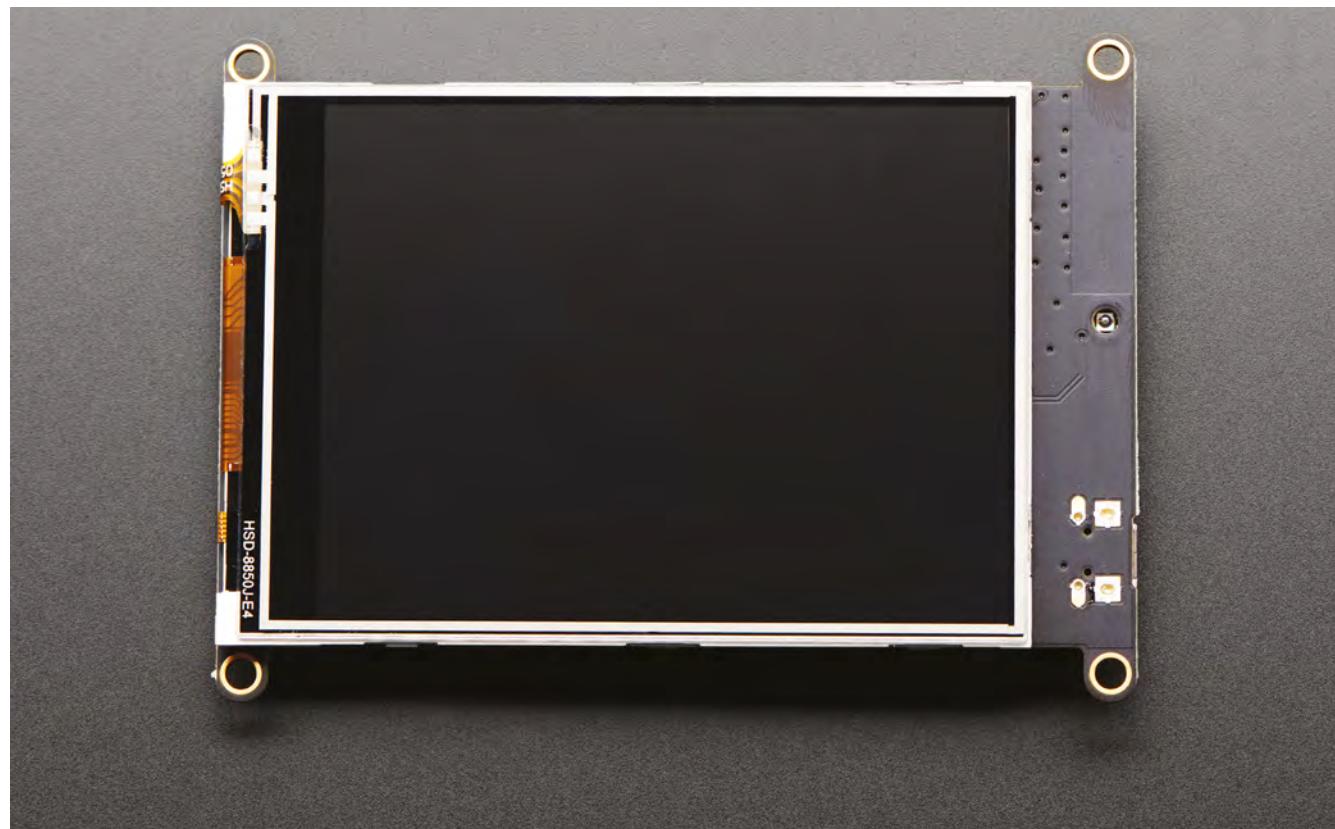
```
updated")
time.sleep(10)
```

This requires a few bits set up in order to work. The `ts` object is the touch sensor for the screen – it's set up with the following line:

```
ts = adafruit_touchscreen.Touchscreen(board.TOUCH_XL, board.TOUCH_XR,
                                       board.TOUCH_YD, board.TOUCH_YU,
```

**Above ↗**  
You can link lots of different events to the phone app, so it makes a great source for all your notifications

## SCHOOL OF MAKING



You can have as many different event types as you like **associated with your account**, each triggering a different applet

```
calibration=((7000,59000),(8700,55000)),  
size=(320,240))
```

The WiFi object is an `adafruit_esp32_wifimanager` object that's created with the following:

```
esp32_cs = DigitalInOut(board.ESP_CS)  
esp32_ready = DigitalInOut(board.ESP_BUSY)  
esp32_reset = DigitalInOut(board.ESP_RESET)  
spi = busio.SPI(board.SCK, board.MOSI, board.  
MISO)  
  
esp = adafruit_esp32spi.ESP_SPIcontrol(spi,  
esp32_cs, esp32_ready, esp32_reset)  
  
status_light = neopixel.NeoPixel(board.NEOPixel,  
1, brightness=0.2)  
wifi = adafruit_esp32spi_wifimanager.ESPSPi_  
WiFiManager(esp, secrets, status_light)
```

Finally, `secrets` is a dictionary containing things that you don't want to share publicly. In this case,

it needs to include your WiFi SSID, WiFi password, and IFTTT key (more on this later). You'll need a file on your CircuitPython device called `secrets.py` that contains the data structure in the following form:

```
secrets = {  
    'ssid' : 'XXX',  
    'password' : 'XXX',  
    'ifttt_key' : 'XXX'  
}
```

You can bring this into your main `code.py` program with the following code:

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

You can see how this all comes together as this is the example on the IFTTT library here: [hsmag.cc/XzdCAs](https://hsmag.cc/XzdCAs).

You'll also need to include the MPY file for the IFTTT module. You can download this from [hsmag.cc/IFqTxx](https://hsmag.cc/IFqTxx). Make sure you get the version that matches the version of CircuitPython you have. Unzip the file and copy the MPY file to the `libs` directory on your board.

## WEB OF THINGS

We're almost there with the CircuitPython board, so let's turn our attention to the setup on IFTTT. First, go to [ifttt.com](https://ifttt.com) and create a free account.

Scripts in IFTTT are known as applets. You can create many of them in your account and have them running concurrently. To create a new applet, click on your profile in the top-right, then select Create. In the new screen, you should see, in big letters, If This Then That. Clicking on 'This' will let you select the action that you want to trigger your applet. Select 'Webhooks' and then 'Receive a web request'. Web requests are 'events' in IFTTT lingo. You can have as many different event types as you like associated with your account, each triggering a different applet. However, so that IFTTT can tell which is which, each needs a name. We called ours 'circuitpylink' (you might recognise this from the code above), however, you can use any text string you like (we had difficulties with strings with spaces in, so we'd recommend avoiding these).

You should then be taken back to the If This Then That screen, and you can click on 'That' to select the action to perform. Select Notifications, then 'Send a

## OTHER IFTTT INPUTS AND OUTPUTS

There are other ways of getting data into an IFTTT applet. You can scroll through the huge list when you press the 'Plus' button, but here are a few of our favourites:

**Adafruit** – this links your IFTTT applet to an Adafruit.io feed. This way, you can feed data into your trigger while still retaining a copy to use elsewhere.

**Button Widget** – this works the opposite way to our applet, in that data goes from the IFTTT app to wherever you want it to.

**GitHub** – start applets when events happen to your source code.

**Twitter** – tweet your applets into life.

You can also send data to a wide variety of sources including:

**Android SMS** – this uses your phone to text details to whoever you like. It uses your SMS plan, so this may cost you money, and it'll only work if your phone has an internet connection.

**Email** – send a message to anywhere in the world.

**littleBits** – control your littleBits hardware.

**Google Sheets** – save your data in an online spreadsheet.

Notification From the IFTTT App'. Here you can enter the text that you want to appear in your notification. As well as plain text, you can add 'ingredients' which are tags that will be replaced by data that's sent. IFTTT lets you send three values names: Value1, Value2, and Value3 – we'll only use the first.

The message for our notification is then:

**Message from CircuitPython: "{{Value1}}"**

That's the glue for our system completed. The only thing left is the final part – we need to set up our phone to receive and display the notification. Fortunately, this is easy. Just install IFTTT from your phone's app store and log into it.

## KEYS TO SUCCESS

We're almost all the way there now. The only thing left is the IFTTT key that we need in order to authenticate our requests. This lives in the **secrets.py** file. We've left it until now because we need to get it from the IFTTT app. We don't know why you have to get it from the app rather than the webpage – it's possible that this is a bug and will change by the time you read this as the online help suggests that you should be able to get it from the webpage.

Go into the app and select Webhooks, then click on the cog in the top left. You should see your account info here which includes a user\_url which will be in the format: **maker.ifttt.com/use/<thisisyoursecretkey>**.

**" After a few seconds, you should see a notification appear on your phone "**

Copy the text that's in the place of <thisisyoursecretkey> into the appropriate place in your **secrets.py** file. Every time you make changes to your webhooks setup, your key changes, so if you stop getting messages through, it's worth checking that your key hasn't changed.

Once that's in place, everything is set up. Touch the screen of your PyPortal (or do whatever you've set up to trigger the message), and after a few seconds, you should see a notification appear on your phone. If it doesn't, check that both devices have internet access, and the IFTTT key is correct.

While we've covered a very trivial use in this article, this method of sending data to IFTTT from CircuitPython devices gives you a powerful way of linking builds together across the internet and monitoring them in real time. ☐

# At a glance: 3D printing

Find out what makes up a desktop fabricator



**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.

**F**used filament fabrication or FFF, (sometimes called under the trademarked term Fused Deposition Modelling or FDM), is the most common form of 3D printing. It can work in a few ways, but by far the most common is a printer taking a filament of plastic, heating it until it's soft, and placing it on a print bed, building up an object layer by layer. FFF printers differ a bit in their exact setup, but they all have the same basic parts. Let's take a look at what they are.

## STEPPER MOTORS

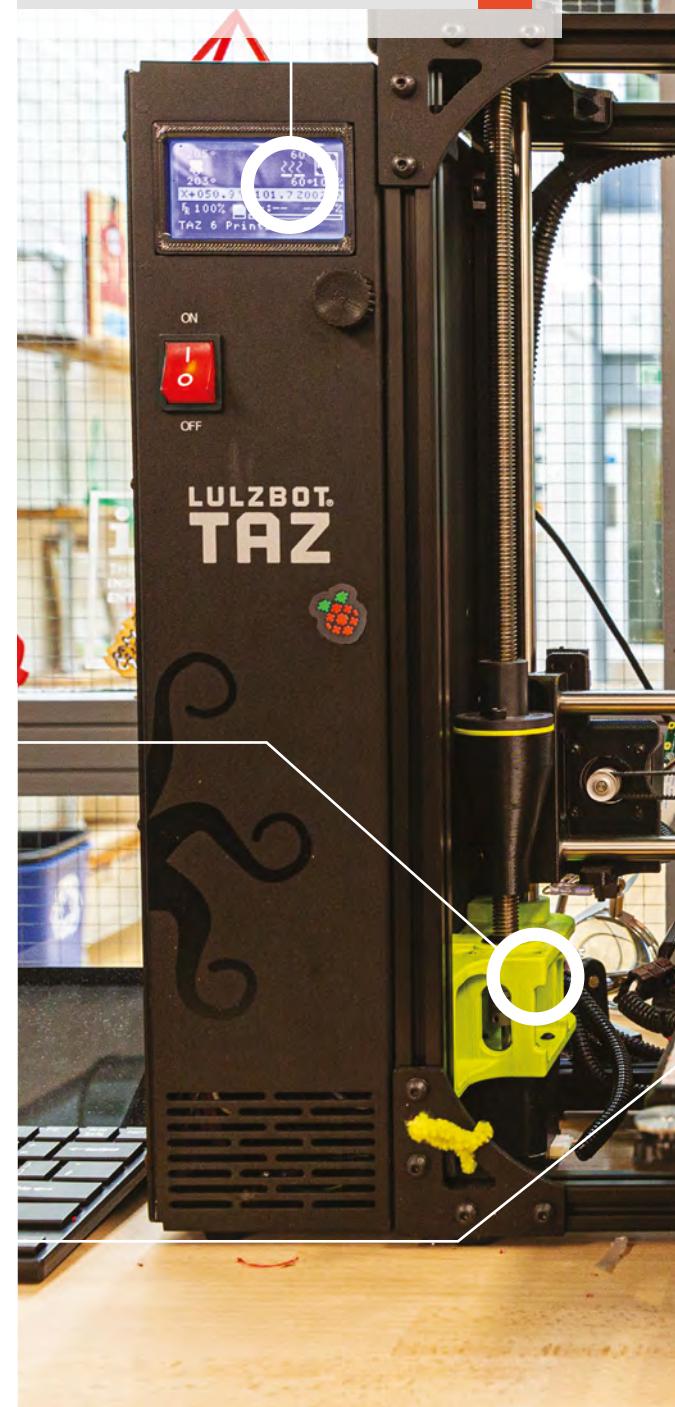
The extruder is positioned by three sets of stepper motors – in the X, Y, and Z axes. On some printers, the extruder is moved while the print bed stays still, in others the print bed may move in one or more axis. Some printers have more unusual set-ups of motors. The gearing and model of the motors affect both the speed and accuracy of the printer.

## FANS

Molten plastic comes out of the extruder in the correct place, but if it stays molten, it won't stay in the right place for long. Fans help the plastic cool and solidify quickly, which helps you print in finer detail. It might sound like this is the exact opposite of what we said about a heated print bed, but actually, they're both part of the same thing – controlling the cooling process. The more control you have over it, the better your 3D printing experience will be. Some people go even further, and put their printers in heated boxes where they can control the air temperature around the prints, but this is only needed for exotic filaments.

## CONTROL SYSTEM

Your printer needs something to tell it what to do. Usually, these take files in G-code format and convert them into the physical movements of the motors. Some printers require you to transfer your files over on an SD card, while some let you plug your computer in via USB. Note that some printers may need to be connected to a computer for the entire time of the print.



**FRAME**

Everything is held together in a frame. There are many materials this can be made from, but the rigidity of the frame is critical to the accuracy of the printing. The stiffer the frame, the more accurately you'll be able to print.

**EXTRUDER**

There are three basic parts to an extruder – the motor that pulls in the filament, the heater that melts it, and the nozzle through which it's pushed into the printer. All three functions are critical. If the motor can't grip the filament properly, it can slip, and this causes print problems. The heater has to be held at the right temperature for the material being used (different filament types need different temperatures, and not all printers can get hot enough for all filament types). Nozzles can wear out, particularly if you use one of the more abrasive filaments.

**FILAMENT**

This is the material that your printer makes objects from. The most important thing is to make sure that your printer is compatible with the filament, and the most important points here are the temperature your printer can achieve, and the diameter of filament your printer can accept.

There's a dizzying array of filaments available, and they vary in terms of colour, strength, flexibility, and look. Most printing filament is made of polylactic acid (PLA) a bioplastic that's fairly easy to work with. One problem with PLA (and some other filaments such as ABS) is that it absorbs moisture from the air, and this causes problems when printing, so it needs to be stored in as dry a place as possible.

**PRINT BED**

This is the bit that is printed on. While it may sound basic, there are a few things to consider. Firstly, the print bed has to be exactly level. Each layer of print is only a fraction of a millimetre thick, so if the print bed is off by even a tiny amount, it can cause problems with the print. Some printers require you to level it manually (such as using screw adjusters), while others have auto-levelling.

You print hot plastic and then it cools to solidify. If you print hot plastic directly onto a cold print bed, this can cause problems like warping. Having a heated print bed gives you more control over the cooling process, and results in better prints.

**REEL HOLDER**

You get filament in reels, and this needs to feed freely into the printer. Many printers ship without reel holders, and printing your own holder is a common first project for 3D printer owners.

# Make artistic KiCad PCBs with Inkscape

Use `svg2shenzhen` to give your circuits a little more style



**Jo Hinchliffe**

@concreted0g

Jo is a contributor to the Libre Space Foundation, 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, milling machines, and CNC kit!

**T**here are some amazing art PCB projects out in the world, and approaches to making non-standard shaped PCBs have varied and often been complex. `Svg2shenzhen` is an Inkscape extension that allows Inkscape to draw directly on KiCad equivalent layers and export these from Inkscape as KiCad projects. These projects can either be sent directly to manufacture, or can be tweaked and added to using the usual KiCad workflow. This enables PCBs to be created that couldn't be made using KiCad alone.

To get started, we need to navigate to the amazing badgeek's GitHub repository for `svg2shenzhen` here – [hsmag.cc/BQpUAJ](https://github.com/BQpUAJ). Click the green 'Clone or download' button in the upper right-hand corner, and then select 'Download ZIP'. Once it's downloaded, open Inkscape and double-check where your user extensions are installed. Do this by clicking Edit > Preferences > System and then note the location of 'User extensions' (**Figure 2**).

Close Inkscape and then unzip the downloaded `svg2shenzhen` zip file to the location you found in the previous step. When you reopen Inkscape,

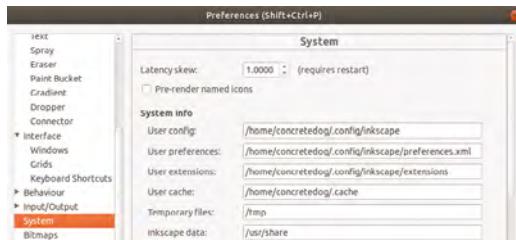


**Figure 1** The completed PCBs, made using a combination of Inkscape, `svg2shenzhen`, and KiCad

## QUICK TIP

After installing libraries, it's a good idea to close and restart the Arduino IDE.

**Figure 2** ◇  
Using the Inkscape preferences menu to locate where to install extensions



you should find that, if you select ‘Extensions’, `svg2shenzen` will appear in the drop-down menu.

Hovering over `svg2shenzen` in the drop-down menu should reveal the extension has three options: ‘Help’, ‘Prepare document’, and ‘Export to KiCad’. Click the ‘Prepare document’ option. You should see a dialog box in which you can set the size of the document. It’s not critical, apart from it needs to be big enough to contain your desired design; input a size and then click ‘Apply’. Close the dialog box.

If you now open the layers window, either by clicking Layer > Layers or with the **CTRL+SHIFT+L** keyboard shortcut, you can see that the Inkscape document has now been set up with numerous layers that match some of the physical layers of a PCB, named as they are labelled in KiCad.

Let’s begin by drawing on the ‘Edge Cuts’ layer – this is the layer into which we draw the edges of the PCB. Within KiCad we are limited to arcs/circles and straight lines, whereas in `svg2shenzen` we have no limitations to the geometries we can draw. To draw the outline of the rocket PCB (**Figure 1**), we used the Bézier tool and some common path tools in Inkscape, including Union and Divide. A rule for edge cuts created with `svg2shenzen` is that they must be a single path that is completely connected. It’s easy when drawing perhaps with multiple tools in Inkscape to create an outline that has a couple of unconnected nodes, making it incomplete.

Once you have made an outline shape of your PCB, you might next want to add some details, shapes, and lines that will appear as exposed copper when the board is manufactured. For the rocket PCBs, we wanted to also add some components

## SEND DIRECT

In this tutorial, we are going to export the design from Inkscape using `svg2shenzen`, and then further manipulate the design using the KiCad regular workflow. However, `svg2shenzen` is also designed to be able to create complete art PCBs in KiCad that could be sent directly to manufacture. The authors of `svg2shenzen` use PCBWay and even have an option in the export dialog box to open the PCBWay website after export. PCBWay needs you to upload separate Gerber files from KiCad for manufacture. Later in this tutorial, we will use OSH Park to order PCBs, as the site directly takes the KiCad ‘PCBnew’ file and doesn’t require Gerber files to be generated. It also has the benefit of providing in-browser renders of the layers of the board, which is useful for double-checking your design.

and traces to the front copper layer in the standard KiCad workflow, so we decided to add the more decorative items to the back copper layer of the board. Before we add some decoration, let’s look at how the mask layers work. Currently, the mask layers are set to ‘disabled’. To enable them, simply select the layer and rename the layer by removing the ‘-disabled’ text. For example, the back mask layer would change from ‘B.Mask-disabled’ to ‘B.Mask’. This actually (due to the nature of PCB fabrication) sets the mask layer to not appear where you draw a black item. In this configuration, you have to manually draw where you want mask to not appear. Changing the mask layer name to ‘B.Mask-invert’ sets this layer so that what you draw is where you get mask on the PCB, but for the rocket project,

we found the simplest solution was to change the mask layer to B.Mask-auto. In auto mode, when you add an object to the corresponding copper layer (B.Cu in this case), it automatically

removes the mask in that area, allowing the object you have drawn to be seen through the removed mask as copper.

