MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | June 2019 | Issue #19

BUILD A ROBOT

TESTED MAKER GPS MODULES

MAKE THIS ROBOT FROM SCRATCH

PLUS

- Build a polar plotter
- **Metalwork with precision**
- Control time with the 555 chip
- **How makers are rebuilding Kosovo**

June.2019
Issue #19 £6

9 772515 514006 19

Raspberry Pi PRESS

LASER CUTTING **RUST** NUTS & BOLTS **MAKEFEST**

Semiconductors
Electromechanical
Hyperfast
Power
Automation
Connectors
Switches
Passives
Circuit Protection
Interconnect
Crystals
Logic
RFID
IoT
MCUs
Solenoids
Linear
Sensor
Capacitive Touch
IO-Link
Keyfob
Augmented Reality
Digital Omnipolar
Ultra Low-Power
Narrowband
RF Evaluation
Dev Boards
5G Mobile
Virtual Reality
Proximity Sensor
Shunt Sense
Microservice Architecture
RF Antennas
Class-G Amplifier
Rezence Compatible
Embedded Computers
SPI Interface
IoT 3D Alteration
Earth-Friendly Display
RF Directional Couplers
Axis Tilt
NFC
XCVR
FPGA
Rectenna
Sensor Interface
Quantum Computing
Smart Home Technology
Infrared Thermopile
Triboelectric
Recycling Radiowaves
Decimated Power-Efficiency
Motors
Re-Entrant Leveraged Design
Embedded Cellular
Thermocouple Interface
Magnetic Sensors
Bipolar Digital Latching Sensor
Logic Mesh-Networked
Microwave
Cable
PMIC
I2C
SoC
TEGs
Artificial Intelligence
Data Acquisition
Isolators
Filters
Relays
Motors
Zettabyte Era
Internet of Things
Clock/Timing
Immersive Experience
Triboelectric
Remote Control
Tools
EMI
Betavoltaics
Batteries
Solar
PIR Sensor
Na-TECC
TMR
Energy Harvesting
Memory

Welcome to HackSpace magazine

Robots – electromechanical creations that do your bidding, have been gradually making the transition from science fiction to real life for decades now. They do manual work in factories, dangerous work in firefighting, and chores such as vacuum the home. What more could they do though? Well, that's up to you. We're here to help you get started, but after that, it's over to you and your imagination.

We're also digging through our workshop for some retro tools this issue. At the back of our components case, we found some 555 timer chips to inject some life into our projects without the overheads of a microcontroller. Meanwhile, we're also looking at how to breath new life into old tools by restoring them to their original glory. For all this and more, turn the page and get stuck into 132 pages of hacking and making.

BEN EVERARD
Editor ben.everard@raspberrypi.org

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

get in touch at
hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.org

hackspacemag

hackspacemag

ONLINE

hsmag.cc

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.org

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.org

Sub Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Harriet Knight, Sam Ribbits

Photography
Mark McNulty

CONTRIBUTORS

Lucy Rogers, Andrew Huang, Jo Hinchliffe, Cameron Norris, John Proudlock, Mayank Sharma, Dave Astels, Andrew Lewis, Jenny List, Matt Bradshaw, Sophy Wong, Gareth Branwyn, Les Pounder, Marc de Vinck

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.org

DISTRIBUTION

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

SUBSCRIPTIONS

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

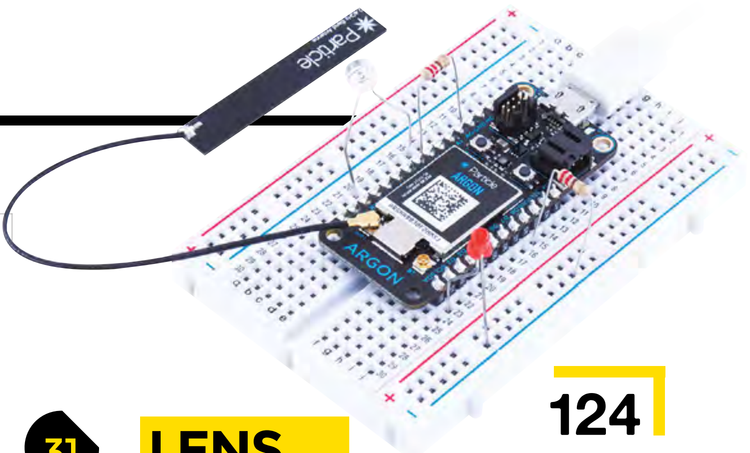
To subscribe
☎ 01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

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

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

Contents



124

06 SPARK

31 LENS

Tutorial

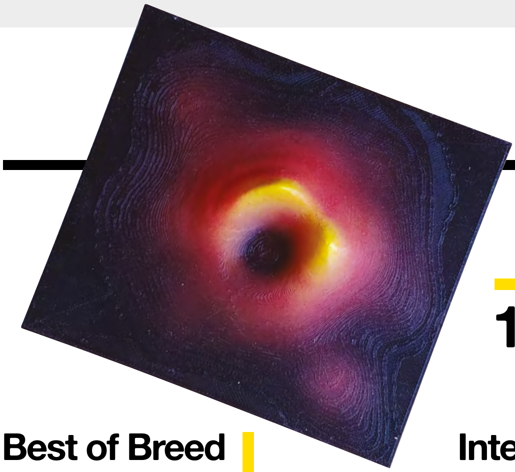
Laser-cut shiny things

100 Make wearable blinkenlights with the power of lasers

Cover Feature

32

122

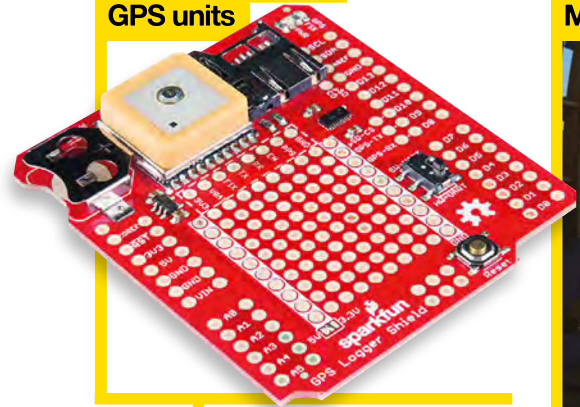


16

06

Best of Breed

GPS units

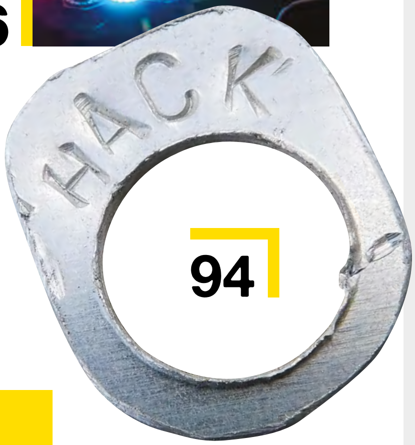


116 Track objects near and far with the power of satellites

Interview

MakeFest founders

60 Makers deserve to party – so get yourselves to Liverpool MakeFest!



94

73 FORGE

113 FIELD TEST

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

The Junk Drum Machine

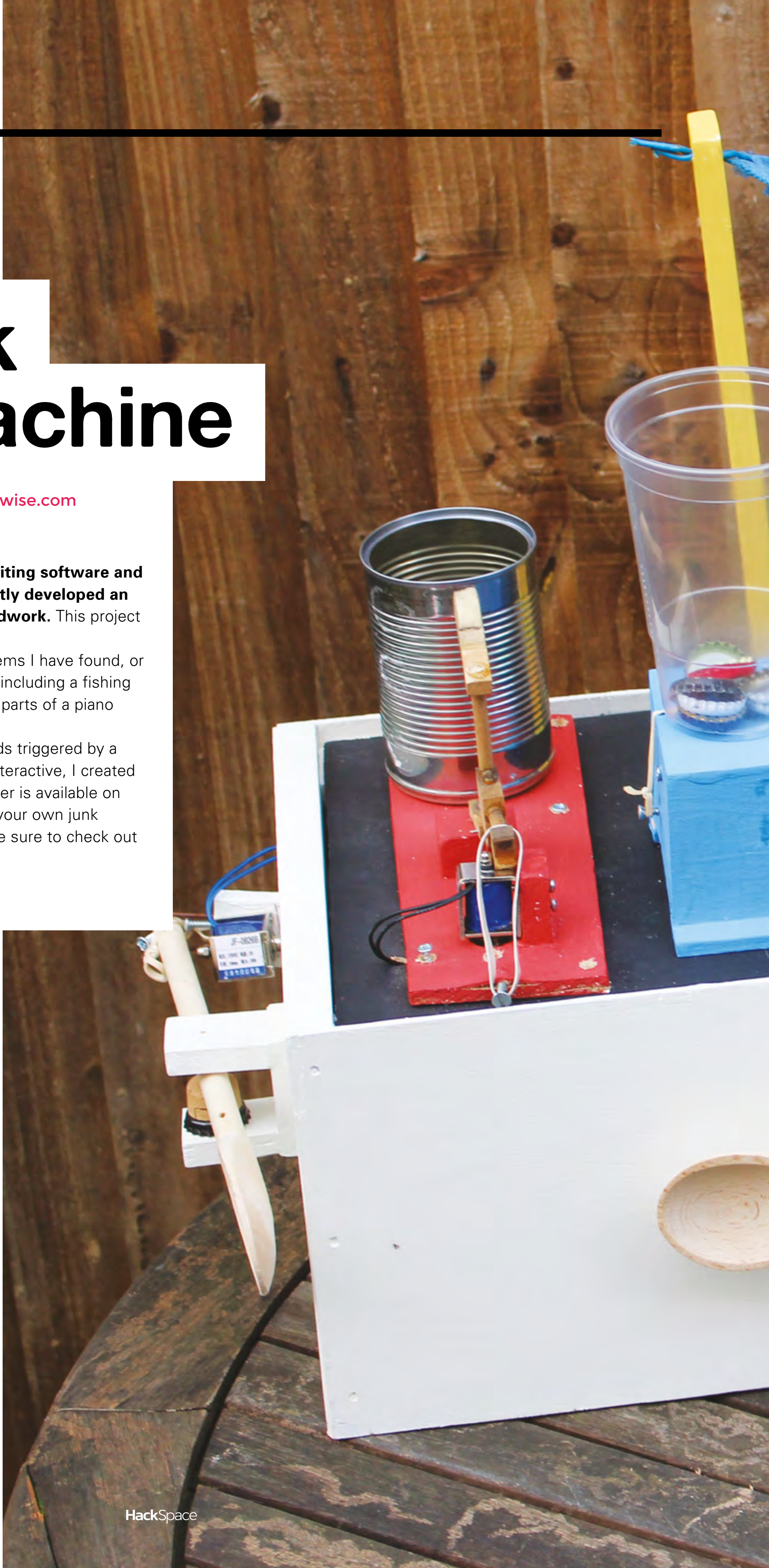
By BanjoWise

banjowise.com

I have many years of experience in writing software and have a passion for music, but I recently developed an interest in both electronics and woodwork. This project brings all these together.

Most of the drums are made from items I have found, or would otherwise have been discarded, including a fishing net from the beach, an empty beer bottle, and parts of a piano rescued from a neighbour's skip.

The internals are quite simple: eight solenoids triggered by a Raspberry Pi via a relay. To make the project interactive, I created a browser-based step-sequencer. The sequencer is available on GitHub (**hsmag.cc/zfOVZE**), so you can build your own junk drum machine or modify it to fit your needs. Be sure to check out the video: **hsmag.cc/zISToX** □