### LET’S GET MASKED!

Having set the B.Mask layer to auto-select the B.Cu (Back Copper Layer) and draw our items onto this layer, we add the copper rounded triangular sections seen on the rocket (**Figure 1**), and it automatically adds these shapes onto the mask layer to allow them to be revealed through the solder mask. This means that if you want to disable the visibility of the B.Cu layer in Inkscape (we will do so later when we want to place an object on the other side of the PCB), you will need to close both the B.Mask-auto →

### YOU’LL NEED

- ◆ A laptop with KiCad and Inkscape installed
- ◆ Some familiarity with basic use of Inkscape

Once you have made an outline shape of your PCB, you might next want to add some details, shapes, and lines

### QUICK TIP

Whilst on the `svg2shenzen` GitHub repository, check out the images of amazing designs made in `svg2shenzen` for inspiration.

## TUTORIAL

### QUICK TIP

A rule for the edge cuts layer is that you shouldn't group items. If you do use groups, make sure you ungroup and make it a single path before exporting the project from Inkscape.

and B.Cu layer visibility 'eye' icons to make the shapes disappear.

Next, we enabled the B.Silk layer by removing the '-disabled' text and add the HackSpace magazine logo icon. As we are working on the back layer of the PCB, we need to imagine that the PCB is lying on a surface and we are drawing on the bottom of the PCB. Considering this, we realise that we need to reverse any items we place on the back layer for them to appear correctly on the PCB.

### SORTING THE SILKSCREEN

As silkscreen layer items are generally a single colour (often white), we made the HS logo by subtracting the HS

letters from a solid filled circle, so that on the finished board we will see the HS letters as the underlying mask colour. Finally, using the horizontal flip tool, we reversed the HackSpace

magazine logo and then added it to the B.Silk layer. One layer we haven't used in this tutorial is the drill layer. In *svg2shenzhen* it is treated largely in a similar way to other layers. A slight difference when placing a drill point in *svg2shenzhen*, rather than in KiCad, is that drawing a circle creates a simple hole and we need to remember to add a copper pad surrounding it if needed.

The last job in Inkscape (other than corrections later!) was adding the concreted0g logo to the other side of the board. As this was placed on the front silkscreen layer (F.Silk), it did not need to be reversed. However, we did want to place it in a position where currently it couldn't be seen in

As a test to see if it worked, we added a simple circuit that is a breakout board for an ATtiny85 microcontroller

**Figure 3** Toggling the visibility of copper and mask layers so you can see areas of silkscreen layers

**Figure 4** Opening the project in KiCad for the first time allows you to see your *svg2shenzhen* design in the KiCad 3D viewer

Inkscape due to the placement of items on the B.Cu and B.Mask-auto layers. Simply toggling the visibility eye icons removes these items from view, allowing us to place the logo correctly (**Figure 3**).

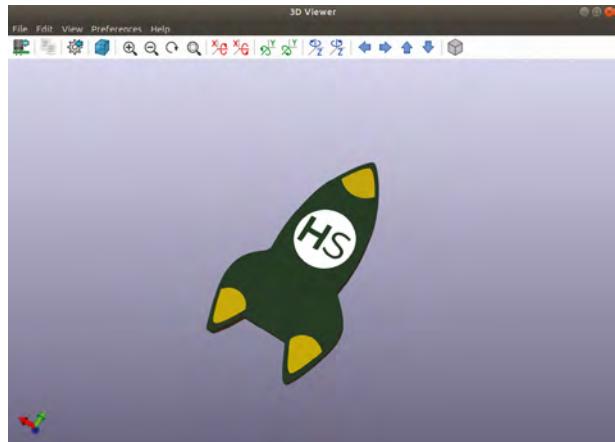
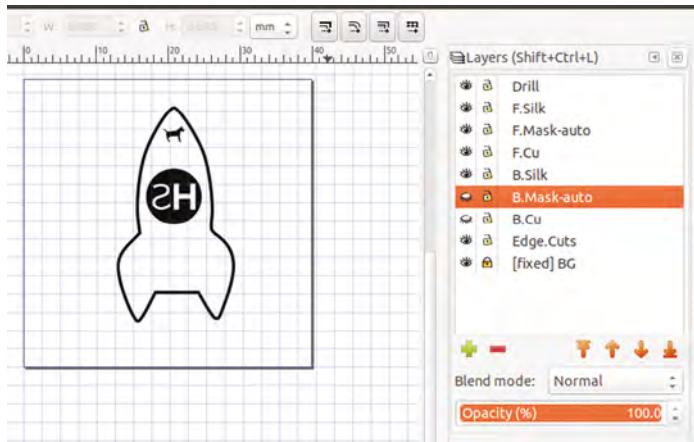
Having finished everything we need in Inkscape for this project, we need to use the *svg2shenzhen* extension to export the project into a KiCad project file and PCBnew file to allow us to continue the work in KiCad. Save your work first, if you haven't done so already, and then click Extensions > *Svg2Shenzhen* > Export KiCad. You should see a dialog box with numerous options. The first is the path to where you want the files to be exported. The second dropdown menu allows you to select between exporting

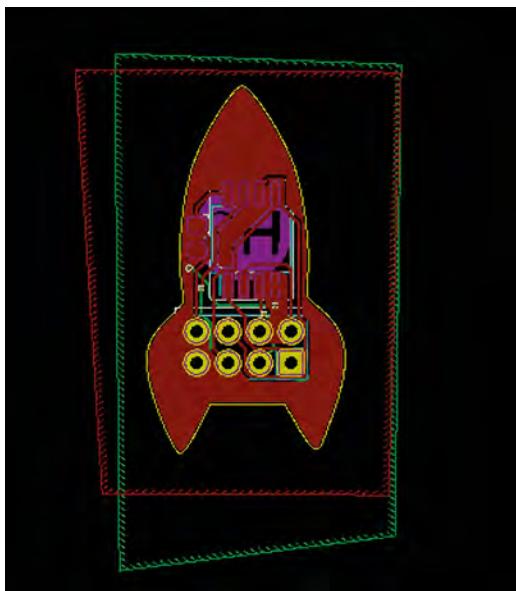
as a KiCad project, a KiCad module, or a PNG. We are going to set this to the first option so that it creates a KiCad project and a PCBnew file for us to open in KiCad.

The second option is useful if you want

to use *svg2shenzhen* to design a module that you can export and use in any KiCad projects. We left the 'threshold' and 'export dpi' options set as the defaults, and also left 'auto flatten Béziers for edge cuts layer?' checked.

The debug mode, when checked, gives some debug messages when you export which may be useful for debugging or reporting back to the *svg2shenzhen* team if you discover an issue, but may be unchecked for most usage. As we aren't using *svg2shenzhen* to do the entire design (we are going to add some components in KiCad) and as we don't want to use the PCBWay service, uncheck the 'launch PCBway' option. Finally, you can set the





**Figure 5** ◊  
KiCad recognises the edge cut lines made in Inkscape and floods inside them correctly

dialog to automatically launch the exported project in KiCad. Check this option and then click Export.

You should now see KiCad boot, and it should open the PCBnew application and have your design already there. One of the first useful things to do is to check that everything is as it should appear by clicking View > 3dViewer and you should see a rotatable 3D model of your PCB as it stands (**Figure 4**).

Whilst it is totally possible to build an entire PCB in Inkscape using `svg2shenzhen`, we were interested to see if the `svg2shenzhen` output could be added to using the typical KiCad workflow to enable a combination of two approaches to PCB design.

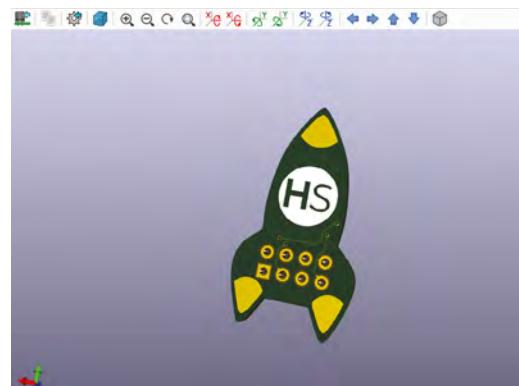
As a test to see if it worked, we added a simple circuit that is a breakout board for an ATtiny85 microcontroller. First, we saved our work in PCBnew and closed it and opened Eeschema. Opening Eeschema gives a dialog box that the project does not contain a schematic file, and asks if we want to create one. Having created one, we have a blank Eeschema document. Into this, using standard KiCad workflow and libraries, we added an ATtiny85 component, power and GND inputs, a bypass capacitor across the VCC and GND pins on the microcontroller, and an 8-pin header to which we are going to break out the ATtiny.

We then wired the schematic using the wire tool and again, as we did in the KiCad tutorials of issues 17 and 18, annotated the schematic, assigned footprints to the schematic symbols, and created a netlist.

Moving back to PCBnew, we then read the netlist and that imported the components and the rat's nest perfectly, whilst still keeping the elements we drew in Inkscape. Perfect! We then laid out and routed the PCB with the new components. We were interested to see that KiCad correctly observed the board edge cuts, particularly when it came to flooding the PCB areas with copper fill zones. As you can see from **Figure 5**, it's simply a case of using the 'Add Filled Zones' tool to draw an area larger than the PCB and KiCad will detect the board edges and pour the filled area within the PCB edges. Finally, it's also worth noting that if you notice at any time something you need to correct in Inkscape (in our example we had placed the HackSpace magazine logo too low initially as it clashed with the header pins), you can go back to the saved Inkscape file, make a change, and re-export the KiCad layers from Inkscape and overwrite the earlier files. This won't affect the Eeschema documents or netlists within the KiCad project file, but it will overwrite the PCBnew file. This means that you will lose any work you have undertaken in PCBnew (our added components and routing for example), so it's worth triple-checking you are happy with the Inkscape layout parts of the board before adding lots of work in KiCad.

Having completed the design in both Inkscape and KiCad, it's worth double-checking the board again in the 3D viewer before placing your order (**Figure 6**). If you are using OSH Park, which allows you to directly upload the PCBnew file, then after finishing your design you simply need to save your work in KiCad and upload.

We were really impressed with `svg2shenzhen`, finding it a highly usable piece of software and indeed it makes creating complex or organic shapes for KiCad simple to achieve. If you give it a try and create something, make sure to send us a picture or share a tweet with us @[hackspacemagazine](#). □



## QUICK TIP

You can use the KiCad 3D viewer to check everything is correctly laid out and orientated on the different layers.

## QUICK TIP

All the techniques used in KiCad in this tutorial are included in the KiCad tutorials featured in HackSpace magazine issues 17 and 18.

# Wireframe

Join us as we lift the lid  
on video games



Visit [wfmag.cc](http://wfmag.cc) to learn more

# WIN!

## A Strafer Chassis Kit

courtesy of goBILDA



Go to  
[hsmag.cc/win](http://hsmag.cc/win)

WORTH  
**\$499**

Enter by 1 November  
for your chance to win!

### Terms & Conditions

Competition opens on 18 September 2019 and closes on 01 November 2019. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed, or administered by, or associated with, Twitter or Facebook.



# Basic brazing: Tin can caddy

Learn to braze and make a caddy for your brazing rods



**Dr Andrew Lewis**

@monkeyssailor

Dr Andrew Lewis is the owner of [Shedlandia.com](http://Shedlandia.com), a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.

**B**razing is an extremely useful technique for joining metal, and while people often say that 'welding is stronger than brazing', that's not strictly true. It all depends what metals you're trying to join, and what type of brazing rod you're using to join them together. In this project, you'll learn how to gas braze some old cans together to make a handy container to store your brazing and welding rods.

## BRAZING ISN'T WELDING

Coffee cans with plastic lids are great for storing things in, but they're too short to store welding rods or other long items. Joining multiple cans together is a great way to learn how to braze, and illustrates some of the advantages of brazing over soldering and welding.

Start by preparing your coffee cans. Lots of the cans available in the supermarket have a metal lip at



## Above

Brazing equipment comes in all different shapes and sizes, and the tools that you use will depend on personal preference and the job that you're working on

**Above and right**

Clean paint from the cans with a wire brush to get the best joint, and remember to cut the bottom out of the can before you braze it in place

**BRAZING SAFETY**

Working with anything hot carries certain dangers. The most obvious one is burning yourself, so make sure you don't touch hot metal. All smoke and fumes are bad for you and some are more dangerous than others. Ensure that the place you're working is well ventilated and make sure that any smoke goes away from your face. Make sure that you're aware of any risks with the particular metals you're joining. Avoid working with anything containing zinc unless you have the appropriate skill and equipment.

**QUICK TIP**

Certain types of metal release toxic and corrosive fumes when heated. Check what's safe, wear appropriate safety gear, and work in a well-ventilated area.

**Above**

It's best to keep your brazing rods labelled, as different alloys can look very similar to each other. Here, you can see silicone-bronze (top), brass (middle), and 55% silver rods (bottom)

the top where the foil seal attaches. You'll need to use your can-opener to remove the lip from all of your cans, and then remove the bottom of all of the cans but one. The can with the base in will be used as the bottom of your storage tube, and the other cans will be brazed on top of this can.

Whenever you're preparing to braze something, it's important to make sure that your joining surfaces are clean, dry, and have any surface coatings removed before you start trying to heat them up. Use a wire brush or some abrasive wool to remove any paint or other coatings from the inside and outside of the can where they will be joined, and remove any oil, dirt, and rust that you can see.

**"** You'll need to use your can-opener to remove the lip from all of your cans, and then remove the bottom of all of the cans but one **"**

Stack one of the bottomless cans on top of the can with the base still attached, and make sure that it sits neatly in place. Most cans have a slightly reduced base that allows them to be stacked on top of each other easily but, if your cans don't hold firmly in place, you can use a magnet to keep them steady while you're working. Just make sure that the magnet is positioned well away from the heat source, as too much heat will ruin it.

Now you have one can stacked on top of the other, you can apply some flux and get ready to start brazing. You have some choices about which type of flux and brazing rod that you use. If you're familiar with soldering, you probably know that flux stops the heat of the gas flame from oxidizing the metal you're trying to join, and that helps the filler metal you're using to flow nicely around the joint. If you want a very smooth joint, you can paint all of the way around the part that you are joining with a liquid flux, and use a silver alloy brazing rod. The silver alloy flows very easily and evenly into the heated joint by capillary action. If you're not worried about the joint looking →

**YOU'LL NEED**

- ◆ **Propane or MAPP gas torch**
- ◆ **Brazing rods** (silver or brass is fine)
- ◆ **Brazing flux** (powder or liquid)
- ◆ **Steel coffee cans** (or similar)
- ◆ **Can-opener**
- ◆ **Wire brush or wheel**
- ◆ **Needle-nosed pliers**
- ◆ **Spray paint**
- ◆ **Personal protective equipment** (safety glasses, flame-resistant clothing)

# Basic brazing: Tin can caddy

## TUTORIAL



### PICKING YOUR GAS

Different types of gas burn at different temperatures. Butane is the coolest, followed by propane, MAPP, and then acetylene. For smaller projects, you can get away with using butane or propane torches with thin (about 1.2 mm) brazing rods. For larger projects, you will need the extra heat of MAPP or acetylene to get the joint hot enough to melt the brazing rods. Hotter isn't always better, though, and you can completely melt small projects with an aggressive gas flame. Remember that when you're brazing, you're trying to melt the filler rod and not the metal that you're joining.

### QUICK TIP

Metal conducts heat. Make sure you're working on a heat-proof, non-flammable surface, and be careful what you touch!

### JOINING THIN METAL

Trying to weld very thin metal is very difficult, and often results in 'blowout', where the power of the arc or flame is so high that sections of metal melt away leaving a hole. This is a particular problem when you're trying to weld thick metal to thin metal, and while there are all sorts of techniques to counteract this, it's not easy. Brazing doesn't rely on melting the metals that you're joining, so blowout is much less likely. In fact, if you manage to get blowout, you were heating the metal far too much. The brazed joint will be weaker than a welded joint, but that doesn't mean it'll be too weak to do the job.

perfect, you can use powdered flux and a brass or bronze rod to just 'tack' around the can in two or three places. Powdered flux looks like fine, dry sand and will stick to a hot brazing rod. The easiest way to use flux powder is to preheat your brazing rod with the gas torch, and then dip the rod into a jar of the powder for a second before removing it. You can then use the flux-coated rod on the joining surfaces, and re-dip the rod whenever you need more flux. The instructions in this article assume that you're using brass rod and flux powder for the rest of this project, because it's the cheapest and easiest way to braze.

The basic technique for brazing is the same as soldering. You need to heat the metals you're joining, and use them to melt the brass filler rod. You're not trying to melt the brazing rod with the flame. Start by using your gas torch to heat up the join between the two cans. Try to apply the heat evenly all around the joint at first, and then linger for a few seconds on the area that you want to melt the brazing rod to get it up to temperature. Heating all around a joint is good practice because metal expands when it heats up, and will distort if it's heated unevenly. It's probably fine with a tin can, but when larger projects cool down unevenly, the tension created by the distortion can be strong enough to break the brazed joint.

### NOT TOO HOT, NOT TOO COLD

Keep the tip of your gas torch pointed at the joint, and touch your flux-coated brazing rod to the section you're heating. The alloy should melt quickly, and flow into the joint. Push more of the brass rod into the joint as it melts, moving along to increase the length of your braze, re-dipping the rod in flux powder as needed. If the brazing rod seems to just slump or just sit on top of the joint without melting, then you haven't applied enough heat. Either go slower, use a hotter flame, or use a thinner rod.

Continue brazing all the way around the joint, and then allow it to cool. Test the strength of the joint by trying to pull the cans apart and grunting dramatically, and then you can continue stacking cans and brazing them until you reach the desired length for your container. Once you're finished, clean the joints with a wire brush or abrasive pad, and spray the outside of the can with acrylic paint to make it look pretty. If you're feeling particularly efficient, you can use different colours to indicate different types of welding rod.

You can start experimenting with brazing other projects and build your skills. Look into using different types of brazing rods and finding out the strengths of each type. There are some special rods for joining specific metals like aluminium and stainless steel, and there are other general use rods that have different temperature and flow properties. Silver brazing rods flow very well when they're heated, and that means

**You can start experimenting  
with **brazing other projects**  
and build your skills**

they can penetrate a joint through capillary action. The more silver the rod contains, the better it will flow. This means that high silver (about 45%) alloys are best for joining where you have tightly fitting parts, but aren't very good for filling larger gaps. Low silver (about 30%) alloy rods, bronze rods, and brass loaded rods are better for filling gaps. You can even build up layers by keeping careful control of the heat and rate you feed the alloy in. □

## TERMINOLOGY

Words often get misused when it comes to brazing, welding, and soldering. Hard soldering is actually silver brazing, and is sometimes called (incorrectly) silver soldering. Silver brazing uses an alloy with roughly 50% silver. However, silver solder is much weaker and uses about 5% silver. Cast iron welding is usually a brazing operation using a nickel rod, although it is possible to weld cast iron given the right tools, a controlled heating and cooling process, and special welding rods. In contrast, brazing rods for cast iron are more thermally stable, and lower heat means there's less chance of the stress causing the iron to crack as it cools.



### Above

Get the joining surfaces hot, and use them to melt the brazing rod. Try to keep the heat even, so that the surfaces don't warp too much



### QUICK TIP

Hold the brazing rod with needle-nosed pliers to keep fingers safe.

### QUICK TIP

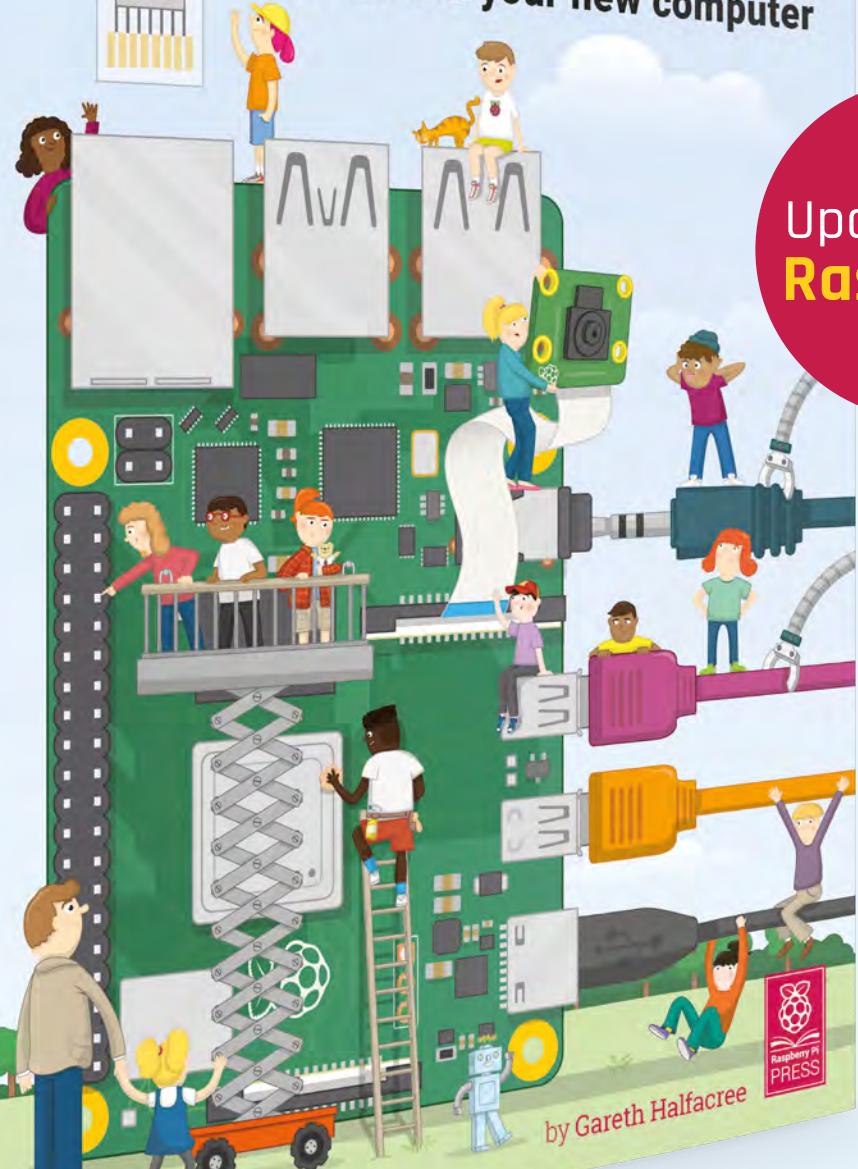
You can also use brazing to create a decorative effect by applying a thin coat of alloy over the top of a base metal.

2nd  
Edition

FULLY UPDATED FOR RASPBERRY PI 4

THE OFFICIAL  
**Raspberry Pi**  
Beginner's Guide

How to use your new computer



Fully  
Updated for  
**Raspberry  
Pi 4**

The official Raspberry Pi Beginner's Guide

2nd Edition

by Gareth Halfacree



# THE OFFICIAL Raspberry Pi Beginner's Guide

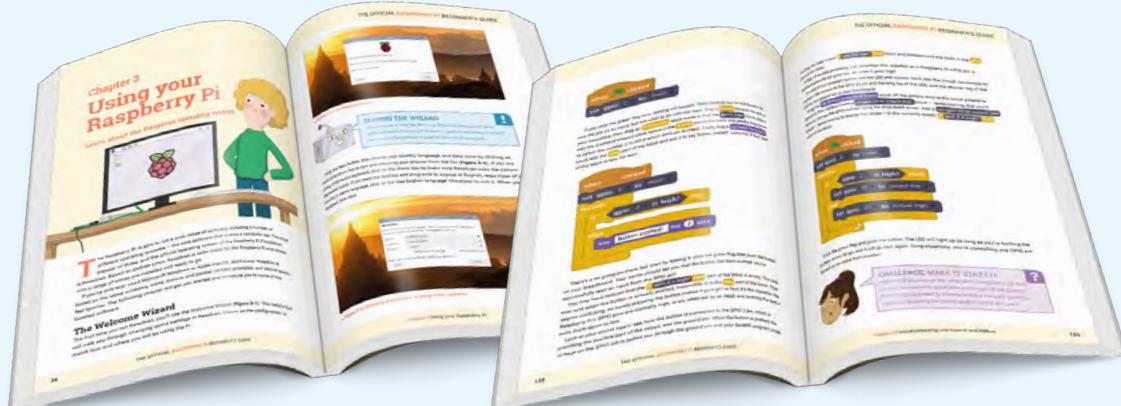
**The only guide you need to  
get started with Raspberry Pi**

## Inside:

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

**Plus much, much more!**

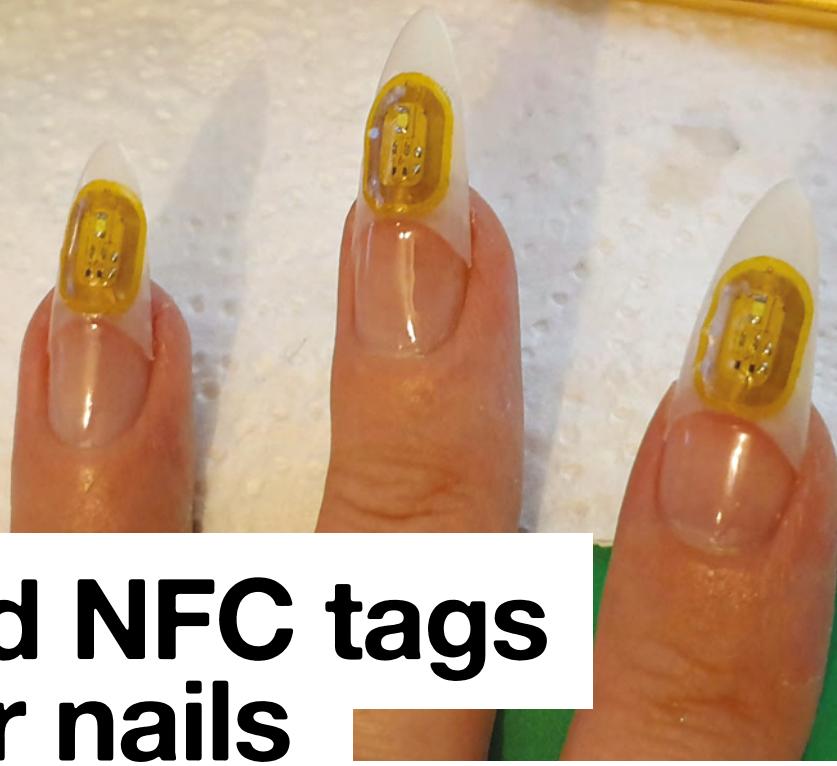
**£10 with **FREE** worldwide delivery**



**Buy online: [magpi.cc/BGbook](http://magpi.cc/BGbook)**

Embed NFC tags in your nails

TUTORIAL



# Embed NFC tags in your nails

Add electronics to your hands and become a cyborg (for four weeks)



Tanya Fish

@tanurai

Tanya has a day job as a crew member at Pimoroni – making learning materials for schools, devising and running workshops, and writing tutorials. In her spare time, she makes stuff because she can't help herself, enthuses loudly about all things STEAM, and herds chihuahuas. She also lectures for Sheffield Hallam University on emerging technologies and has experience as a conference speaker and technology presenter at STEM events and digital festivals.

**R****FID implants, including Amie DD's Tesla-opening one in her arm, are rapidly becoming more mainstream.** However, some biohacking has been mainstream for longer, with varying degrees of permanence. Merriam-Webster defines biohacking as "biological experimentation done to improve the qualities or capabilities of living organisms." We've been putting cochlear implants and pacemakers in people for decades.

However, you don't have to go to such extreme lengths to get started: putting functional tech in your nails is a form of biohacking, albeit 'soft' biohacking. By attaching something to your nails you can have it embedded and part of you at all times... until the nail grows out, at least. For a three- to four-week commitment, we look at how to make a small circuit waterproof, scuff-proof, and perhaps workshop-proof (as long as you don't use acetone a lot).

## PREPARATION

As with all fabrication, preparing the working surface is an essential starting point, and it's no different with nails. Just as you would prepare a wall for painting, you need to prepare the nail so it is smooth, free of

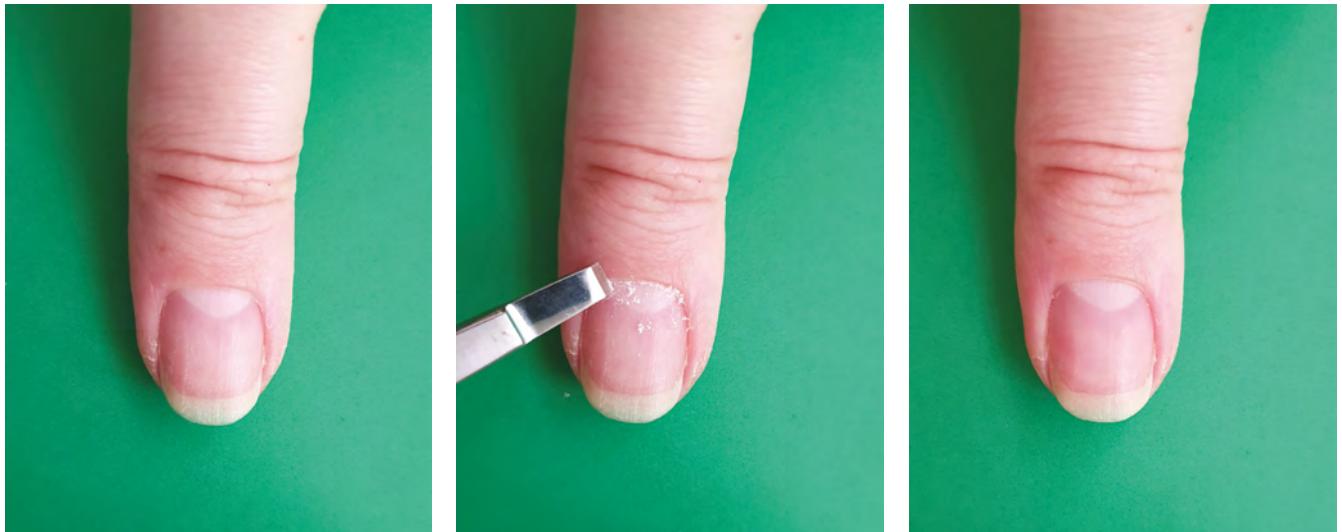
dirt and grease, and has a good surface for adhesion. Starting with clean hands, check for any broken or rough parts to the nail and smooth them with a file. You don't need to file the surface, just the edge. Where your skin joins the nail is a bit called the cuticle, and it is largely dead skin. Push along the

**By attaching something to your nails you can have it embedded and part of you at all times**

nail with the orange stick from the top of your nail towards your hand, and push off any dead skin you find. Do not push the stick under your skin.

## PRIMER

Now the cuticle is sorted out, it won't get under the edge of the acrylic and let it lift up. Nails have natural oil in them, so we need to dehydrate the nail. To do this, wipe with isopropyl alcohol. Try not to touch it with your fingertips from now on, as it could put



grease back on. We have got to the stage where our nail is sugar-soaped. To prep the surface, use a mix called 'acid-free primer'. You can pick this up at beauty stores for around £5. Paint a thin layer on, and you will see the nail turn matte as it dries. We have a prepared nail!

### CHOOSE YOUR ACRYLIC

The next stage is to apply a base layer of acrylic. This will be a thin layer so that we have a smooth surface to put the light or data tag on. If it turns out bumpy, we can file it.

There are two options here – acrylic liquid and powder, or acrylic gel. The process is similar in that both make a blob of plastic that's mouldable over the nail, but the liquid and powder sets in the air, while the gel sets under a UV lamp. The liquid and powder is cheaper than the gel, however, you only get

**Below** Painting primer on the nail – look for when it turns matte



seconds to shape the blob, whereas with the gel you get as long as you want.

### PRACTICE MAKES PERFECT!

If you have set A (liquid and powder), you will want to practice this a few times in a well-ventilated space. Put the liquid in the glass dappen dish, and open the pot of acrylic powder. On some waxed paper or a silicone baking sheet, first, dip the brush in the liquid and tap off any drips. Pull it gently across the acrylic

**Above** Before, during, and after removing the cuticle

**“** The liquid and powder sets in the air, while the gel sets under a UV lamp **”**

powder, and you will see a blob form. Push that blob down onto your paper or sheet, and dip the brush straight back in the liquid. Wipe the brush on some kitchen roll. Use the brush to mould the blob by gently tapping it until it sets. Get a feel for how quickly it sets and how long you have to work with it. Too much liquid will make it runny, and set too fragile, leading to acrylic lifting off the nail or cracking. Too much powder will make it crumbly, and set lumpy and breakable.

If you have set B (gel and UV lamp), you just need to put a dab of the gel on the nail, dip the brush in the slip liquid to stop it sticking to the gel, and push it into shape, before curing under the lamp for the time stated on the packaging.

### APPLY THE BASE LAYER

When you're ready, apply a small blob to your nail and tap it out to spread it. Once it gets close to your skin, →

### YOU'LL NEED

- ◆ A willing volunteer (or your own hands)
- ◆ Paper towel to protect your table
- ◆ Nail file
- ◆ Orange stick (a blunt stick with a flat surface)
- ◆ Lint-free wipes
- ◆ Isopropyl alcohol
- ◆ Acid-free primer
- ◆ Sable brush (not a plastic brush)
- ◆ Dappen dish or shot glass
- ◆ Silicon baking sheet for practice
- One of these two sets:
- ◆ A Acrylic powder and acrylic monomer liquid
- ◆ B Acrylic gel, gel slip liquid, UV lamp 36W or above

## TUTORIAL

### QUICK TIP

The first acrylic nails came out of the polymers for dentures – OPI, who make fancy nail varnish, used to be a dental supplies company.



**Above** Left – liquid and powder. Right – gel

**Below** Use your nail, or the other end of your brush, to smooth the tag down, to avoid putting oils from your fingers on it

**Below** Check for any flaws before varnishing

stop tapping, as it will spread a little more afterwards. As the acrylic sets, it becomes more and more matte looking. When it looks matte, tap the acrylic with one of your other nails. You will hear a clear tap if it's set, but a dull one if it's still setting. With the gel, you will need to cure it for the time it states on your brand, which is usually between 30 and 60 seconds.

### APPLY THE TAG

Now, apply the data tag (that's what I'm using, a MIFARE Ultralight NTAG213) or LED, and smooth it down. Rub over it with the back of your nail, making sure all edges are stuck down and that there are no air bubbles under it. Take your time over this.

Now repeat the acrylic layer step. You need to make sure all of the tag is covered, and that it is a

smooth arch shape across the nail. Don't worry if you now have an out of focus-looking tag inside there, you can make it clear as glass by putting a top layer on later. This layer should be thicker than the last one.

### FINISHING OFF

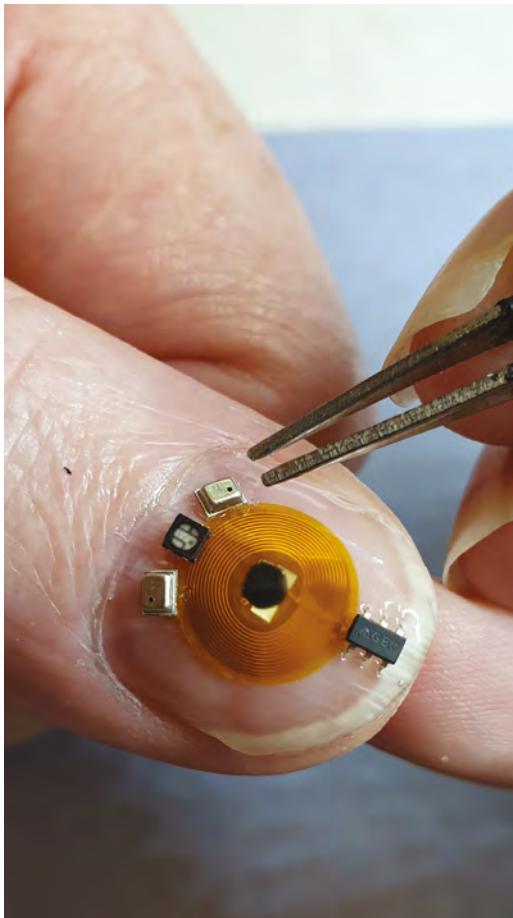
Finally, once you've smoothed out any bumps, apply a thin layer of clear nail varnish if you want to see the circuit inside. If not, decorate with nail varnish or glue-on gems. For the lights, it's important to leave a small gap in any varnish just around where the light itself is.

### HOW DOES ACRYLIC SET?

The liquid is ethyl methacrylate (EMA) and some inhibitors. The building blocks of a plastic polymer are monomers, which is the EMA. The inhibitors stop them from joining together in the bottle. The powder is a polymer with initiators in. The catalyst in the liquid triggers the initiator in the powder to combine with a monomer, which combines with another monomer, and so on until they form long chains of polymers, trapping the tiny polymer beads from the powder, creating a strong plastic structure.

In the UV gel, the monomers are in the gel, and the initiators are also in the gel, but they are inactive. When UV light hits them, the photoinitiators set off the chain reaction to make the monomers join into polymer chains. Note there are no polymer beads in this type, which makes it more flexible.





**Above** ◇  
Surface-mount components used for decoration

### EMBEDDING ACTUAL COMPONENTS

If you want, you can embed actual components for decoration. Before you add the top coat, use superglue to position the components on the surface of the nail. For strength, you can use another layer of acrylic over them, making sure to build it up between the components by pushing it into the gaps, or just varnish over them.

### REMOVAL

When you are ready to remove the acrylic nail, trim off the top edge as you would trim your nails. You may find that snips/hippers are better than nail clippers for this. Use a file to lightly file the top layer, to allow acetone to penetrate it. Put some acetone on a cotton wool pad and place it over the nail. You can hold it in place with some kitchen foil. After a few minutes, remove the foil and pad, then push the softened acrylic away from the finger, towards the top of the nail. If it hardens, repeat the process until you have removed all of the acrylic. Rub some oil (olive oil works fine) onto the nail to restore flexibility. □



### QUICK TIP

Acrylic nail materials can be used in other projects for fabrication of small items or repairs.

**Left** ◇  
Check for any flaws before varnishing

**Below** ◇  
Thanks to Sheffield Hardware Hackers and Makers at Sheffield Hackspace for being hand models. (@shhmakers)

“ Apply a thin layer of clear nail varnish if you want to see the circuit inside. **If not, decorate with nail varnish or glue-on gems** ”



# Hacking servo motors for more rotation

How to remove the control circuitry and mechanical stop on a standard servo motor



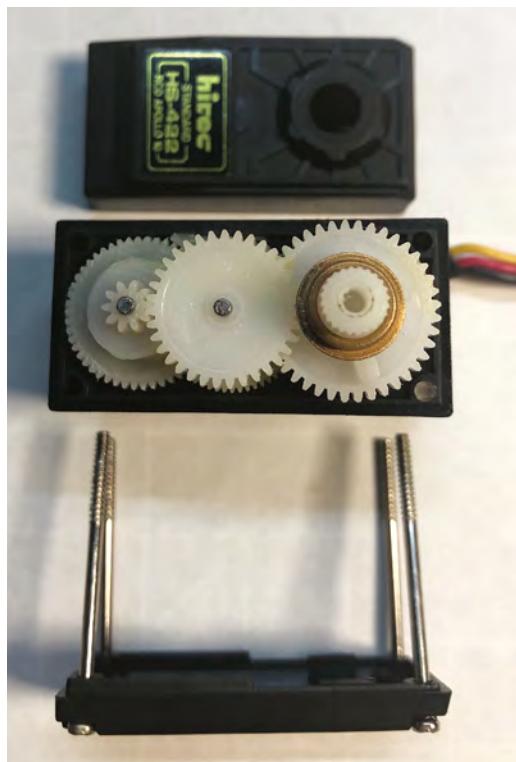
**Gareth Branwyn**

@garethb2

Gareth has been a lifelong practitioner (and chronicler) of DIY tech, media, and culture. He is the author of ten books, including *Make: Tips and Tales from the Workshop*, and is a former editor for Boing Boing, Wired, and Make:.