Right Everyone can be a musician, and anything can be an instrument

CRAFTED & TIMELESS
LONDON PRIDE
PiBeat

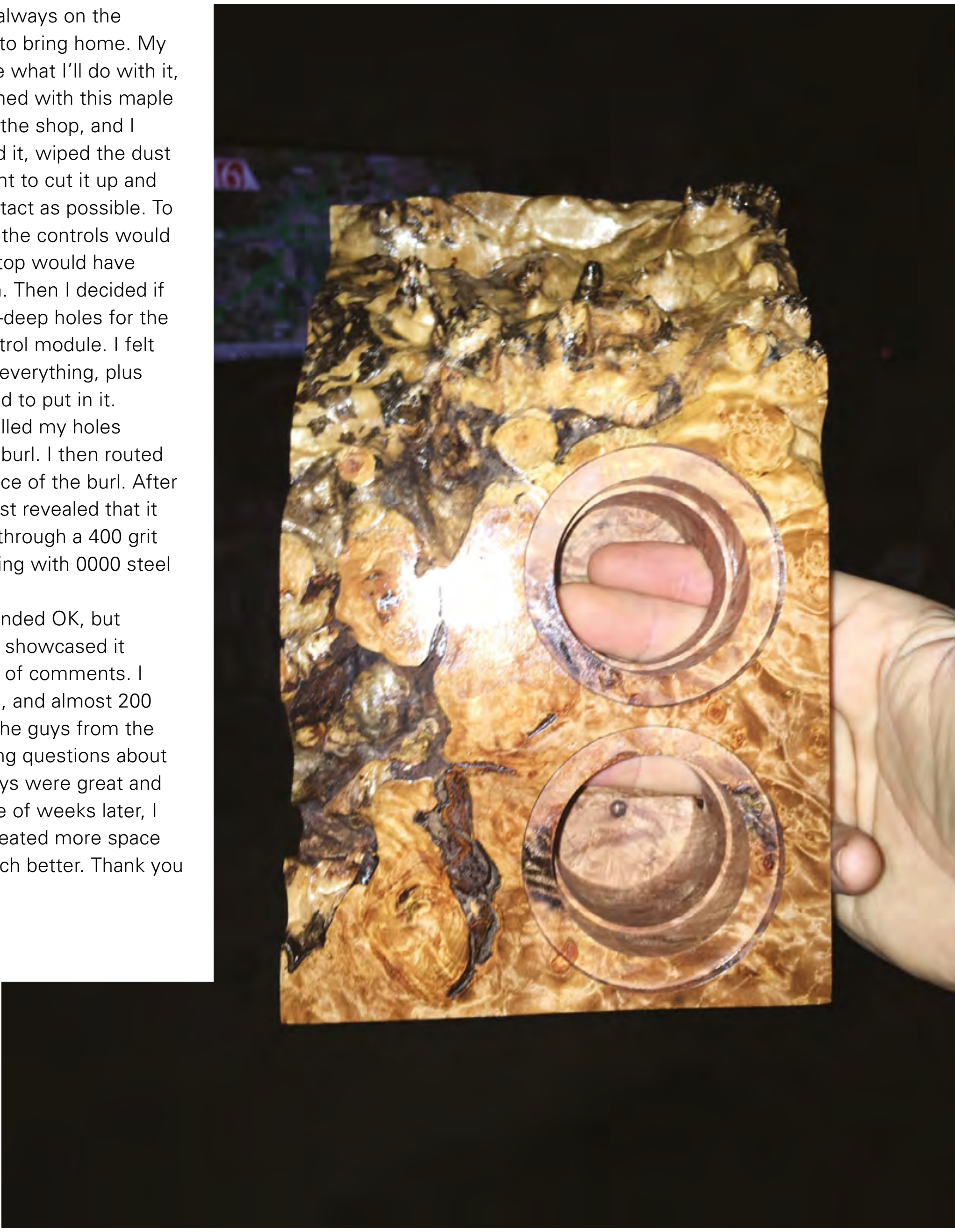
Maple burl speakers

By **Cory Wilton** hsmag.cc/gGfo0Y

I am a self-admitted wood junkie. I am always on the lookout for an interesting piece of wood to bring home. My shop is filled with it. I seldom have a clue what I'll do with it, until it just 'hits' me. That is what happened with this maple burl. I was perusing the piles of wood in the shop, and I imagined this burl as a speaker. I grabbed it, wiped the dust off, and the wheels started turning. I didn't want to cut it up and piece it back together. I wanted to keep it as intact as possible. To maximise the size of speakers I could put in it, the controls would have to be somewhere besides the front. The top would have been the logical choice, but there was no room. Then I decided if I put the controls on the side, I could drill extra-deep holes for the speakers and an extra-deep cut-out for the control module. I felt this should give me plenty of room to connect everything, plus enough room for the oversized battery I planned to put in it.

I marked the location of the speakers and drilled my holes roughly three inches deep in the 3⅜-inch thick burl. I then routed a grove for the speakers to sit flush with the face of the burl. After all the cuts were finished, a quick dry fit and test revealed that it was time for sanding and varnishing. I sanded through a 400 grit sandpaper and several coats of urethane, sanding with 0000 steel wool between coats.

I really liked the way that it looked and it sounded OK, but it didn't have quite enough bass for my liking. I showcased it on Reddit thinking I'd get 20 likes and a couple of comments. I was really shocked that I got around 5600 likes, and almost 200 comments. The crazy thing for me was when the guys from the audio subreddits started commenting and asking questions about stuff that I didn't even know existed. Those guys were great and taught me a lot about the audio world. A couple of weeks later, I took it all apart, went back to my router, and created more space inside the burl for air flow. Now the bass is much better. Thank you Audio Dudes of Reddit! □



Right
"It made me very nervous using a three-inch-long straight bit as a makeshift pattern bit, but it did get the job done, and I can still count to ten on my fingers. I considered that a win"

Wobble Garden

By **Robin Baumgarten**

hsmag.cc/TNXbJO

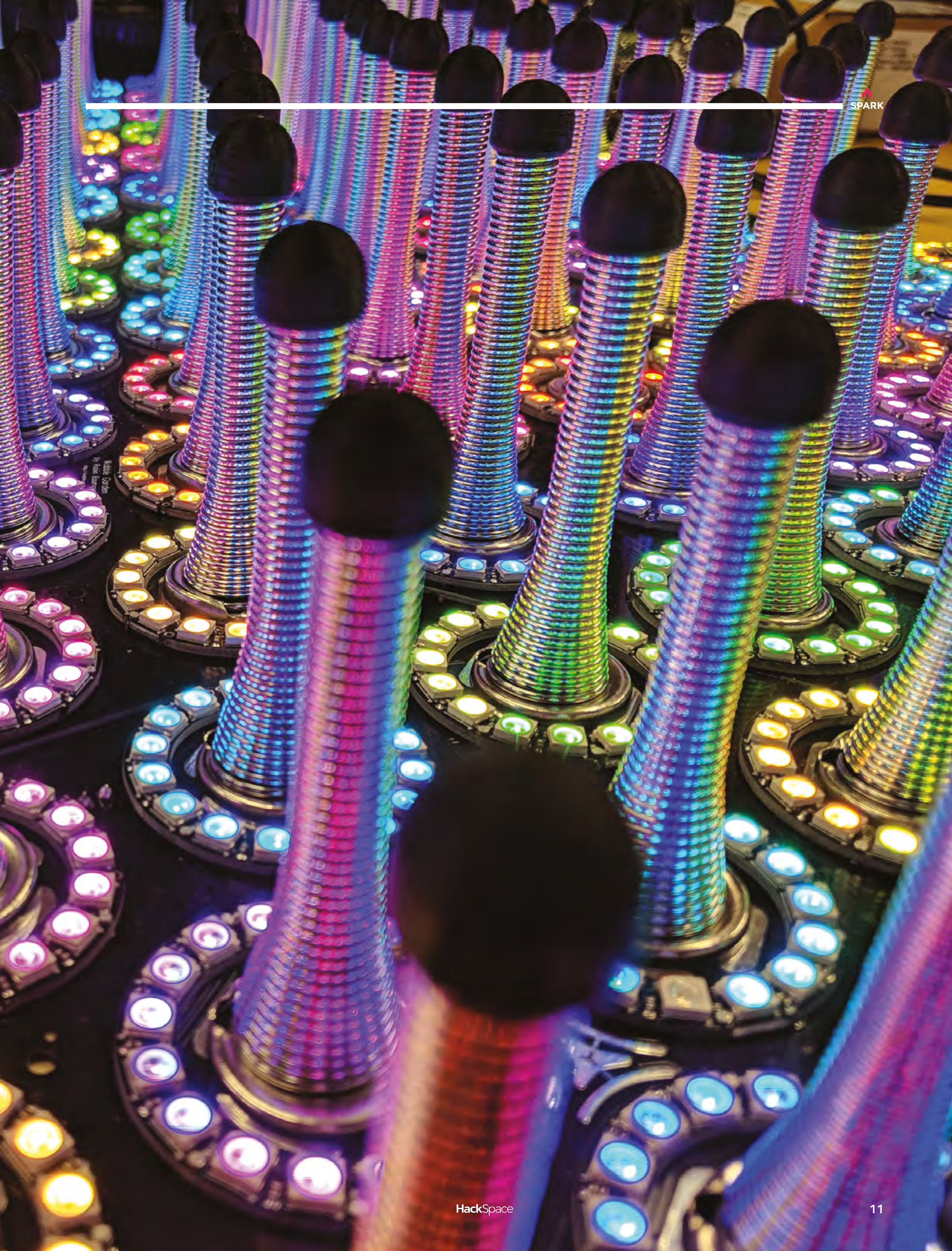
Robin Baumgarten is an independent hardware game designer based in London. Coming from an artificial intelligence and commercial game development background, he is now building award-winning interactive installation art and experimental custom controller games. He's also a keen game jammer, wannabe nomadic, and photographer.