**S**mall hobby servo motors are commonly found in radio-controlled planes, in miniature robots, and in many other types of projects that require precisely controlled movement, slow motion, and decent

**torque.** Servos are relatively cheap and come in all sorts of sizes, and usually come with a set of 'link' or 'control horns', nylon plastic arms and disks that attach to the motor shaft, and have tie holes that allow you to attach control rods, wheels, and other objects. It can position these fairly accurately with a 180-degree arc using pulse width modulation (PWM) or pulse code modulation (PCM).



**Right ♦**

A standard hobby servo taken apart to show the top plate, the main body (with gear-box), and the bottom plate with four screws

If you don't need this control, but do like the form factor and need the ability to spin more than 180 degrees, you can convert your servo into a geared motor. Let's find out how.

To transform our servo into a regular DC motor with gearbox, all you have to do is complete the following two simple 'hacks'.

## REMOVING THE FINAL GEAR TAB

The large 'final gear' in the servo's gearbox has a mechanical stop on it that prevents the motor from travelling outside of its fixed 180-degree arc. To remove this stop to allow continuous rotation, do the following:

### Step 1: Remove the bottom screws and top plate

There are commonly three parts to a servo motor's plastic case: the top plate, the main housing, and the bottom plate. All of these are usually held together by four long Phillips screws located on the bottom plate. Use a suitable driver to carefully remove these screws. With the screws removed, carefully pull off the top plate. You should now see the three main gear assemblies.

### Step 2: Remove the middle gear

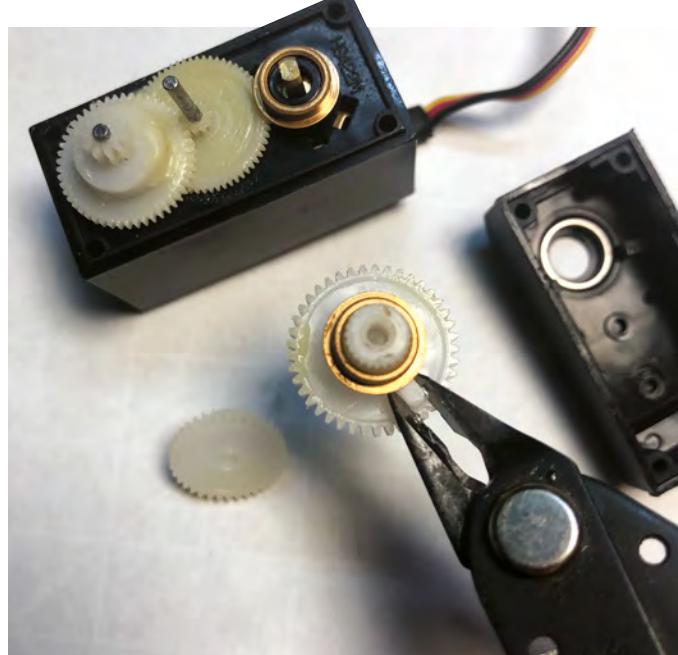
Carefully remove the centre gear from the assembly and set it aside. It will go back where it came from.

### Step 3: Remove the final gear

The final gear will have a small, clearly visible tab on it. Remove this gear.

### Step 4: Remove the mechanical stop

Using a pair of angle cutters, razor knife, or another suitable tool, remove the little tab from the final gear. File or sand it down so that nothing will interfere with its full rotation when you put it back. Note: If you have metal gears in your gearbox, the stop will usually be a small pin. You can use the tip of your side cutters to gently pry the pin from the gear.



## QUICK TIP

Because of the popularity of servo motor hacking to create continuous rotation motors, many servo motor makers now sell versions of their servos without the electronics and the mechanical stop. These motors are usually marked 'Continuous Rotation' on the motor casing.

### Left ♦

The PCB and DC motor of the servo. All three wires get desoldered from the PCB, and the red and black wires get soldered directly to the terminal on the motor

### Above ♦

The middle and final gears removed and the mechanical stop on the final gear about to be snipped off

### Step 5: Replace the final and middle gears

Replace the now-altered final gear, and then carefully reseat the middle gear. Make sure that all of the gears mesh as they are supposed to.

### Step 2: Desolder the power wires

Locate the solder points on the PCB for the red and black power wires and desolder those. Also, desolder the two short power wires from the PCB that are currently attached to the motor terminals.

**These are usually held together by four long Phillips screws**

## REMOVING THE CONTROL ELECTRONICS

With the gears sorted out, it's time to remove the circuit board motor controller, desolder the wires on the PCB, and solder the power wires directly to the terminals on the motor. Heat up your soldering iron and do the following:

### Step 1: Identify the wires and desolder the control wire

There will be three wires. One, the control wire, likely yellow (or white or orange), is for controlling the electrical pulses that create the precision movements of your servo. We won't be connecting that to anything. The other two, likely red and black, are the power wires. Go ahead and, using your iron and a solder sucker or solder wick, heat the solder point on the control wire (yellow) and remove this wire from the PCB.

### Step 3: Remove the PCB

With all three connections desoldered, you can now remove the PCB from the servo housing. You won't be needing it here.

### Step 4: Solder the power wires to the motor

Now, go ahead and solder the red and black wires directly to the terminals on the DC motor. On your motor, the polarity of the motor will likely be indicated by a mark or colour letting you know which terminal is negative or positive. Here, our positive terminal is marked by a red dot.

### Step 5: Put everything back together

With the PCB removed and the power wires connected directly to the motor, it's time to put everything back together. Snap the top plate back on, the bottom plate, and screw the assembly back together with the four screws.

That's it. You have transformed a servo motor into a continuous rotation gear motor that you can use with an off-board motor controller/MCU in a robot drive train or similar application. □

## YOU'LL NEED

- ♦ Standard hobby servo motor
- ♦ Small Phillips screwdriver
- ♦ Angle cutters
- ♦ Small file
- ♦ Soldering iron and solder
- ♦ Solder sucker or solder wick



# Motorised wing: making your cosplay move



Sophy Wong

@sophywong

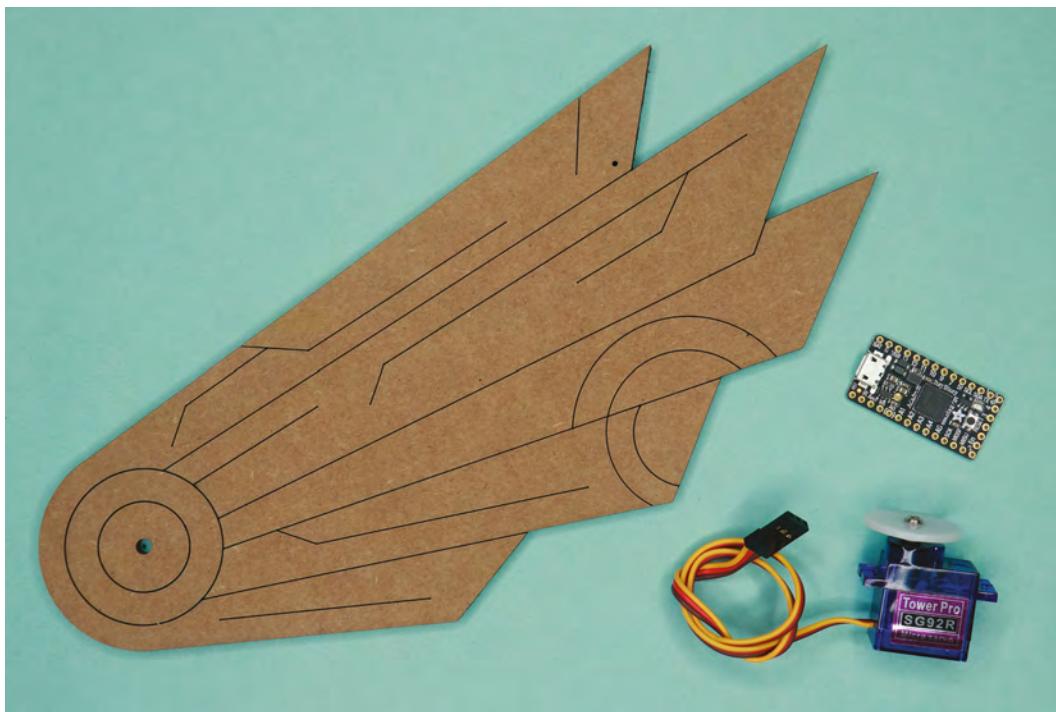
Sophy Wong is a designer, maker, and avid creator. Her projects range from period costumes to Arduino-driven wearable tech. She can be found on her YouTube channel and at [sophywong.com](http://sophywong.com)

Add a servo for breathtaking effects

**A**dd movement to your cosplay with a servo motor! In this intermediate build, we'll make an articulating wing that opens and closes at the touch of a finger. This project is a perfect add-on to a sci-fi robot costume, a steampunk aviator, or any costume with a shoulder pauldron. We'll bring our wing to life with some crafting, soldering, and a little bit of code!

We've kept this build simple with everyday materials like cardboard and string. Clean, used shipping boxes are a great source of 4 mm corrugated

cardboard. A laser cutter makes quick work of cutting out the wings, but if you don't have access to one, use scissors or a craft knife to cut the cardboard by hand. Our cyberpunk-style wing template includes design lines that can be engraved or traced on the surface of the wing for more texture. Download our template at [hsmag.cc/issue23](http://hsmag.cc/issue23), or create your own custom wing shape from scratch. Just make sure the pivoting side of the wing is a nice, round circle, for smooth movement. Once cut, paint the wings to match your costume – a coat of silver spray paint made our cardboard shine bright.



**Left ♦**  
The starting point  
for our wing

## YOU'LL NEED

- ♦ **ItsyBitsy M0 Express**  
[adafruit.com/  
product/3727](https://adafruit.com/product/3727)
- ♦ **Micro servo and servo horn**  
[adafruit.com/  
product/169](https://adafruit.com/product/169)
- ♦ **LiPoly backpack for Trinket and ItsyBitsy**  
[adafruit.com/  
product/2124](https://adafruit.com/product/2124)
- ♦ **LilyPad sewable on/off switch**  
[sparkfun.com/  
products/9350](https://sparkfun.com/products/9350)
- ♦ **LiPo battery 3.7 V 1200 mAh**  
[adafruit.com/  
product/258](https://adafruit.com/product/258)
- ♦ **Elastic shoulder mount for GoPro**
- ♦ **Wire and soldering supplies**
- ♦ **Copper tape**
- ♦ **Solderless breadboard for prototyping**
- ♦ **Cardboard**
- ♦ **Spray paint, acrylic paint, or markers**
- ♦ **Bamboo skewer**
- ♦ **Plastic bottle cap**
- ♦ **Small plastic washers**
- ♦ **Hot glue gun and glue sticks**

The ends of the wings are sewn together with thread, so that when the front panel is lifted by the servo, the wing spreads open like a fan. You'll want to use strong thread for this, something that can withstand the stress of being tugged on by the servo. Heavy-duty thread should suffice, and still be thin enough to slide between the panels. If you've already got a costume to mount your wing to, great! If not, a GoPro shoulder mount makes a great premade harness. You author purchased the one shown here for about \$10 on Amazon. Make sure the wing is held tight to your shoulder so that it doesn't slide around when actuating. If you have long hair, keep it far away from the servo.

If you've never worked with motors before, the servo in this project is a great place to start! A servo is a motor that is controlled with position data – you tell it where to point and it moves there. The servo in this project can rotate up to about 180 degrees, but we will only use about 60 degrees of movement. Servos are extremely useful and fun to play with, but they can be damaged if mishandled. Do not turn the servo by hand or force its movement, and avoid sending it beyond its limit in your code.

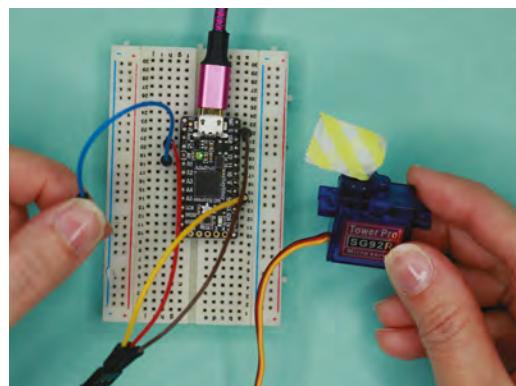
We'll control the servo with a small microcontroller: an ItsyBitsy M0 from Adafruit. This board has a special 5V output pin that is perfect for our servo, and can be programmed in beginner-friendly CircuitPython. To make our project mobile and easily rechargeable, we'll add a LiPoly backpack to the ItsyBitsy. Our power source is a LiPo battery, which must be handled with care for safety. If you're unfamiliar or uncomfortable

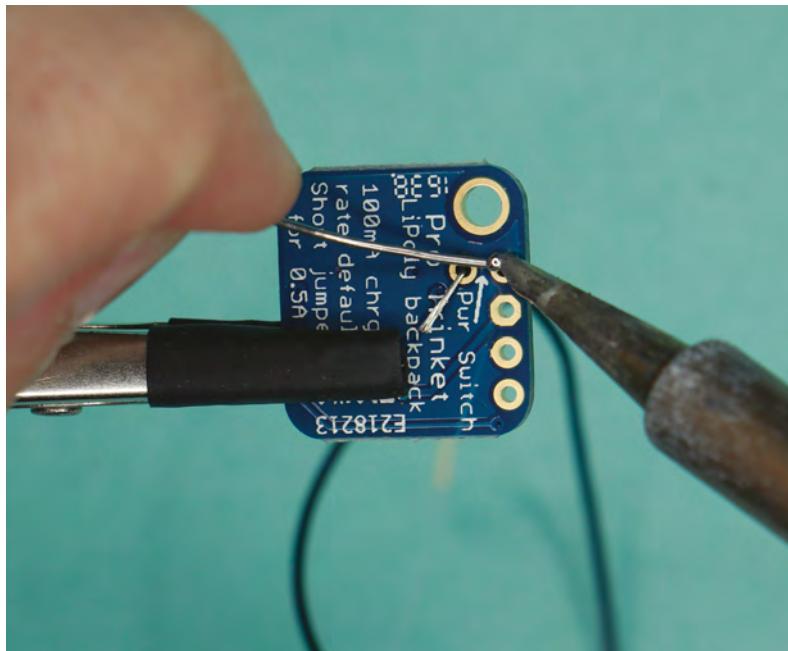
with LiPo batteries, this project can also be powered by a USB power bank via its micro USB port.

## PROTOTYPING YOUR CIRCUIT

The first thing you'll want to do is make sure your servo works properly, so start by gathering your materials and prototyping your circuit. If you think you'll be using an ItsyBitsy M0 in projects often, consider investing in a second one and soldering the included headers into the pins. Keep this breadboard-ready ItsyBitsy on hand for prototyping and experimenting with circuits. (Another option is to build this entire circuit on a solderless breadboard and install the whole breadboard in your project, if you have room for it.) →

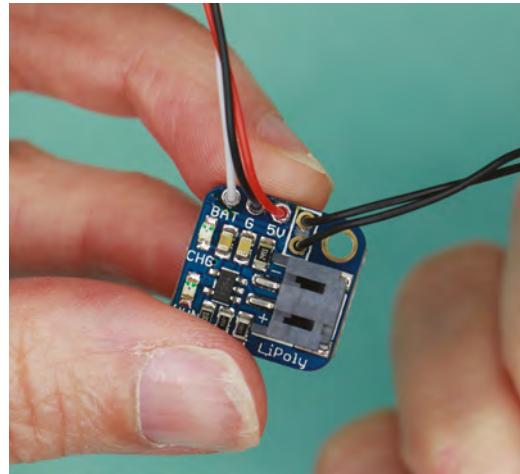
**Figure 1 ♦**  
The basic circuit





**Above** A cuttable trace enables a switch

**Above Right** The LiPo battery connector powers our project



Follow **Figure 1** (previous page) to mock up the main portion of your circuit. At this point you don't need to connect the switch or the LiPoly backpack, just connect the servo to the ItsyBitsy M0 on pin 5, and an open wire on pin A1. That should be enough to program the ItsyBitsy and test the servo.

### PROGRAM WITH CIRCUITPYTHON

Connect the ItsyBitsy M0 to your computer with a micro USB cable. The board will appear as a drive named CIRCUITPY. Head to [hsmag.cc/KNXgNL](http://hsmag.cc/KNXgNL) and follow the instructions to update your board with the latest stable release of CircuitPython.

While you're there, grab the latest library bundle as well from [hsmag.cc/HZqwVA](http://hsmag.cc/HZqwVA). For this code, we'll only need the 'adafruit\_motor' library. Copy it from the library bundle into the **lib** folder on the CIRCUITPY drive.

CircuitPython can be written and edited in any text editor, but we highly recommend using a code editor like Mu Editor. To learn more about Mu, and for help getting started with CircuitPython, visit [hsmag.cc/jJfvcN](http://hsmag.cc/jJfvcN).

Once your board is updated, and the motor library is installed, you're ready to code! Open your editor and create a new **code.py** file on your ItsyBitsy. (If your ItsyBitsy already has a **code.py** file, delete it and start afresh.)

As always, our code starts with importing the libraries we'll be using. **pulseio** allows us to use PWM (pulse-width modulation) to control our servo, and **touchio** gives us access to the capacitive touch capabilities of the board. The 'adafruit\_motor' library is a big one, and we'll only need the servo portion of it.

```
import time
import board
import pulseio
import touchio
from adafruit_motor import servo
```

Next, we'll set up pin A1 as a capacitive touch input named **buttonTouch**. This will act as a 'button' to control our wing. We'll also make pin D5 a PWM output, and set it up as a servo named **my\_servo**. Finally, we'll create a variable called **wingUp**. We'll use this variable to track the state of our wing in the code. The value of this variable will be **True** when the wing is up, and **False** when the wing is down. We'll assume that the wing is down when the board is turned on, so we'll set the initial value of **wingUp** to **False**. We'll also create a variable named **upAngle** for the open position of the servo. You can tune this value as needed to make your wing open fully.

```
# Create a touch capacitive input on Pin A1
buttonTouch = touchio.TouchIn(board.A1)

# Create a PWMOut object on Pin D5
pwm = pulseio.PWMOut(board.D5, duty_cycle=2 ** 15, frequency=50)

# Create a servo object, my_servo
my_servo = servo.Servo(pwm)

# Track the state of our wing, false when down, true when up
wingUp = False

# Position of servo when wing is fully open. Not to exceed max angle of servo.
upAngle = 60
```

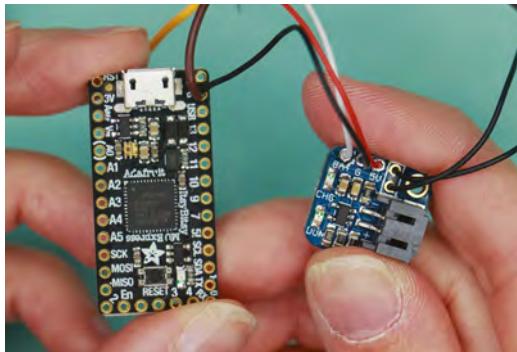
The main loop of our code is a simple conditional statement (`if, elif`). The `touchio` library makes it easy to use capacitive touch inputs to trigger events. If the capacitive input is touched and the wing is down, the servo will move to 60 degrees and the value of `wingUp` will be set to `True`. However, if the capacitive input is touched and the wing is up, the servo will move back to 0, and the value of `wingUp` will be set to `False`. A short `time.sleep` in each condition helps us avoid multiple triggers per touch.

```
while True:
    if buttonTouch.value and wingUp is not True:
        my_servo.angle = upAngle
        wingUp = True
        print("wingUp = True")
        time.sleep (0.5)
    elif buttonTouch.value and wingUp is True:
        my_servo.angle = 0
        wingUp = False
        print("wingUp = False")
        time.sleep (0.5)
```

That's it for our code! Saving this code in the `code.py` text file will automatically restart the board and run the program. Touching the wire connected to A1 should make the motor turn 60 degrees and stop. Touching the wire again should make it turn back to zero and stop. If you haven't connected a servo horn to the spindle yet, a piece of tape can help track the servo's movement while testing.