Wobble Garden is an experimental hardware platform that invites you to explore a mysterious LED spring garden by touching and wobbling an arrangement of sensing springs. The current prototype of the installation has 48 springs and LED rings and is scalable to a several-metre large installation. The bright and colourful lights reflect off the springs and create a unique audio-visual and tactile experience, with a selection of different modes available such as games, interactive animations, and an audio experience. The hardware is currently in development and will be made available in a modular form soon. The software will be open, so other developers can create their own interactive experiences on Wobble Garden.

There's a large version of this called Quantum Garden, which is been developed in collaboration with Quantum Physicists of the University of Turku, Finland: **hsmag.cc/NubuGo**.

Quantum Garden is an interactive installation, visualising a quantum computing simulation, controlled by interacting with 228 touch-sensitive springs. A colourful visualisation on LED rings invites players to touch the springs and influence the outcome of the simulation, and then shows the result in a series of abstract circles. □

Right ↗ This looks great in still images, but the shimmering effect of it in action can't be captured in print


Wobble Garden
by Robin Baum

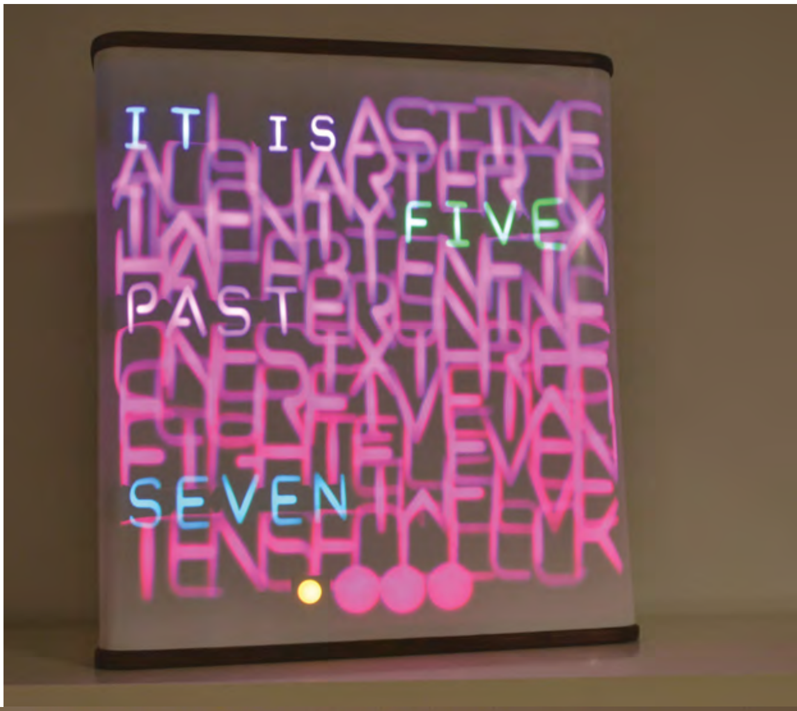
Word clock

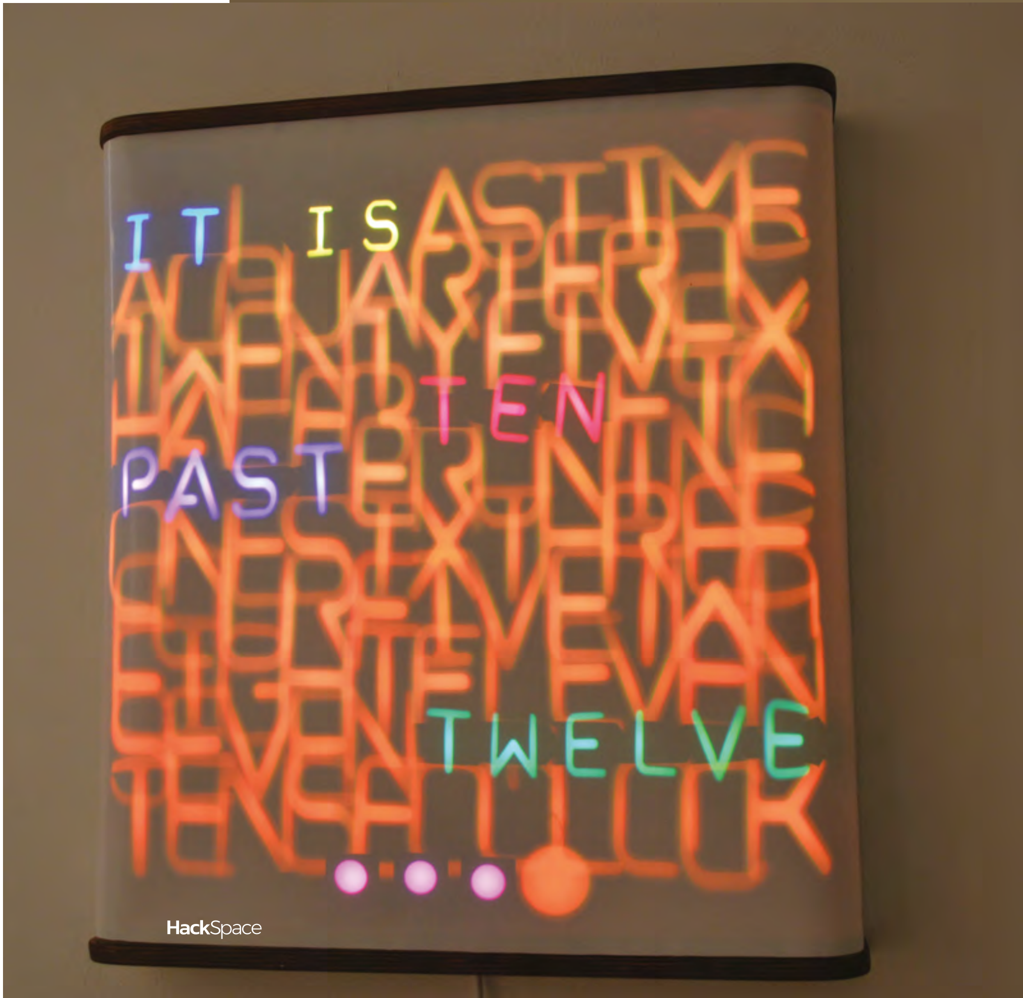
By **Mosivers**

hsmag.cc/SwgwCw

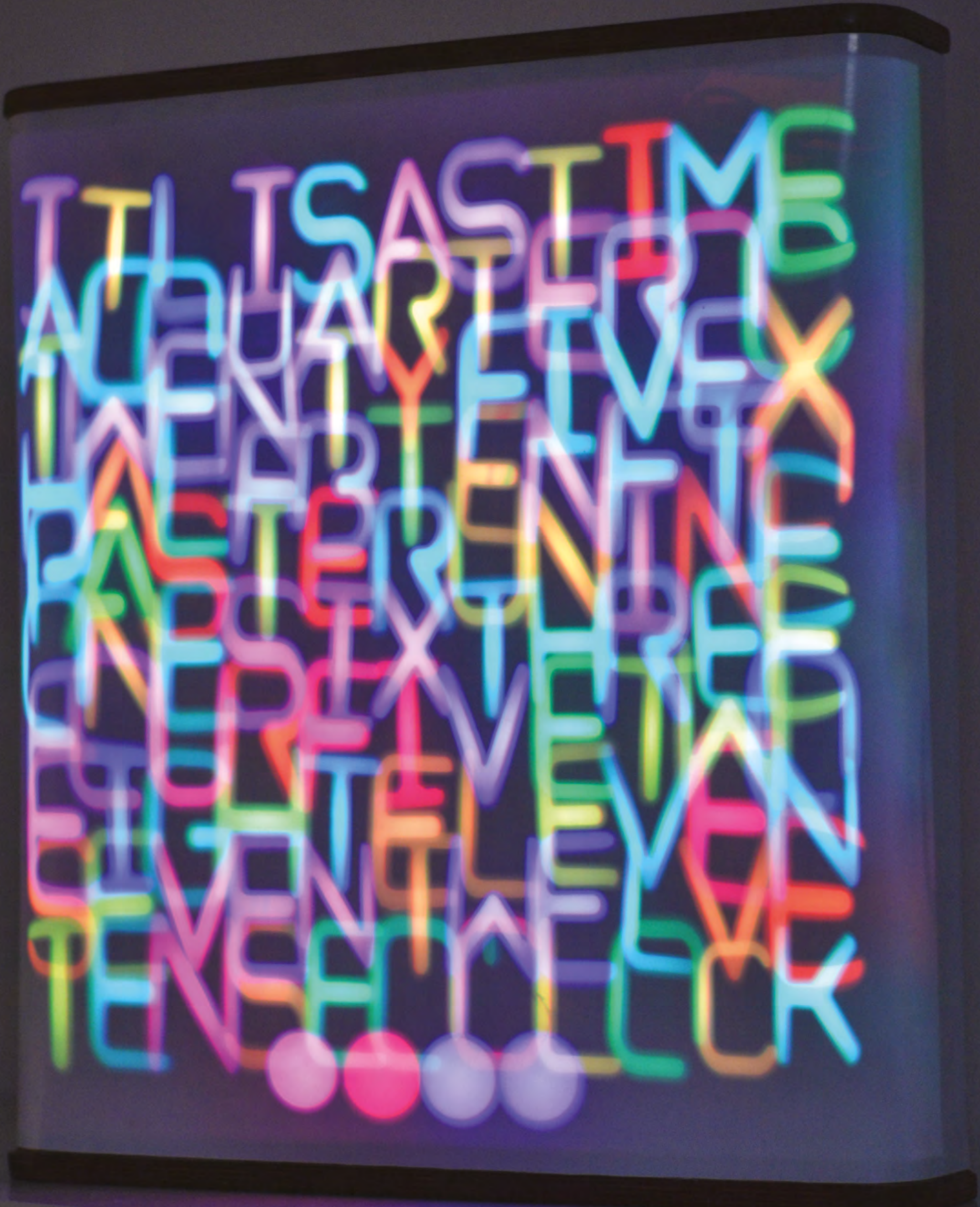