Once assembled, you may need to tune the `upAngle` value to make your wing open completely. Increase the value in small increments, and be careful as you approach the maximum angle of your servo (usually about 180 degrees): pushing the servo too far can damage it.

**Below** The ItsyBitsy lives up to its name



## BUILD YOUR CIRCUIT (FOR REAL THIS TIME)

Now that you're sure your servo is working properly, it's time to solder the circuit together for good. Start by adding an on/off switch to the LiPoly backpack. Use a craft knife to cut the exposed trace between the two switch pins. Then solder one short wire to each pin of the switch.

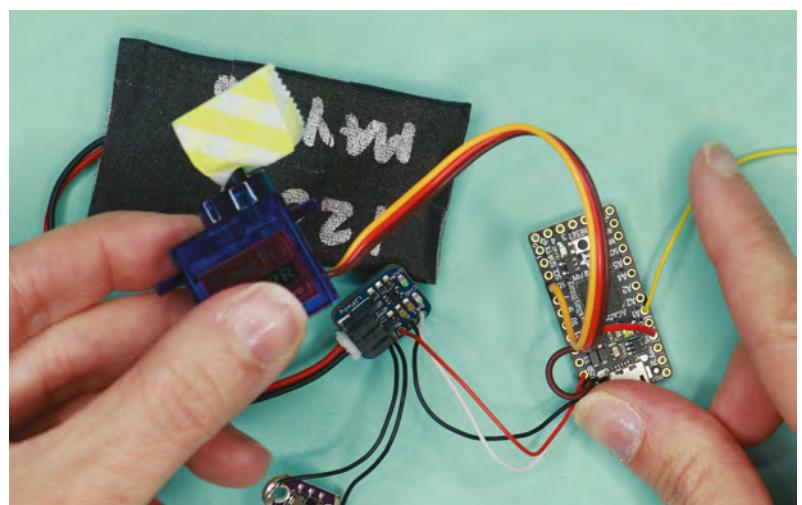
Solder each wire to one of the pins on the on/off switch. We used a LilyPad sewable on/off switch that we love for both sewn and soldered circuits – the flat bottom makes it easy to secure to wearable projects.

**“ Increase the value in small increments, and be careful as you approach the maximum angle of your servo (usually about 180 degrees)**

Next, solder three wires into the remaining pins on the LiPoly backpack: BAT, G, and 5V. The backpack is ready to be connected to the ItsyBitsy, but don't solder it quite yet. The ItsyBitsy M0 only has one ground pin, and both the LiPoly backpack and the servo need to connect to it. The pins on the ItsyBitsy are small, so we'll need to solder both wires at once.

Snip the connector off the servo wires and pull the three individual wires apart for about five or six centimetres. Insert the ground wire from the servo and the ground wire from the LiPoly backpack into the ground pin on the ItsyBitsy M0 and solder them in place. Then, finish soldering the servo by connecting its power wire to the Vhi pin, and its data wire to the 5! pin on the ItsyBitsy M0. Finish soldering the LiPoly backpack by connecting the 5V →

**Figure 2** The final circuit



## TUTORIAL

**Below** ♦  
You can follow  
our designs or  
get creative with  
your own



**Below** ♦  
Hot glue makes  
everything better



wire to the USB pin, and the BAT wire to the BAT pin on the ItsyBitsy M0.

Finally, cut another small piece of wire, about 5–6 cm long, and solder it to pin A1 on the ItsyBitsy M0. This wire will connect to a piece of copper tape on the shoulder pauldron for our capacitive touch input. The final circuit is shown in **Figure 2** (previous page).

### ASSEMBLE THE WING

Using our template, or your own design, cut three identical wing panels out of cardboard. We used a laser cutter to cut our panels, but the shapes are designed to be simple enough to cut by hand. The straight edges should be easy to cut (carefully!) with a craft knife, and the round side may be easier to cut with heavy-duty scissors. You'll want the round side to be a perfect circle for smooth pivoting action when the wing opens. Make the holes with a heavy-duty hole punch, or simply use the bamboo skewer to poke through the cardboard.

Once the panels are cut out, paint and decorate them to match your costume. Trace or score the design lines from our template for a futuristic look, or draw your own surface design. The surface of the wings must be flat so that the panels can slide past each other, so stick to paint or markers for your design.

To previsualise what you are building, stack the wing panels on top of each other, aligning the holes on the left. Fan the panels out to find the fully open position of your wing: the bottom-right edge of each panel should just cover the thread hole of the wing under it. (See our finished project for reference.) Now you're ready to sew the wing together with thread.

For this part of the build, it's easiest to work with the panels facing down on your work surface. Thread

a needle with about 40 cm of heavy-duty thread. Starting with the front panel, loop the thread through the thread hole in the wing-tip and tie a knot on the back of the wing. Then, pass the thread over the front of the wing and down to the next panel. Loop the thread through the thread hole on the second wing, align the panel in its open position, and tie a knot on the back of the panel.

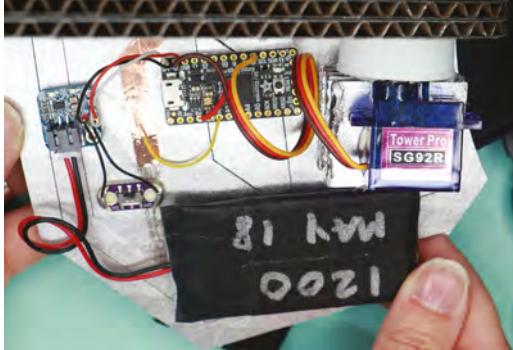
Repeat this process to add the third panel, tie the final knot, and snip the thread tail short. Slide the bamboo skewer into the pivot hole to check your work – the panels should open and close easily like a fan. The thread will be taut when the wing is open, and loose (but hidden between the wings) when closed.

**“** We used a laser cutter to cut our panels, but the shapes are designed to be simple enough to cut by hand **”**

Remove the skewer and cut it in half to make it easier to work with. Slide the front panel onto the skewer so that the skewer protrudes a few centimetres through the front of the panel. Add a small plastic washer to the skewer on each side of the panel, and attach each washer with a dab of hot glue. This front panel needs to be solidly connected to the skewer, so take your time with this and make sure the glue is completely dried before moving on.

Slide the remaining two panels onto the skewer, adding a plastic washer onto the skewer between them. Do not glue the washer to these panels – they should rotate freely around the skewer. Check your work by resting the bottom edge of the panels on your work surface and rotating the skewer. The front panel should rotate upward, fanning the other panels out behind it. On the front of the wing, snip the skewer flush with the glued washer. On the back, cut it down to about 1 cm.

It's possible to purchase couplers for attaching a shaft to a servo, but in the spirit of this build, we are winging it with everyday materials and the standard servo horn that came with our servo. A plastic cap from a water bottle will be our coupler! Check to make sure that the servo horn you'll be using will fit inside the cap; if it's too long (like ours was), carefully snip the tips off the ends. Screw the horn into place onto the servo.



Drill a hole in the centre of the plastic cap just big enough for the skewer to go through, and slide the cap onto the skewer, with its flat side against the back of the last wing panel. Add a plastic washer and secure with a glob of hot glue.

Glue the servo horn into the cap with a generous layer of hot glue. You may need a lot of hot glue for this step, and there are a couple of things to watch out for. Use enough hot glue to securely attach the servo horn, but do not let the glue rise past the horn into the spindle of the servo or its movement will be inhibited. It can also take quite a while for a thick layer of hot glue to dry, and you will need to keep the motor upright in place until the glue is solid. Do not move the wing or actuate the motor until the glue is completely cool and dry.

### MOUNT EVERYTHING ON THE PAULDRON

While the glue is cooling, we can work on the shoulder pauldrone, which is the base of the wing. Use our shoulder pauldrone template to cut the cardboard out as before. To curve the pauldrone over your shoulder, roll it with a rolling pin while gently forming it into a curve. Paint the cardboard to match your wing. Remove the plastic GoPro mount from the GoPro shoulder harness, and attach the harness to the underside of the pauldrone with hot glue.

The partial circle in our design makes a great area for the capacitive touch ‘button’. Cut copper tape to cover the circle, and press into place. Use more copper tape to run a trace to the back of the pauldrone, where our electronics will be located behind the wing.

To mount the wing on the pauldrone, run a bead of hot glue along the bottom edge of the back wing panel only, and press it in place on the pauldrone. The two front wings must not be connected to the pauldrone, only the back wing. Add another bead of hot glue to the back of the join for good measure. The servo must be held down securely, so that its movement is transferred to the wing. Make a



platform out of cardboard layers and hot glue to support the servo. Hot-glue the platform in place under the servo, then glue the servo securely to the platform.

Use foam tape to secure the ItsyBitsy M0, the LiPoly backpack, and the on/off switch to the pauldrone behind the wing. Make sure none of the electronic components touch the copper tape trace. Shorten the trace if you need more room for your components. Strip the end of the wire from A1 and connect it to the copper tape trace with another small piece of copper tape.

To secure the battery, use adhesive Velcro so that it can be removed and replaced if needed. It’s a good idea to wrap the battery in gaffer tape for extra protection.

### SPREAD YOUR WINGS!

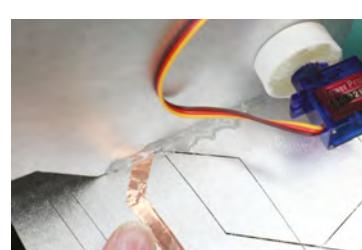
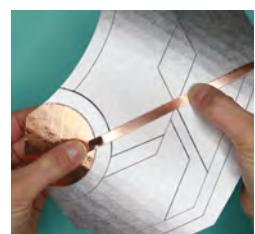
Your build is complete! Plug the battery into the LiPoly backpack and turn on the switch. Touch the copper circle to make your wing open – touch the circle again to make it close. Very Valkyrie! For a sleek look, you can hide the electronics with a decorative panel, but make it removable for repairing and recharging. With the LiPoly backpack installed, the battery can be recharged in place, via the micro USB port on the ItsyBitsy M0.

Of course, with all those leftover I/O pins on the ItsyBitsy M0, you can add even more functionality to your costume. Lights? Sounds? What will you add? Show us your electronic cosplay at [hackspace@raspberrypi.org](mailto:hackspace@raspberrypi.org) □

**Above Left** Wired up and ready for flight

**Above** Flight goggles are optional

**Below** Copper tape is a great material for conductive wearables



# Send data like it's 1970

An old protocol that's still got a lot to offer



**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**ost of the time you want a microcontroller to be an independent device operating on its own. Give it power, and let it do its processing. However, sometimes we need to connect it to something

larger – a general-purpose computer. Perhaps you need the processing power of a PC, or perhaps you want to send data to the internet from a microcontroller without network access.

Fortunately, there's one protocol built into almost every operating system and microcontroller: serial (see box below). Back in the day, when trains ran on steam and all this were fields, computers had a D-shaped serial port that used the Universal Asynchronous Receive and Transmit (UART) protocol, but that has long since gone and now we have only USB. While this is a different protocol, thanks to a bit of software, and USB to UART bridges in modern microcontroller boards, we can usually ignore the fact that they are different and send data out of a UART, via a USB connection, and receive it on a computer.

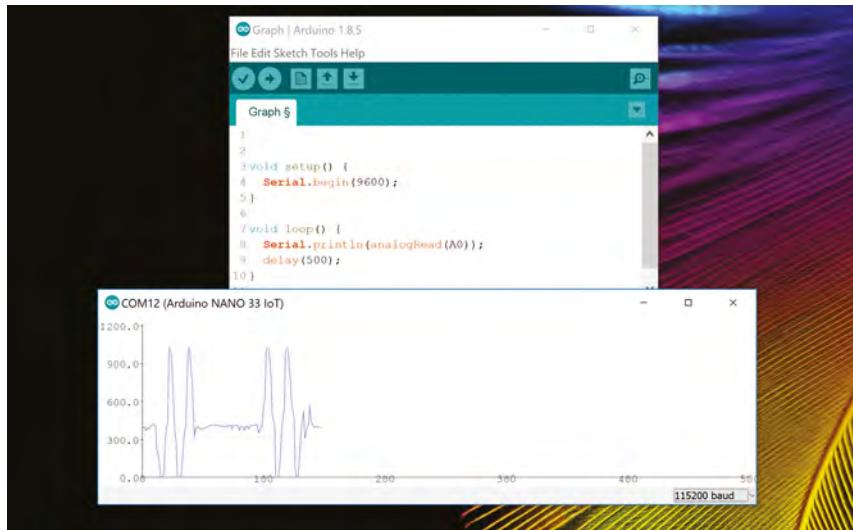
Let's look at a very simple example (you should find this in Examples > Communication > Graph in the Arduino IDE).

```
void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(analogRead(A0));
    delay(2);
}
```

As you can see, we have to first initialise the speed of the UART port. You have to set this on both the sending and receiving end (which we will look at next). In general, lower speeds are likely to be more stable than higher speeds. 9600 is about the lowest we ever go, and we usually have no problem going up to 115,000. Once it's created, we can simply push data out of the UART. In this case, it's sending the data from A0. If this is unconnected and floating, you could get a range of values, but you could hook this up to a sensor for more meaningful numbers.

If you upload this code to your board, you can then view the data in the serial monitor (Tools > Serial Monitor). Make sure that the speed is set to 9600 and it should come through. The Arduino IDE also



## Left ◇

The plotter lets you view values coming from sensors in real time

## SERIAL

Technically, the word 'serial' refers to any protocol where data is sent bit after bit on a single wire. There could be one or more support wires, but as long as the data is sent on one wire, it's serial. The alternative is parallel, where data bits are sent down multiple wires simultaneously.

However, in common usage, serial refers to Universal Asynchronous Receive and Transmit (UART), a protocol that's been used since the 1970s to transfer data between devices.

## NOT JUST COMPUTERS

Here, we've looked at how to connect a computer to a microcontroller and communicate via USB serial, but there are loads of other devices that you can link your microcontroller to via serial. One popular option is Bluetooth serial links. These are hardware devices that expose a Bluetooth connection as a serial port. Any data you send down the wired serial link between the microcontroller and Bluetooth module will be sent over the Bluetooth link to whatever the module is paired with. You can use this to send data back and forth between your microcontroller and mobile phone (there are several Bluetooth serial apps available), or other Bluetooth-enabled devices.

has a Serial Plotter that converts numbers sent over a serial connection into a graph.

The `Serial.println()` function can take a few different data types, including floats (as shown here) and strings.

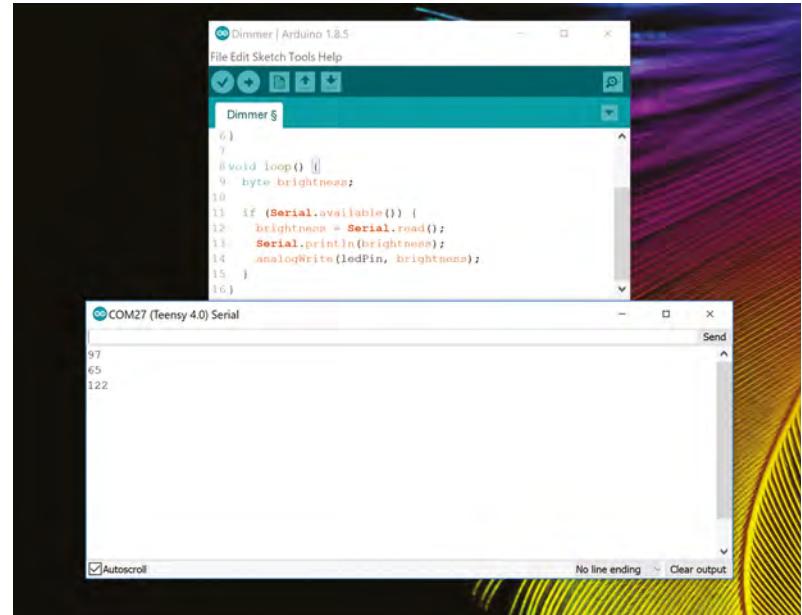
This is a great way of seeing what's going on in your scripts, but it can also be used to send data to programs running on your PC. Most programming languages have a library or module for working with serial connections, so you can send data directly from your microcontroller to software running on your PC.

## BACK AND FORTH

So far we've seen data sent from the microcontroller to the computer, but the serial protocol is bidirectional (well, really, it's two unidirectional channels pointing in opposite directions, but let's not worry about that), which means we can also send data from our computer to the microcontroller. The following code (which you'll find in your Arduino IDE in Examples > Communication > Dimmer) takes data in and uses it to control the brightness of an LED.

```
const int ledPin = 13;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop() {
```



```
byte brightness;

if (Serial.available()) {
  brightness = Serial.read();
  Serial.println(brightness);
  analogWrite(ledPin, brightness);
}
```

**Above** Serial can send data in both directions

While `Serial.println()` can take a wide variety of formats, `Serial.read()` will always return a single one-byte character. In this case, we're using this to control the brightness of the LED. However, by default, the serial monitor will also send characters for line end and carriage return, so we need to turn these off. Change 'Both NL and CR' to 'No Line Endings'. Now, when you enter a character into the box and hit Send, the LED will be adjusted based on the ASCII value for that character. With letters, 'A' is the lowest and 'Z' is the highest brightness.

Because of the one byte limitation, it's easiest if you can limit data sent from your computer to your microcontroller to single bytes. Commands can be single letters, etc. However, if this isn't possible, you can build up data by reading multiple bytes (i.e. calling `Serial.read()` several times) and stitching the data together in the way you need.

Using serial, you can quickly and easily create custom USB gadgets for your computer, or send data back and forward to control your makes. □

DON'T MISS THE BRAND NEW ISSUE!

YOUR OFFICIAL RASPBERRY PI MAGAZINE

# The MagPi

Issue 86 | October 2019 | magpi.cc

The official Raspberry Pi magazine

10 RETRO GAMING KITS  
The best arcade controllers, kits and consoles

POWERFUL RASPBERRY PI PROJECTS

25 rewarding builds for masterful makers

SUBSCRIBE FROM JUST £5

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



**FREE PI ZERO W STARTER KIT\***

With your 12-month subscription to the print magazine  
[magpi.cc/12months](http://magpi.cc/12months)

\* While stocks last

**Buy online: [store.rpipress.cc](http://store.rpipress.cc)**

# FIELD TEST

## HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG  
**112**

### DIRECT FROM SHENZHEN: **ESP32 CAM**

A cheap, hackable webcam  
that you can program yourself

PG  
**122**

### CAN I HACK **HOT WIRES**

Electronics meets board-games

PG  
**116**

### BEST OF **BREED**



Raspberry Pi power supplies: Batteries,  
UPSes, and Power-over-Ethernet

## REVIEWS

124

### Teensy 4.0



The fastest microcontroller?