A friend of mine and I were making a regular word clock for his girlfriend as a Christmas gift. During this, we noticed that it is possible to project the letters from the back onto a white sheet of paper. Moreover, we were able to create interesting effects by bending the paper so that individual letters change size and become blurred. After that, we tried to come up with a design for a word clock which makes use of this effect by being able to move each of the 114 letters and dots using a servo.

We knew that this would be a challenging project, but it turned out to be even more tedious than we thought, because you basically have to repeat every step 114 times. However, in the end, I think we created something original and unique. □





Right While we've seen many word clocks in the past, this is by far the best-looking



Lego weather station

By Team Hackeet

hsmag.cc/s0ejWa

We made this weather station in Lego as part of a contest launched on Hackster.io.

The idea was to build something that anyone could easily repeat, to show that technology can be fun and poetic and not just cold or dehumanised, as it may sometimes appear.

Just ask the station for tomorrow's weather and it will elegantly turn the carousel to show the corresponding character (the swimmer if weather is nice, or the one with the umbrella if it rains for example).

The station is managed by a Raspberry Pi 3, on which we installed Snips.ai for speech recognition, and Hackeet low-code platform for programming, while forecasts are retrieved from OpenWeatherMap. □

Right We're working on a British version that can show the four conditions: drizzle, rain, damp, and storms

Objet 3d'art

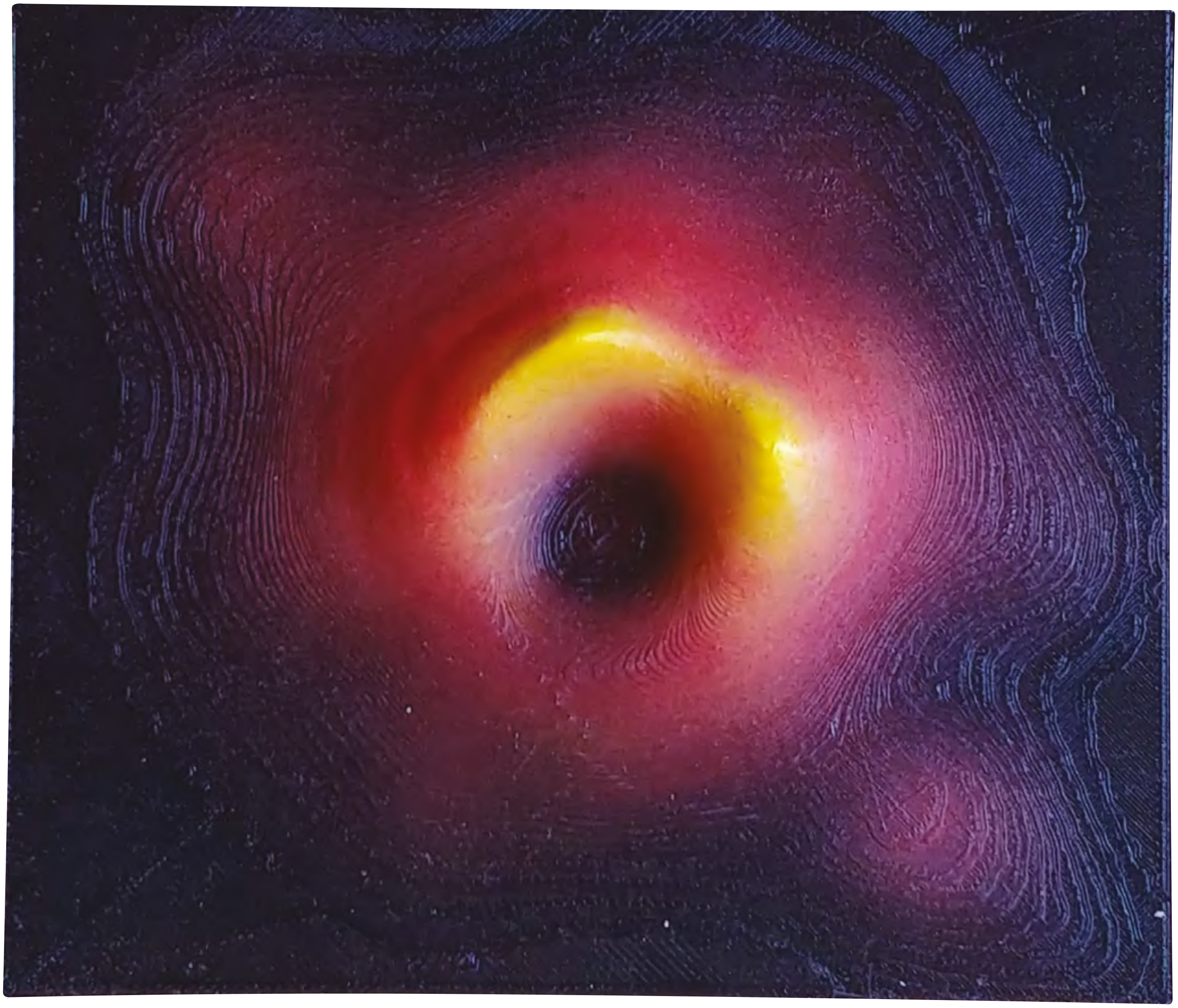
3D-printed artwork to bring more beauty into your life

The science world got a treat in April 2019 with the publication of the first ever photograph of a black hole. Well, kind of – black holes suck in light, so you can't actually take a picture of one directly. The image was instead generated by tons of data gathered by the Event Horizon Telescope array, made up of individual radio telescopes around the world. This brilliant original piece of work by Dean Segovis takes that photo and takes it to (as the Prodigy would say) another dimension.

There's a feature in Cura that enables you to extrude a colour upwards out of a 2D image, as long as it's high-contrast, making it 3D. You can change the dimensions, and the smoothing of the image – based on the video that Dean has put up on his YouTube channel, the defaults look pretty good, with a touch extra smoothing.

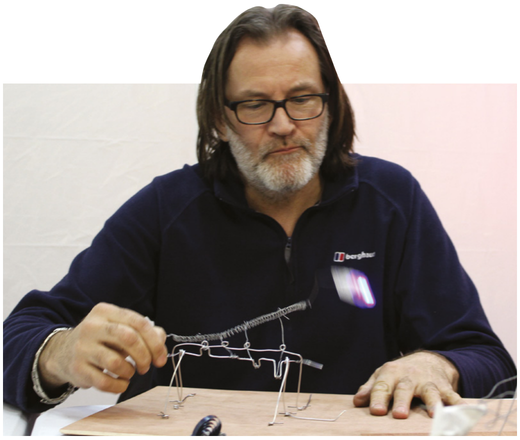
From Cura, you can export to G-code or an SDL file for printing. The model was then airbrushed to match the colours of the photo. Bingo – a black hole for display on the wall. It's slightly smaller than the real thing at 120 mm square (black hole M87 is 6.5 billion times the mass of the sun), but it is just as good a talking point. □

hsmag.cc/ayCuVt



Meet The Maker: Dan Morrison

The master of perfection



Dan Morrison makes beautiful, beautiful things. We're all used to having bits of wires sticking out the back of projects, or things working OK but not being quite finished, or constantly looking at something and thinking 'I'll do it better next time'. None of that for Dan Morrison, the brains and hands behind Blott Works. Everything he makes looks just right. He's blended a mix of techniques and materials and come up with designs that are, at once, familiar and like nothing we've seen before.

"I started thinking of ideas for Blott Works about seven years ago. I'm in my mid–late 50s now, and I've done a few different things in my life. It had always been in the background, the creative angle, making things in different arenas.

"I was working on software development in London during my 30s, moved up to a small town in Yorkshire, and got a job as a funeral director.