128

### Programino

An alternative way to code

126

### MONSTER M4SK



Halloween the easy way

129

### Power Carving

Spin your way to wood art

DIRECT FROM  
**SHENZHEN**

# ESP32-CAM

A programmable WiFi-enabled camera for under £7

By Ben Everard

 @ben\_everard

**Below** There's no mount for the camera, but the TF card port is conveniently positioned



A

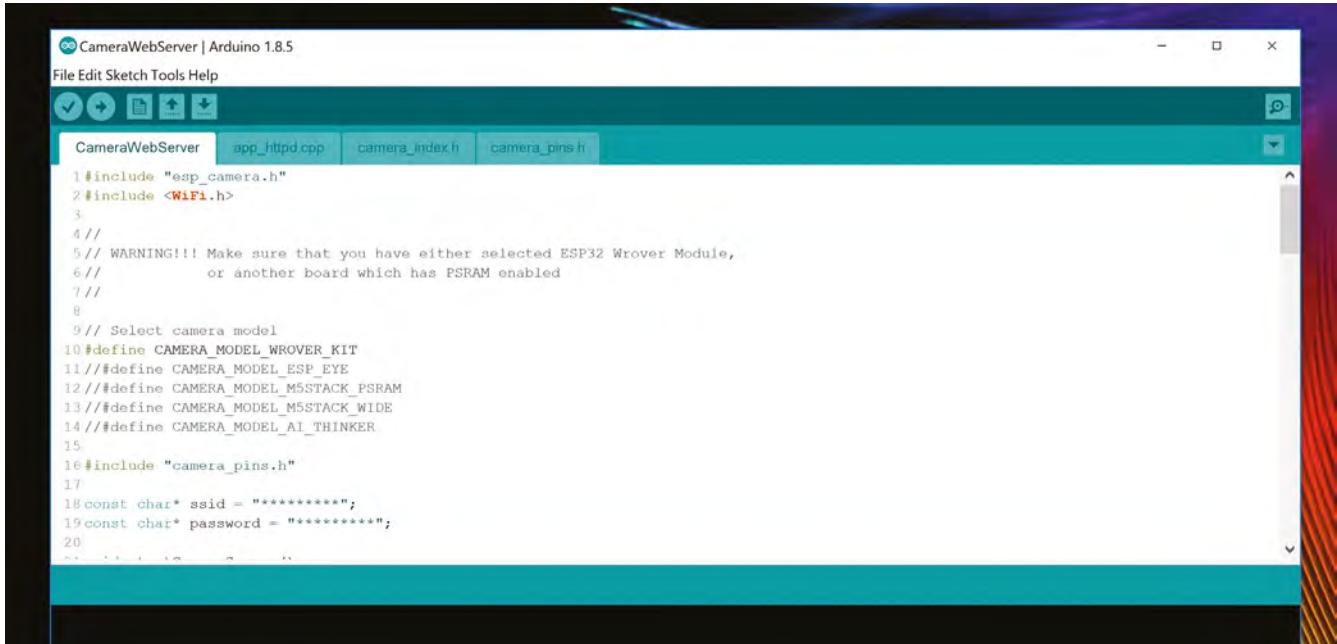
**cheap, hackable, WiFi-enabled camera is a holy grail for makers.** When we first heard about the ESP32-CAM from AI-Thinker, we had to find out if we had such a product for our maker toolbox. We ordered one for £6.66 (including delivery to the UK) from YX Electronic Components on AliExpress.

It came in two parts. The main board (which is an ESP32 module mounted on a larger board) and a separate camera module. The flex cable from the camera slots into the main board. Although there's no specific mounting for the camera, it sits nicely on top of the memory card slot, so with a drop of glue or tape, you can secure it easily.

The board is based on an ESP32 module, which is a programmable microcontroller with built-in WiFi and Bluetooth, with an additional 4MB of external RAM. Also on the board is a memory card slot and a camera connector that can take an OV2640 or OV7670 camera module (the former was included in the price of our board). In principle, this is a pretty beefy unit. The ESP32 has plenty of processing power, with two 120MHz 32-bit processing cores.

The memory card slot is listed as 'TF' (which stands for TransFlash) on every spec sheet we've seen. This is a little odd as this is the original name for what has now become microSD. We're not sure why they're listed using this name and not the current microSD, but you can use microSD cards in this port (though you will have to add support to them in your software).

Let's take a look at what it takes to get the board up and running. There are a few ways of programming the ESP32-CAM, but we're most familiar with the Arduino IDE, so we tried this. To install the ESP32 boards, go to File > Preferences and add



'[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)' to the Additional Board Manager URLs box – this is a comma-separated list, so add a comma before it if you already have any URLs in the box. In the Tools > Boards section, you should now have an ESP32 Arduino section. In that section, you should see an entry for AI Thinker ESP32-CAM. However, we weren't able to program the board using this. Instead, we used the definition for another, similar board: the ESP32 Wrover Module.

Once you've selected this, you should find the web server example sketch, which is a great test to make sure everything's connected properly. This is in Files > Examples > ESP32 > Camera > CameraWebServer.

You'll need to make a few changes to this sketch. First, add '//' to the start of the following line to comment it out.

```
#define CAMERA_MODEL_WROVER_KIT
```

Then delete the // at the start of this line to enable it.

```
##define CAMERA_MODEL_AI_THINKER
```

Finally, you'll need to enter your WiFi details in the following lines:

```
const char* ssid = "";
const char* password = "";
```

You'll also need to set the Partition Scheme to Huge App and the Upload Speed to 115200 (these are in the Tools menu). That's the software side of things

set up for this board, so let's now take a look at the hardware.

You may notice that there's no USB port on this board, so you'll need a USB to serial adapter to program it.

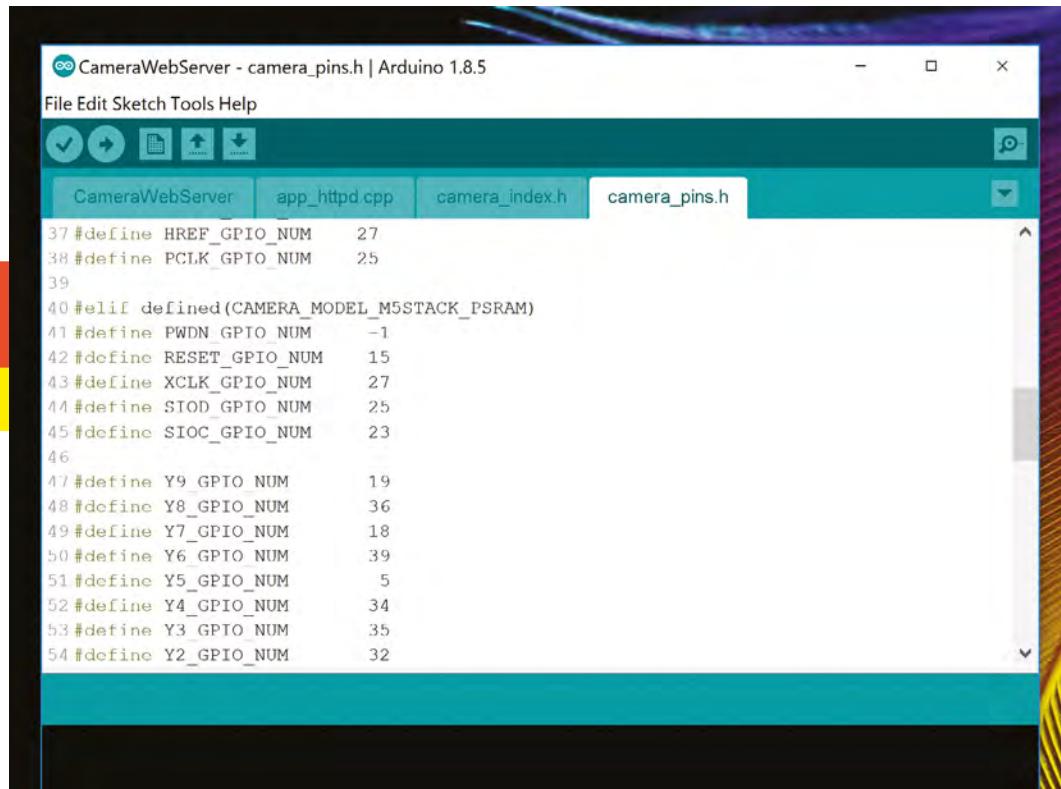
Any USB to UART adapter should work, and they are available for a few pounds. We used a Particle Debugger because we happened to have one with us, but this is overkill for this project.

You'll need four connections between the UART adapter and the ESP32 Camera: 5V (this may be labelled VUSB) to 5V, GND to GND, TX on the USB adapter to U0R on the ESP32-CAM, and RX on the adapter to U0T on the ESP32-CAM. Also, you'll need to connect IO0 on the ESP32-CAM and GND – this puts the board into flashing mode, and we'll remove this once we've programmed the board.

With that set up, you can plug your USB to UART adapter into your computer's USB port, and now you should be able to select a Port from the Tools menu in the Arduino IDE. Press the upload button (the ➔

**Above** The Camera example works out of the box, but isn't easy to read for beginner programmers

## REGULAR



The screenshot shows the Arduino IDE interface with the title bar "CameraWebServer - camera\_pins.h | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, download, and other functions. The tabs at the top show CameraWebServer, app\_arduino.cpp, camera\_index.h, and camera\_pins.h, where camera\_pins.h is currently selected. The code editor displays the following C++ code:

```
37 #define HREF_GPIO_NUM      27
38 #define PCLK_GPIO_NUM       25
39
40 #elif defined(CAMERA_MODEL_M5STACK_PSRAM)
41 #define PWDN_GPIO_NUM        -1
42 #define RESET_GPIO_NUM       15
43 #define XCLK_GPIO_NUM        27
44 #define STOD_GPIO_NUM        25
45 #define SIOC_GPIO_NUM        23
46
47 #define Y9_GPIO_NUM          19
48 #define Y8_GPIO_NUM          36
49 #define Y7_GPIO_NUM          18
50 #define Y6_GPIO_NUM          39
51 #define Y5_GPIO_NUM          5
52 #define Y4_GPIO_NUM          34
53 #define Y3_GPIO_NUM          35
54 #define Y2_GPIO_NUM          32
```

### Right ↗

The same code can run on other ESP32-based camera kits, such as the ESP32 M5Stack, which is slightly more expensive but can run off a LiPo battery

arrow icon) and – after a short wait – your code should be on your board. There'll be a message in the black box at the bottom of the screen saying that the upload was successful (or not). You can now unplug the IO0 to GND connection but leave everything else connected for now. Open the serial monitor in the Arduino IDE, set the baud rate to 115200 (bottom-right corner), and press the reset button on the ESP32-CAM.

In the serial monitor, you'll see a bit of debug code followed by something like:

**Camera Ready! Use ‘<http://x.x.x.x>’ to connect**

If you point a web browser to that URL, you'll see the camera control page. Press 'Start stream' to see the output of the camera.

For a cheap IP webcam, this works well if you're happy using the Arduino IDE to upload credentials. You can get streaming output at up to 1284×1024 pixels (we got a frame rate of about 6fps at this size, and faster at smaller sizes). The fact that it outputs data in a standard format, and so works well with open-source IoT hubs, is a bonus. There's also some rudimentary face recognition, but we found this to be quite unreliable.

The cooling on the board does struggle if you're streaming continuously. The board gets very warm, and some users report it stopping working. We'd expect this to be a bigger problem if the unit is enclosed, so if you plan to use a case of some sort, you may need to take this into account with a heat sink or even a fan.

### UNDER THE HOOD

Getting to this point has been a little fiddly, but not too complex for anyone familiar with microcontrollers; however, going further can be a bit more complex. In principle, you should be able to integrate this camera with your projects in all sorts of ways.

There is a library for basic image gathering available at [hsmag.cc/uIC0JF](https://hsmag.cc/uIC0JF), and an image recognition framework at [hsmag.cc/uHtRDf](https://hsmag.cc/uHtRDf). Both of these are designed to work with the ESP Integrated Development Framework (IDF), but can be got to work with Arduino if you would rather program in that way. You will need to be fairly comfortable in C programming to have a reasonable chance of success in developing with these. Hobbyists used to dealing with user-friendly wrapper libraries may struggle.

One thing that has made the ESP8266 and ESP32 products easy to use is a community that's built firmware to make it easy to flash new features onto your device. Just take a look at the range of community options for the Sonoff smart relays for examples.

We've come across a couple of other firmwares for the ESP32-CAM:

- ESPHome, which helps you set up a smart home system using ESP devices (see [esphome.io](https://esphome.io) for more details).
- AutoCamera, which creates a wireless network on the device and when you connect to this network, your browser will automatically show the video stream. This uses the same method that some WiFi networks use to force you to a login screen as soon as you connect. See [hsmag.cc/VAFmHQ](https://hsmag.cc/VAFmHQ) for details.

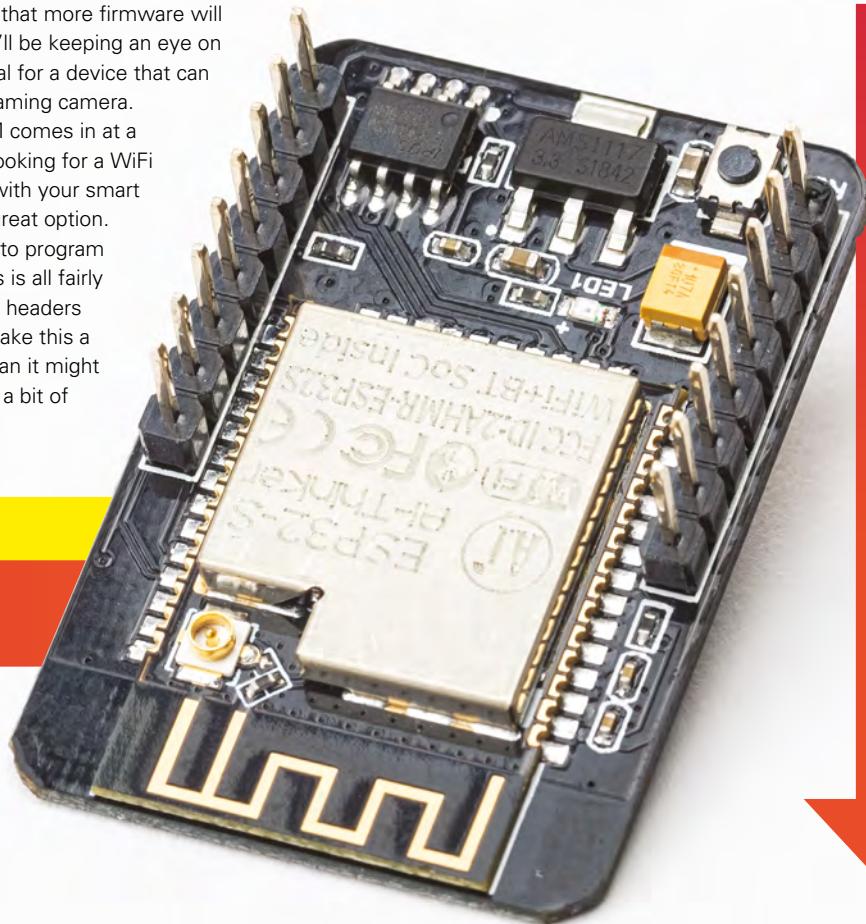
The ESP32-CAM is fairly new – it came out in early 2019 – and it's quite possible that more firmware will be available in the future. We'll be keeping an eye on this, as there's lots of potential for a device that can do more than a standard streaming camera.

The AI-Thinker ESP32-CAM comes in at a fantastic price, and if you're looking for a WiFi webcam that you can link in with your smart home system, then this is a great option. You'll need to do a little work to program it and create a mount, but this is all fairly straightforward. The soldered headers and lack of mounting holes make this a slightly more complex task than it might otherwise be, but it's nothing a bit of

hot glue can't fix. The external antenna port on the board is a great addition if you're looking to place this at the far reaches of your WiFi network.

If you want to go deeper and integrate a camera into your projects, you can do quite a bit with the ESP32-CAM, provided you've got the programming skills to make it work. Casual users who want features beyond basic streaming webcams may wish to check what additional code is available for them to build off before making a purchase, but for under £7 you may prefer to take the plunge and see how you get on. □

**"**  
You'll need to do a little  
work to program it and  
create a mount, **but this is**  
**all fairly straightforward**  
**"**



DIRECT FROM SHENZHEN

**Right**  
The AI-Thinker  
ESP32 module  
provides brains  
and connectivity

ONLY THE  
BEST

# Power to the Raspberry Pi

A variety of products for powering your Raspberry Pi

By Marc de Vinck

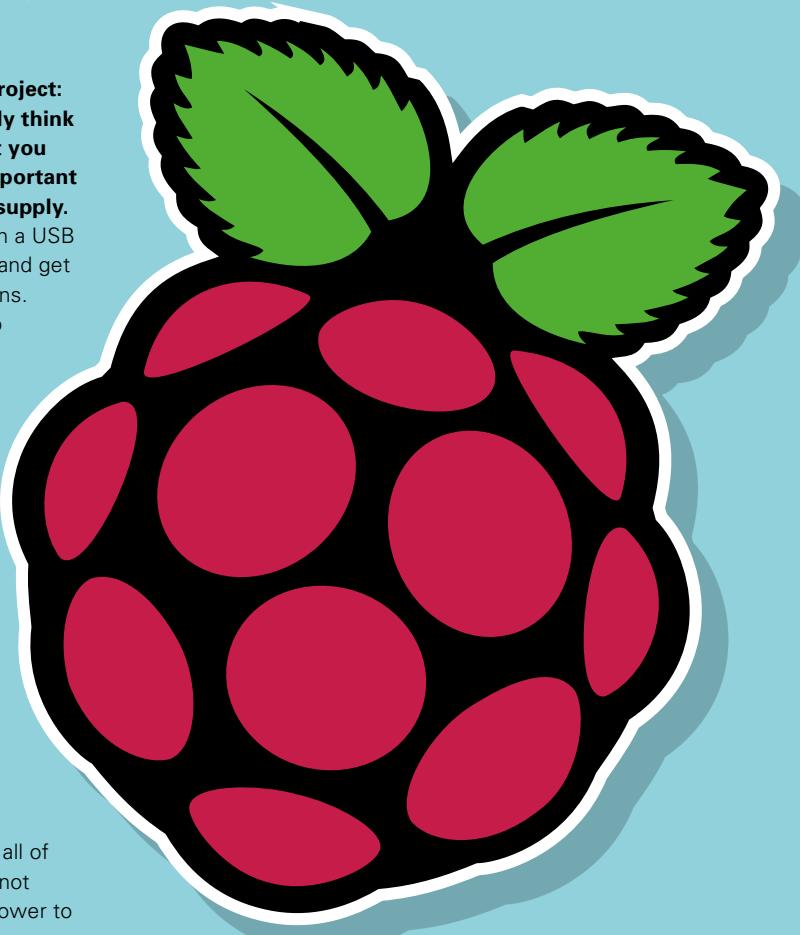
 @devinck

**P**owering your Raspberry Pi project: it's not something you usually think about when starting out, but you will soon realise just how important it is to have a proper power supply.

Typically, you can simply plug in a USB cable, or some kind of 'wall wart'-style plug and get to coding. And this is true in many applications.

The challenge comes when you need to go portable with your Raspberry Pi. It's not as simple as something like an Arduino. Raspberry Pi is a fully-fledged computer and has very specific power needs. And more importantly, you can't simply pull the plug and expect it to be happy and shut down. And that's where many of these accessories shine. From being able to have uninterruptable power, to portable power, or going completely off-grid for extended times. It's all possible if you know where to look.

There are so many power supplies available for your Raspberry Pi, and in this article, we're going to look at some of my favourite solutions. If you have a specific power requirement, make sure to check out all of these manufacturers as there are dozens, if not hundreds, of different products for getting power to your Raspberry Pi.