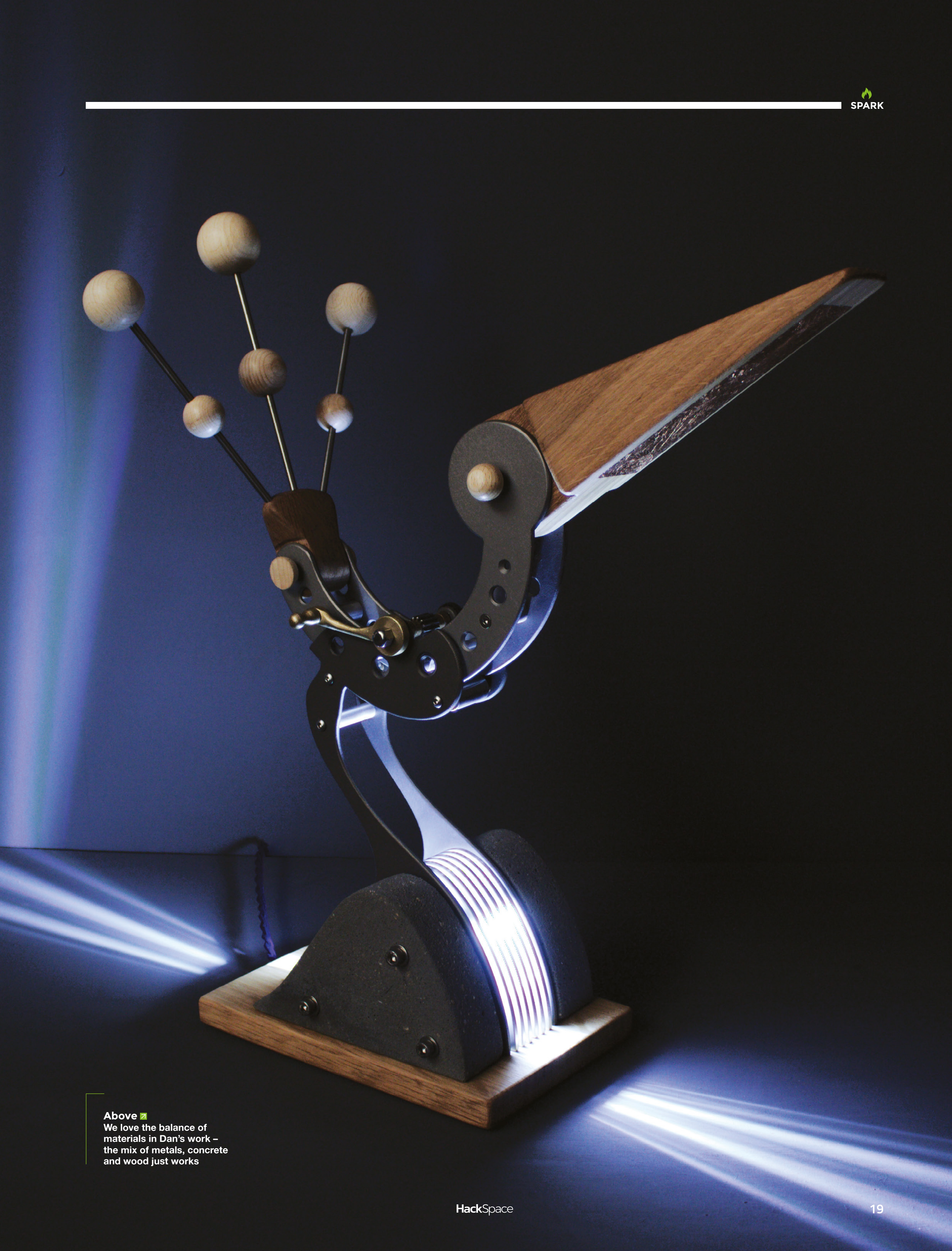
"It was a pretty random move – I'd been in the middle of the dotcom boom of the 1990s. It was all pretty cutting edge – one day here, gone the next. I was approaching 40, having a great time in London, and it felt like the right time to settle down a bit. I moved up Yorkshire, carried on working in software, then saw an advert for a funeral director. I applied thinking I was never going to get it, and the next thing I knew I was a funeral director. "It was fantastic… You get to help people in a really practical way at the toughest time of their lives. In an hour or two, you can make things so much easier for people. People are traumatised, have no idea what to do, and we can just go in and talk them through it.

"After a few years, I started to think that it wasn't going to be my forever job, and so then it was a question of 'what do I do next?'. I spent a couple of years looking back at my life, at the things that I enjoyed, and tried to distil all those things into a job that would take me through to retirement and beyond. I asked myself: 'What will I do when I retire?' And the answer was that I'd probably set up a workshop and make things. I realised that I didn't have to wait to retire – I just like to spend time in the workshop making things.

> "**My strength is that I can apply myself to** pretty much anything"

"So then I spent six months digging in the garden and making myself a workshop, and I took it from there.

"Originally I was going to start it with my friend Andy Plant, who's a kinetic sculptor. Over the years he's done a lot of big, town-centre kinetic clocks, and similar things at arts centres. And he'd always fancied making products to sell to the domestic market. We spent two years working together – this is when we were both full-time with other things, so it was evenings and weekends – and we designed a clock called pebble clock. It was a lovely two years; we worked really →



Above
We love the balance of materials in Dan's work – the mix of metals, concrete and wood just works

Above Everything's made from scratch and built to last well together, designing this thing, working through the kink process, and getting bits cast and that kind of thing.

ENGINEERING AESTHETIC

"I'm inspired by engineering architecture – typically classic cars, trains, boats, that sort of thing. I pull my sources from all over the place, but there's always a mechanical structural essence to what I do.

"I've been working with Paul Parry. We've been following each other's progress ever since, and collaborating more recently. I was a bit quicker to quit my job than Paul was, perhaps a bit more gung-ho about it, and it's been interesting watching the two processes side by side. The big difference between our work is that everything I make is from scratch, from raw materials, so I don't really recycle or upcycle things. I wish I did a bit, because it looks like a lot of fun. I quite like starting from scratch.

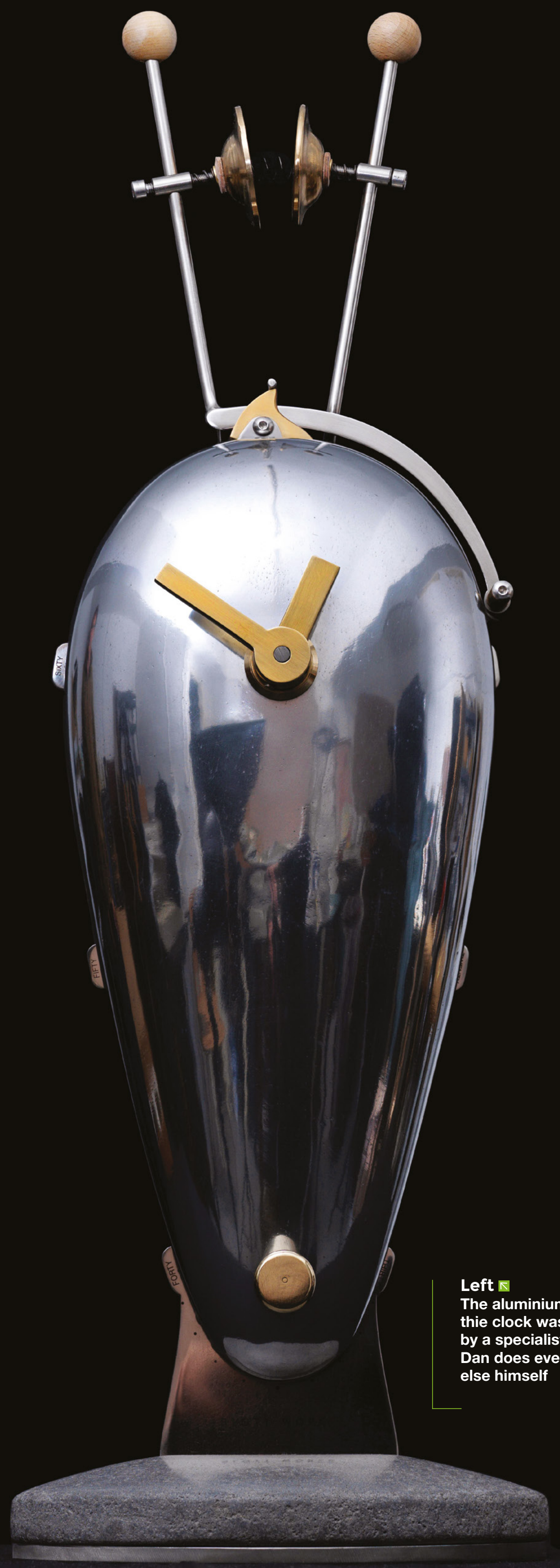
"I cast concrete things. The pebble clock [for example] is very round, bulbous – that's cast at our local foundry just down the road from here. I've spent a lot of time visiting them. I made the patterns, and they cast the aluminium, then they get it all cleaned up. I would love to cast aluminium; I've thought long and hard about it, but I haven't got round to it yet.