# PowerBoost 1000C vs PiJuice HAT

Keep your Raspberry Pi going with intermittent power

**ADAFRUIT** \$20 | [adafruit.com](http://adafruit.com)

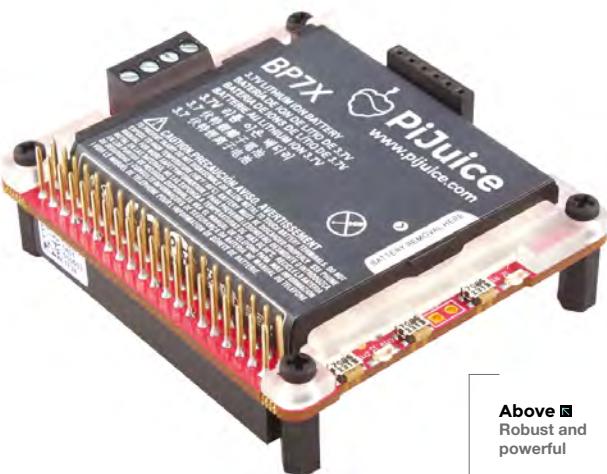
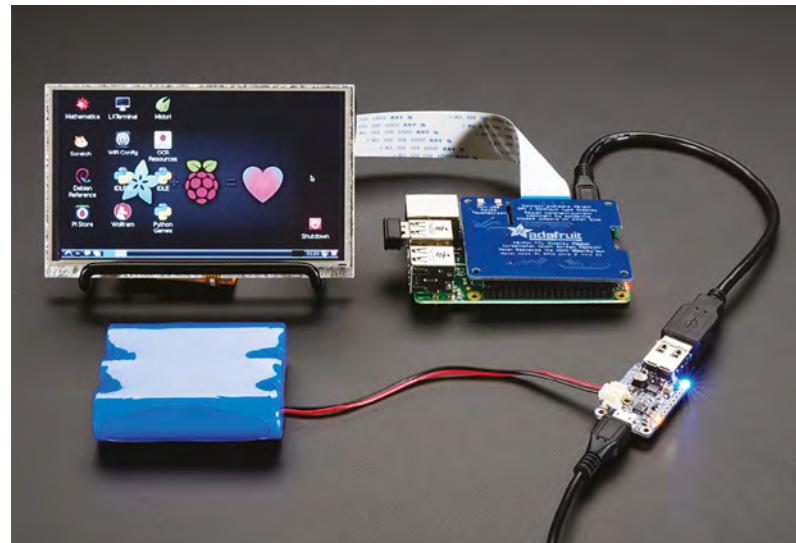
**PISUPPLY** \$60 | [uk.pi-supply.com](http://uk.pi-supply.com)

**Below** A smart power supply for your project

**Credit**  
Adafruit licensed under CC

**N**eed an uninterrupted power supply for a portable Raspberry Pi project? How about hot-swappable capabilities? If so, the PowerBoost 1000C from Adafruit is for you! What makes this board really useful is the ability to make a Raspberry Pi portable with a battery pack, but also have the ability to recharge the system via an integrated load-sharing battery charger circuit.

The charger circuitry is powered directly from a micro USB port and will recharge almost any 3.7V / 4.2V LiIon or LiPoly battery at a maximum rate of 1000mA. The board also features a couple of handy LEDs that will give you feedback about the charging cycle and power. Portable power is fairly easy with a simple battery, but being able to charge a LiPo via USB and never lose power to your Raspberry Pi, which happens to really hate losing power, is really valuable and affordable with the PowerBoost 1000C. □



**Above** Robust and powerful

**Credit**  
Pi Supply

**T**he PiJuice HAT is a great way to take your Raspberry Pi project off the grid with an alternative energy source, like solar or wind. Its compact format fits nicely on top of the traditional Raspberry Pi formats, including an area for the battery. This is a feature-rich power supply, with LED indicators, real-time clock, power monitoring, and lots of options for power in and out. It's a compact, cost-effective solution for your next project. Set your Raspberry Pi free with the PiJuice. □

Its compact format fits nicely  
on top of the traditional  
Raspberry Pi formats

## VERDICT

**PowerBoost 1000C**

Incredibly easy and convenient.

**10**/10

**PiJuice HAT**  
A very versatile power supply for Raspberry Pi.

**9**/10

BEST OF BREED

# Raspberry Pi PoE HAT

RASPBERRY PI ◆ \$20 | [raspberrypi.org](http://raspberrypi.org)

The official Raspberry Pi Power over Ethernet (PoE) HAT allows you to transmit power over Ethernet cables to your Raspberry Pi. It's a convenient way to get power to your project – just keep in mind it has a few limitations. The board is only compatible with Raspberry Pi 3 B+ and Pi 4, and it won't fit inside some cases due to the height of the components. Also, pin accessibility is difficult when the board is plugged in, but many applications don't require the GPIO pins, and there is a workaround using pass-through headers. For cooling, the PoE board features an on-board fan that is controlled via I<sup>2</sup>C. To ensure it gets the power needed, make sure you are using 802.3af-compliant power-sourcing equipment. □

The board is only  
compatible with Raspberry  
Pi 3 B+ and Pi 4



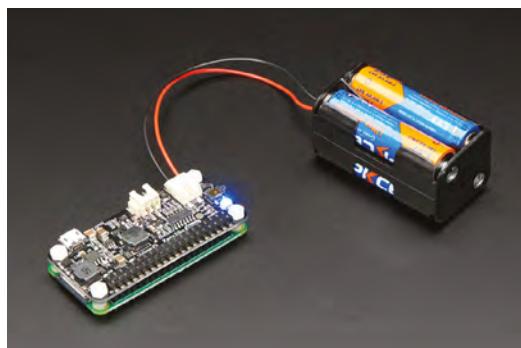
Left ◆  
From the creators  
of Raspberry Pi

**VERDICT**  
Get power and  
data with just  
one cable.

9 / 10

# Zero2Go Omini

ADAFRUIT ◆ \$20 | [adafruit.com](http://adafruit.com)



**N**ow, this is a clever little power supply for your Raspberry Pi. The Zero2Go Omini will allow your project to use multiple power sources. It's smart enough to be able to detect the highest voltage and either increase or decrease the power to a clean 5V. There are LEDs that will let you know exactly what is going on. All you have to do is supply the Zero2Go with 3.5 to 28VDC and get building! This is a great add-on that allows for uninterrupted power from up to three different power sources. □

Left ◆  
Let this board  
decide what's  
the best option  
for power

Credit  
Adafruit licensed  
under CC

**VERDICT**  
A great add-  
on to your  
Raspberry Pi.

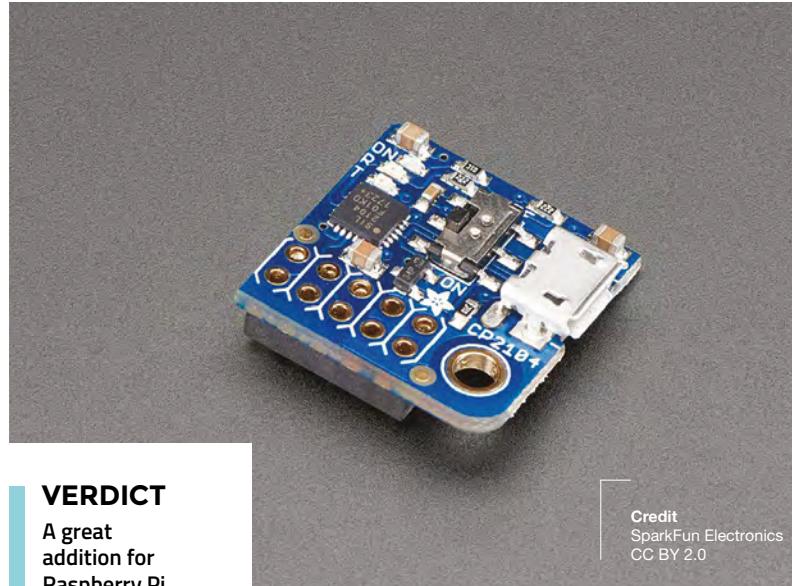
9 / 10

# Adafruit PiUART

**ADAFRUIT** ◆ \$7 | [adafruit.com](http://adafruit.com)

The Adafruit PiUART is not a HAT, or a pHAT, but it is a really handy add-on board for your Raspberry Pi project that is perfect for certain situations, especially when creating a headless setup, since you won't have to plug and unplug the USB all the time when developing your project.

You connect via a micro USB cable for power, and it also makes a serial connection so you can access your Raspberry Pi's console. The board has two LEDs to show when you're actively connected to RX and TX, which is really handy. And there is even a physical on/off switch for when you are done debugging! Best of all, this little board comes fully assembled, so you can just plug it in and get going. Just keep in mind this board is great for Raspberry Pi Zero and A+, but other boards may pull too much power from your computer's USB port. Head over to the product page to read more about the requirements. □



Credit  
SparkFun Electronics  
CC BY 2.0

**VERDICT**  
A great addition for Raspberry Pi Zero or A+.

**10/10**

## SPARKFUN BENCHTOP POWER BOARD KIT

**SPARKFUN** ◆ \$22 | [sparkfun.com](http://sparkfun.com)

At some point, everyone who experiments with electronics comes across an old computer, and other things like printers and keyboards. Sometimes they can be repurposed to serve as a network drive, or run a CNC machine, but other times they are best used as a paperweight. It's those paperweight computers that still may have some useful parts, and that's where the SparkFun Benchtop Power Board Kit really shines.

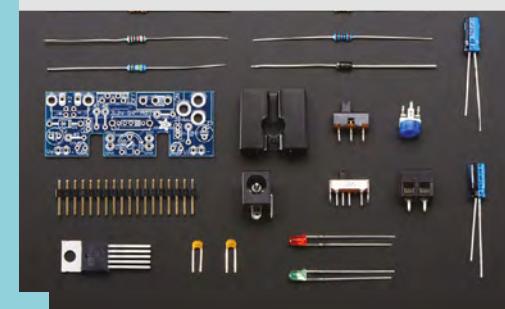
After about 30 minutes of assembly, and a connecting it to a used ATX computer power supply, this kit will provide you with quick and easy access to the typical voltages needed when developing physical computing projects. You will have access to four different voltages, 3.3V, 5V, 12V and -12V. And each will have its own replaceable 5A fuse and ground connector. This is a simple and affordable way to add a lot of functionality to your electronics workbench.



## ADJUSTABLE BREADBOARD POWER SUPPLY KIT – V1.0

**ADAFRUIT** ◆ \$15 | [adafruit.com](http://adafruit.com)

Need a simple way to power up your next breadboard project? The adjustable breadboard power supply kit from Adafruit just might be the perfect solution. Once soldered together, you can use any DC power supply (max of 20 V) with a 2.1 mm plug to get beautiful, and adjustable, power to your solderless breadboard. It's a good beginner project that adds another functional tool to your workbench for prototyping more Raspberry Pi projects.



BEST OF BREED

# Pimoroni Wide Input SHIM

PIMORONI ◆ \$30 | [shop.pimoroni.com](http://shop.pimoroni.com)

With the Wide Input SHIM from Pimoroni, you simply supply a 3V to 16V DC voltage in via a centre-positive 3.5 mm barrel jack connector, or the +/- pins on the SHIM, and it can provide 5V/2A output for your Raspberry Pi.

That's really handy when you need power from a specific source, like your car's 12V system or other non-5V power supply.

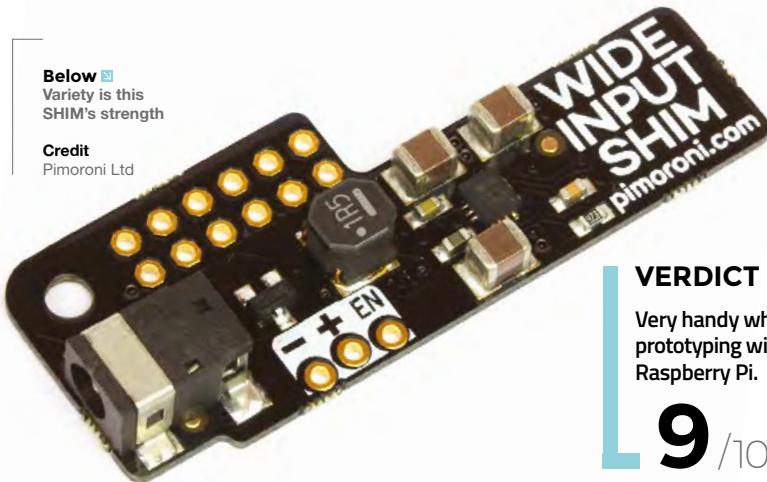
The SHIM comes with a 2x6 female header, which requires a little simple soldering, a 3.5mm OD male to 5.5mm OD female adapter cable, and a 3.5mm OD male to tinned wires cable, which makes getting up and running really simple and fast. □

Below ◆

Variety is this SHIM's strength

Credit

Pimoroni Ltd



VERDICT

Very handy when prototyping with Raspberry Pi.

9 /10

# LiPo SHIM

PIMORONI ◆ \$13 | [shop.pimoroni.com](http://shop.pimoroni.com)

The LiPo SHIM from Pimoroni enables you to quickly add a LiPo battery to your Raspberry Pi project for simple portability.

The SHIM is so small, it's perfect for adding discrete power to your Raspberry Pi Zero, but will work with other 40-pin varieties too. Pimoroni chose the TPS61232 step-up boost converter from Texas Instruments, which provides up to 96% efficiency. There are on-board LED indicators for power and low battery, making it very easy to use. Keep in mind this SHIM will not be able to charge the LiPo battery. It's a trade-off for being so simple, affordable, and small. □

Below ◆

A simple solution for adding external power

Credit

Pimoroni Ltd



VERDICT

Easy and affordable power management.

8 /10

## MINI PUSHBUTTON POWER SWITCH

POLOLU ◆ \$3.95 | [pololu.com](http://pololu.com)

The Pololu Pushbutton Power Switch is a simple, yet fairly sophisticated, power control alternative to using a bulky mechanical switch. The main advantage is having a push-button-based latching power control that allows for one push to turn on the power and another push to turn it off. This version (see the website for more options) operates from 2.2V to 20V and can deliver a continuous current up to 6A.





# THE BEST PROJECTS FROM HACKSPACE MAGAZINE

## THE ULTIMATE SKILLS, TRICKS, AND MAKES



### BUILD A DRONE

The ultimate guide to making  
your own quadcopter



### MUSIC BOX

Build a touch-activated  
music box with no  
coding required



### LASER-CUT TURNTABLE

Create stunning 360° animated  
GIFs with this geared turntable

**£12.99**

**FREE WORLDWIDE SHIPPING**

BUY TODAY AT  
[STORE.RPIPRESS.CC](http://STORE.RPIPRESS.CC)

# Can I Hack It? Hot Wires?

Can we hack a kids' electronics kit?



**Les Pounder**

@biglesp

Les Pounder loves taking things to pieces and seeing how they work. He teaches others how to be makers and tinkerers at events across the UK. He blogs at [biglesp.com](http://biglesp.com)

**Right**

They fit together like Lego and can be used to build a series of cool projects. Oh, and we can hack it!



**I**t is easy to think that learning electronics is now just about picking up an Arduino and writing some code, but that is not the case.

From the 1970s, there have been electronics kits that offered children the opportunity to create simple circuits from a box of components. Typically, these kits were laid out like a board game, but in the early 2000s the kits were reinvented and now come under many names, Snap Circuit and Hot Wires being the two most popular. But are they any good, and can we hack them? Well, there is only one way to find out.

## GENERAL CONSTRUCTION

The components are formed into 'bricks' of many different sizes and shapes. Each piece is made from a thin plastic that can be easily broken, so take care when taking them apart. But, when we do take

them apart, we see a component – such as an LED, button, or transistor – soldered to press studs. These press studs are used to make physical connections/circuits in the experiments, providing a firm, yet easily disconnected, mechanical connection. The press studs are the same as used in clothing, so finding replacements and fitting is a relatively easy process.

## ELECTRONICS

The electronics kit comes with many different components, all packaged in plastic bricks that connect using press studs. However, inside the bricks are common through-hole electronic components. This means that they can be easily replaced, should they break. Yes, you can buy replacement parts online, but a capacitor brick will set you back a few pounds, whereas soldering a new capacitor will cost you 10p!

## YOU'LL NEED

Hot Wires

## COST

£42.98

## WHERE

[hsmag.cc/ZAFyhc](http://hsmag.cc/ZAFyhc)

## HACKABILITY

Oh boy, this is a fun kit to hack! Let's start with the basics – we can connect the components to boards such as Adafruit's Gemma M0 and Circuit Playground Express using crocodile clips. So this means we can power the circuit and use the GPIO (general-purpose input/output) pins to control and react to the same components. But, what if we wanted to make circuits using Hot Wires and Raspberry Pi or Arduino boards? Well, for that we need to use some crocodile clip to female or male header wires, which can be bought for under £10. So this means that an electronics kit made for kids can be used to help children learn more about electronics with computers and microcontrollers.

But let's do a little more hacking. Remember that the plastic bricks can be taken apart. Well, this means that we can replace the components with others. So, if you really want NeoPixels, rotary encoders, or anything else, then so be it. In fact, with a little 3D printing or laser cutting, we can make our own compatible bricks. An Arduino or Raspberry Pi brick would be awesome!

Hot Wires are designed for children to learn simple electronics skills, but they can grow with the child as their learning progresses. By adding new components, and connections to powerful devices such as Raspberry Pi, we can introduce the child to new ideas while still ensuring their safety. We wouldn't introduce a bare resistor or LED to a small child, but with the bricks, we can easily do so. By making your own bricks, using a laser cutter or 3D printer, there is no limit as to what you can do with Hot Wires. The easily found press studs also provide a strong mechanical connection, while still remaining easy to use for little hands.

If you have a little engineer in the family, then this would be a good start to their career. ☺



## SMALL BEGINNINGS

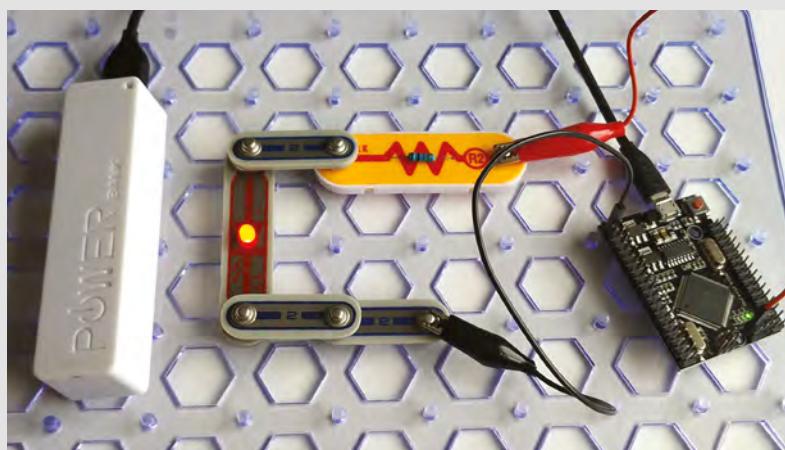
Your author has first-hand experience of the power of Hot Wires and Snap Circuits. Children who attend Blackpool Raspberry Jam have been using them for four years, and they have proven to be an excellent tool. When a new child comes to the Jam, we like to make them feel welcome, and so we dig into the big box of circuits and build something together. The helicopter launch project is always a big hit!

Learning electronics is a lifelong skill and every day we will learn something new, even about things which we are already experts in. These bricks offer children, and adults, a safe and engaging entrance into electronics. We have often used them to design circuits and work out ideas before creating more traditional projects. Their greatest strength is their simplicity – it removes any doubt when trying something new. We simply focus on building, rather than doubting.

**Above** There are a series of components in the box, from standard resistors and LEDs, to NPN/PNP transistors and custom ICs

**Below left** Using some easily found crocodile clip to header pin cables, we controlled a circuit using a spare Arduino board

**Below** The components are standard but soldered to press studs used for mechanical connections



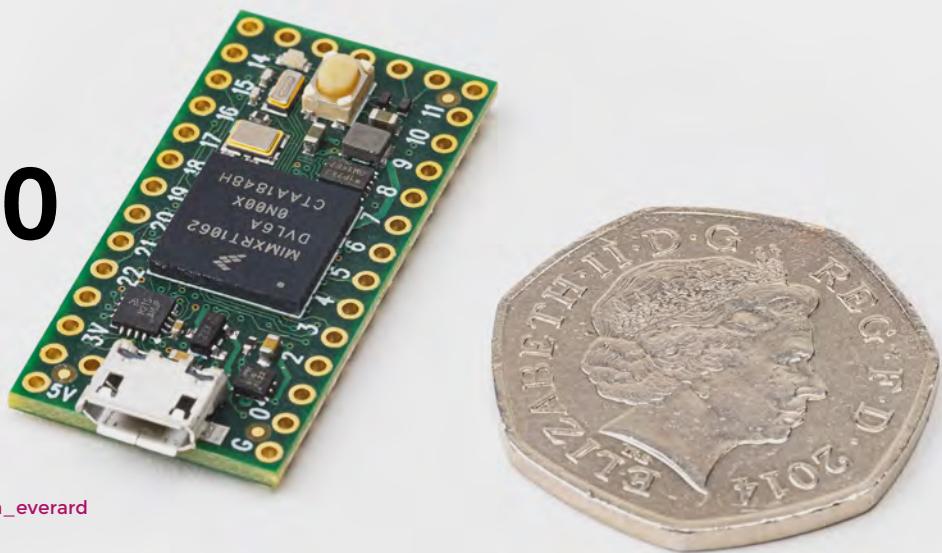
# Teensy 4.0

A 600MHz microcontroller

**TEENSY** ♦ \$19.95 | [pjrc.com](http://pjrc.com)

By Ben Everard

 @ben\_everard



The **Teensy 4.0 specs are impressive.** A 600MHz ARM processor sounds more like the sort of thing you'd get in a general-purpose computer, rather than a microcontroller. It's based on the ARM Cortex-M7F core, so let's take a quick recap of the ARM cores you'll find on microcontrollers. The M series (as opposed to the A series that you'll find in 'Application' devices, such as mobile phones and Raspberry Pis) are 32-bit cores designed for microcontrollers. There are many other cores not based on ARM Cortex designs, such as the ATmega cores in many Arduinos and the Tensilica cores in ESP8266 and 32 devices. The most common ARM Cortex-M cores are:

**M0** Small instruction set, optimised for small size on silicon, low price, and low power (at least, relatively speaking as these are still significantly faster than AVR chips, such as those found in the Arduino Uno). Based on the ARMv6-M instruction set.

**In microcontrollers, it doesn't really make sense to have a notion of one of these being overall 'better' than the others**

**M0+** A machine-code-compatible upgrade on the M0 that adds a bit more oomph.

**M3** Based on the ARMv7-M instruction set with instructions not present in M0 cores such as divide and multiple-accumulate. Code should run faster than on an M0 core.

**M4** The same basic core as an M3, but with digital signal processing (DSP) instructions. These are used extensively in audio processing libraries.

**M4F** An M4 core with additional acceleration for single-precision floating-point calculations.

**M7F** This includes single-precision and (optionally) double-precision floating-point accelerations, as well as DSP instructions. It's a significantly more powerful core than that in the M3 and M4, with a larger pipeline and branch speculation (this is a bit of a confusing feature, but it can result in faster code run time). There's also the option of tightly coupled memory, which allows you to use a small amount of very fast memory.

## CORE CONUNDRUMS

There are a few others, but they're not commonly used in the hobbyist world. In microcontrollers, it doesn't really make sense to have a notion of one of these being overall 'better' than the others, as it depends so heavily on use-case. M0 cores are the least powerful in this list, but on the scale of microcontrollers, they're still fairly powerful and should accomplish many tasks without draining your power supply or bank balance. However, if you need to do any DSP or floating-point operations, then you'll really benefit from an M4F or faster core.

The M7F core in the Teensy 4.0 is more powerful than an M4F core (such as that in the Teensy 3.6), and it can also run at higher clock speeds – 600MHz in this case (though it may be possible to overclock it in the future). The one feature that really has a dramatic speed increase is the support for accelerated double-precision floating-point operations, but this is quite a specialised use-case.

### Above

The Teensy 4.0 really lives up to its name – it's tiny

## BENCHMARKS

We ran a series of benchmarks on some of the fastest microcontrollers we've got, to compare them to the Teensy 4.0. In each case, the benchmark result is the time taken to complete a task intensive in that particular area. Lower is better.

Benchmark	Teensy 4.0	Teensy 3.6 (240MHz)	Adafruit PyPortal (SAMD51 – 200MHz)	ESP32
Integer arithmetic	6.00	38.00	40.00	54.00
Float arithmetic	28.00	79.00	85.00	151.00
Double arithmetic	30.00	620.00	739.00	614.00
GPIO output	65.00	271.00	451.00	265.00

**Below** ♦  
The Teensy breakout isn't available for sale, but you can create your own with instructions at [hsmag.cc/ZyDVhx](http://hsmag.cc/ZyDVhx)

Take a look at the box above for a comparison of performance with other high-speed microcontrollers. There's no doubt that the Teensy 4.0 is, for almost any case, the fastest microcontroller geared for hobbyists by a factor of about three to five (depending on exactly what you're doing with it). There are a few applications that can really benefit from this speed-up.

The Teensy range has been a favourite device for people working with real-time audio, both because they've historically been fast boards, and because there's a great set of support libraries written by Paul Stoffregen (who also sells the Teensy boards). This includes a drag-and-drop creator and a set of libraries that help you write Arduino code to both create and modify audio signals. The Teensy 4.0 is much faster than the previous version (Teensy 3.6), and has four times the memory. This means that you can do a whole lot more. In audio terms, this means you can do more computationally intensive effects, and more of them.

### AUDIO ADAPTORS

The Teensy 4.0 does work with the Teensy Audio Adaptor Board, but the pins are in slightly different positions, so you have to connect it with jumper wires rather than soldering the two boards directly together, as you could do with the earlier board.

Another area where powerful microcontrollers are looking promising is running neural networks, such as using the TensorFlow framework. At the moment, there's a lot of work going on with this (we looked at the SparkFun Edge last issue). On paper, the Teensy 4.0 looks like it would make a good platform for



this, and there is some support for M7 processors, but as yet, there's not a straightforward process for getting all this running. If you're interested in running TensorFlow on microcontrollers, it's certainly worth keeping an eye out for Teensy 4.0 support.

The Teensy 4.0 is a significant step up on performance over any other hobbyist microcontroller board, and available at a great price. If you find yourself lacking the processing power to do what you need, then there's really no competition at the moment – this is the board you need. □

**VERDICT**  
The most powerful hobbyist microcontroller available at the moment.

**9 /10**

# MONSTER M4SK

Add animated eyes to your Halloween get-up

**ADAFRUIT** ♦ \$44.95 | [adafruit.com](http://adafruit.com)

By Ben Everard

 @ben\_everard

**L**e'ts be honest: these are for one thing, and one thing only – Halloween. There are two 240x240 pixel SPI screens connected to a SAMD51 ARM M4F microcontroller. There are also three buttons if you want to trigger different effects, and an audio jack for sound output.

You'll almost certainly want to use this on the go, so there's a LiPo battery port and charging circuit.

In their most basic use, you just plug the power in, wait a few seconds, and they're on and looking around. There are four mounting holes positioned, so it's easy to tie (or zip-tie) these onto costumes, if it's not appropriate to bolt them on. These can also be used to hold on a pair of hemispherical lenses (sold separately) to give these a more eye-like look, particularly if you mount them inside some other make – imagine these eyes poking out through a piece of clothing or a prop.

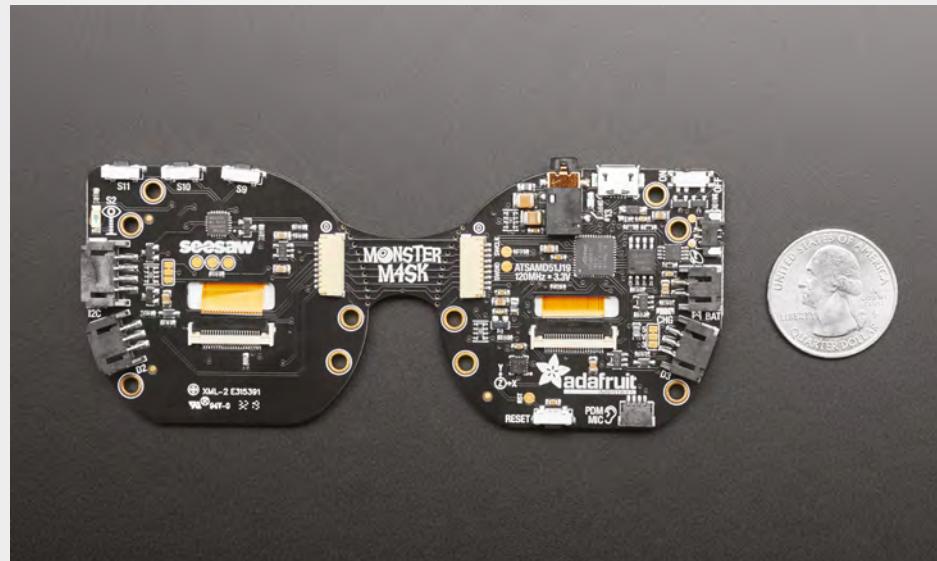
**Below** ♦  
These TFT screens emit light, so work brilliantly in darkened rooms

**Right** ♦  
There's enough connectivity to add quite a bit of extra hardware to your Halloween costume, and power it from your MONSTER M4SK

The quality of the screen and animation is stunningly good. The original name for this eye animation (when it was first coded by Phil Burgess to work on the Teensy) was Uncanny Eyes, and they really are uncanny. For us, it's the animation rather than the realism of the image (though this is also impressive) that gives real personality.

There's plenty you can customise about the eyes, and there's a detailed guide for doing this here: [hsmag.cc/oPgysD](http://hsmag.cc/oPgysD). You can do much of the customisation without getting your hands dirty with programming – the device shows up as a USB disk when plugged into a computer via USB, and you can tweak configurations and image textures without having to recompile.

The mask is designed so the eyes are the same distance apart as on the average human, but if you want more or less, you can detach the two halves of the PCB (see the dotted lines around the nose? Those





snap). This obviously breaks the electrical connection between the two halves of the PCB, but there's a solderable nine-pin JST SH connection. As there's lots of data flowing through this, it can't be extended arbitrarily as electrical interference will distort the signal – around 10cm should be fine, but longer than that and you might find you have problems.

If you need more input or output, there are two three-pin STEMMA connectors, one four-pin I<sup>2</sup>C STEMMA, and a JST connector of a PDM microphone.

#### ON THE SOFT SIDE

There's no getting around the fact that the code to perform this eye animation is complex, but it is well structured and commented, so worth a read through if you're interested in what's going on. You'll find it here: [hsmag.cc/gogAcd](http://hsmag.cc/gogAcd).

This is also a good starting place if you want to use this board for something different. Obviously, a lot of it is very eye-specific, but you'll also find all the details of how to connect to all the on-board hardware.

As the board is fully hackable, you don't have to use it for eyes. You could use it for any project that needs two square screens. Flip them around, add lenses and you've got a DIY VR setup; they could be a screen each for a two-player game; or they could be simply two screens for outputting diagnostic information. Look beyond the eyes and it's a powerful, programmable microcontroller

that you can use just as you would any other microcontroller.

In a world where landfill is full of obsolete gadgets, it's important to be able to reprogram and reuse your technology.

However, despite the repurposability, there's no getting away from the fact that these are designed to be animated eyes. There's something fundamentally endearing about hardware that's well engineered to do a single job – especially when that job is to make people smile. We've looked through the technical

**Above** ♦  
Save time on  
messy carving by  
adding eyes with  
Velcro strips

**“** There's something fundamentally endearing about hardware that's well engineered to do a single job **”**

specs, and how to use it, and how it could be used in the future, but all these are largely irrelevant. What matters with this hardware is how it makes you feel, and it makes us happy. We're used to being surrounded by high-definition screens covered in all manner of images and details, but these are different. Perhaps it's because the form factor is so unusual. Perhaps it's because in eyes we see humanity in each other, and therefore with these, we see humanity in a machine. Perhaps it's because sufficiently advanced magic is indistinguishable from technology and these are magic. Whatever it is, it's one of the most fun gadgets we've tested. □

#### VERDICT

An easy and fun way to add 'wow' to your Halloween costumes.

**L** **10**/10

# Programino

A featureful Arduino code editor

**PROGRAMINO** ♦ from €29 | [programino.com](http://programino.com)

By Ben Everard

 @ben\_everard



**Left ♦**  
The hardware viewer gives details of the board you're programming for

The Arduino IDE has some fantastic things going for it – the range of boards it can program is phenomenal, there's a huge amount of example code for it, and it's simple enough for beginners to use.

However, it also has some limitations – there are few niceties to help you program, and very limited tools to help you manage larger projects.

Programino is an editor for Arduino that lets you write code for the Arduino boards that gives you more tools than the official IDE, but at the same time, doesn't bog the user down with excessive amounts of configuration or customisation. As IDEs go, it's still on the simple side – beginner or intermediate programmers are unlikely to get lost and still find the things they need – yet, it still adds some features that make it easier to write code.

Navigation around your project is easy, thanks to code folding and explorers both for the current file (which helps you jump to a particular section current file), and the project (which helps you keep track of all the files you're using in the current project).

The Tools menu includes helpers for designing LED matrix images (though we'd prefer a few more capabilities here, including the ability to design for RGB matrices), and converting characters. The Web menu has some rudimentary tools for creating HTML output for your projects.

There's nothing revolutionary in Programino, but there are plenty of little things that make life a little easier for the developer. The downside is the lack of hardware support. Out of the box, Programino can send code to a selection of older, official Arduino boards. The solution to this is a button to automatically open your code in the Arduino IDE and then upload it from there.

There's a 14-day free trial. After that, you'll need to pay 29 euro for a private user licence, or 89 euro for a single user business licence. This is reasonable value for money if you're using the extra features.

Programino offers an approachable IDE with enough features to make managing a medium/large project fairly straightforward. Ultimately, IDE choice is a personal thing, but if you're struggling with your current option, Programino is well worth a look. □

**VERDICT**  
A useful IDE for larger Arduino projects.

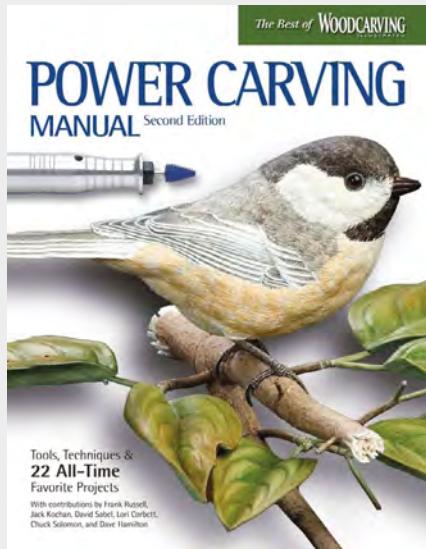
**8**/10

# Power Carving Manual

Woodcarving Illustrated Editors ◇ £12.98 | [woodcarvingillustrated.com](http://woodcarvingillustrated.com)

By Ben Everard

 @ben\_everard



**T**here's a pretty good chance that, if you're reading this, you've got access to a rotary tool, whether that's a Dremel or other brand. Did you know that you can use these tools for wood-carving? There's a range of bits designed for this, including burrs in all shapes and sizes. If you've already got a rotary tool, you can get the basic bits you need to get started power carving for under £10, (although you can certainly spend more as you need a wider range of bits).

The *Power Carving Manual* takes you through the basic techniques, and then on to a range of projects. Each is well photographed and easy to follow. We struggled to reach the level of detail that the authors did, but that's to be expected – we're new to the craft of power carving.

Although this reviewer isn't a particularly skilled traditional wood-carver, he is at least familiar with which end of the chisel goes into the wood, and the experience of power carving is hugely different from chisel carving. Not least, because the grain is far less important. While it is relevant, the carving bits on a rotary tool are much less enslaved by the particular properties of the wood, and you can almost treat it like a homogeneous material.

The tearing of the burrs leaves a very different finish to the smoothness of chiselled wood. This isn't necessarily better or worse, just different.

We thoroughly enjoyed learning, or at least beginning, the art of power carving with this book. Not only have we developed a new hobby, but we've also gained more skill with our rotary tool that we can use for more practical purposes. □

## VERDICT

A fun way of gaining more control over your rotary tool.

9 / 10

# Issue

issue  
#24

ON SALE  
24 OCTOBER

## Brewing

MEAD, CIDER, KOMBUCHA, BEER,  
AND MORE!

### ALSO

- WEATHER FORECASTING
- POWER CARVING
- RASPBERRY PI LAPTOP
- OPEN-SOURCE HARDWARE
- AND MUCH MORE

DON'T MISS OUT

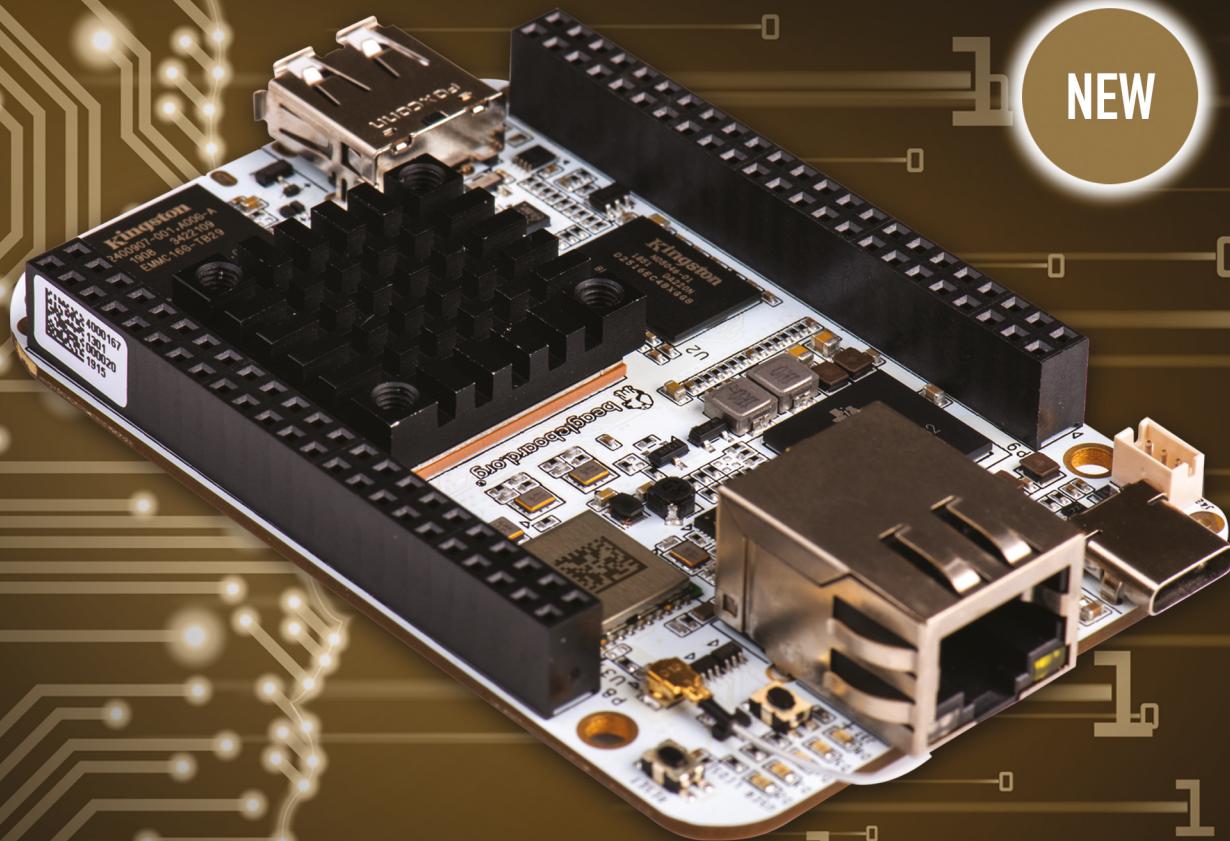
[hsmag.cc/subscribe](http://hsmag.cc/subscribe)





**okdo™**

DESIGN THE WORLD



NEW

SAY HELLO TO  
**BEAGLEBONE® AI**

BUY NOW @ [WWW.OKDO.COM](http://WWW.OKDO.COM)