"My strength is that I'm not an expert in anything in particular, but I can apply myself to pretty much anything. I look at something and think: 'I could probably do that.' In my work, I do a lot of different things – I do woodwork, metalwork, I've got a lathe, I do concreting and various bits, but with some skills, it is best to take them to the experts.

THE PROCESS

"I tend to start with an idea for movement, like a gear linkage for example; something that will produce a bit of character or do a job. Then I work it and work it, sketch it up. I use Fusion 360 a lot now, which is fantastic. Quite quickly, things take on a personality; I find that very mechanistic things, the more I work them, the more they gain an organic feel to them.

"The combination of the oak and the concrete – the oak softens it, then the concrete will ground it and give it a certain grittiness. Then you've got the more precise metal shapes, the bits and pieces that give the work its form. Stainless steel fittings and a bit of brass to warm it all up, and getting that balance right is really important. Too much wood and it looks wrong; too much brass and it's going look blingy. Balancing those four or five elements is a big part of what I do. It's quite mechanical, industrial, but it's also got a personality to it." □



Left
The aluminium in thie clock was cast by a specialist, but Dan does everything else himself

Latent heat of movement

Just do it

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**

I was collapsed on a mat on the gym floor, encompassed by the smell of rubber and old sweat, and thinking "I hate exercise", and also "just ten more press-ups and then I'm done", when I also started thinking about latent heat. You may remember latent heat from school lessons. For example, the latent heat of vaporisation is the energy required to change the state of a material from liquid to gas – when there's no temperature change (for water at 100 degrees Celsius and 1 atmosphere pressure, this requires 2260 kJ/kg).

I started thinking about my own latent heat of movement – a term I have just made up. My definition of latent heat of movement: the amount of mental energy required to change from one phase to another – sitting at the dinner table to standing up and clearing the plates, hiding under the duvet to being vertical, or from being collapsed on the gym floor to getting into position to do those final few press-ups.

I've written before about procrastination and how I kick-start myself into doing tasks, but I hadn't really thought about the mental energy involved. It's all very well to use the trademarked 'Just Do It' (or the more vernacular version 'JFDI') but something – or some energy – needs to give that nudge.

So, I am now experimenting with what tricks I can use to reduce the effort required in my latent heat of movement.

I am now experimenting with what tricks I can use to reduce the effort required in my latent heat of movement

Back in the nineties, I worked in the 'Business Process Re-Engineering (BPR)' team at Rolls-Royce. We were trying to improve the workflow through the factory. This included reducing the number of processes each item had to go through, and the distance required for each part to travel between processes. This is not so easy in day-to-day living, but some things are possible. Like having a standing desk. I could then move from my desk to my electronics bench by just taking a step and turning around. And once I'm moving, it's easier to keep moving, so lining up or collecting a few things at once before I settle back down into my next 'phase' would be helpful.

But it doesn't really solve the problem of me contemplating moving for a lot longer than it takes to actually do the moving. Which is a shame, as I have a thousand really cool projects I want to work on, but it's so much easier to continue to sit and stare at the laptop than stand up and find the parts. □

Flex PCBs: Add a new dimension to your electronics

Getting bent out of shape

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.

If you've ever taken apart any modern mobile gadget, you've probably seen them: various yellow-brown flexible circuit boards, snaking in between tight spots. Flexible PCBs are essential for building anything that's compact or lightweight.

I use flexible PCBs in many of my projects. For example, I helped design a line of educational electronic stickers for Chibitronics that use flexible PCBs as a core technology. Perhaps the neatest part about flexible PCBs is that they allow you to literally add a new dimension to your designs. Rigid PCBs must stay two-dimensional, but flexible PCBs can be folded into origami-like structures that allow you as a designer to access the third dimension.

From a nuts-and-bolts standpoint, designing for a flexible PCB is pretty similar to designing for rigid PCBs. Instead of a rigid FR-4 core, you have a thin (12.5–50 micron) polyimide core. Compared to rigid PCBs, flexible PCBs can achieve the same or better trace widths, and vias work similarly between the two. The main caveat is that flexible PCBs use a 'coverlay' instead of a solder mask. The coverlay is an additional layer of polyimide material, and it's essential for the reliability of the flexible PCB. With a coverlay, a flex PCB can survive tens of thousands of bending cycles; replace that with a solder mask, and it can survive just a few bending cycles. The downside of the coverlay is that it's a die-cut or laser-cut film, as opposed to the photographically imaged ink of a solder mask. This means that the alignment and minimum aperture constraints of the coverlay are much coarser than that allowed by a solder mask.

Until recently, the setup cost for flexible PCBs was prohibitive for small projects

If you find yourself in a tight spot or are trying to make something thin and light, consider trying out flexible PCBs. Until recently, the setup cost for flexible PCBs was prohibitive for small projects, or you had to speak Chinese to access low-cost providers. This has changed, as more board houses are launching English-friendly web stores with quote and order features.

My personal go-to shop, **kingcredie.com**, is capable of building small quantities of flexible PCBs at a relatively reasonable price (a couple of hundred dollars setup fee, plus a buck or two per board beyond that, compared to the several thousand dollar setup fee typically quoted a decade ago). I've really enjoyed having access to the third dimension in my designs; I hope you find it useful to add another 'dimension' to your work as well! □

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**

WRITE TO REPAIR

Thanks for introducing me to the Repair Café movement. There are some things I know how to fix, but I'm not confident with anything that takes mains electricity, and I hate throwing stuff out when it might be repairable. Most companies now abandon customers once they've brought something, so this looks like it can keep my stuff running longer.

Arthur
Bradford

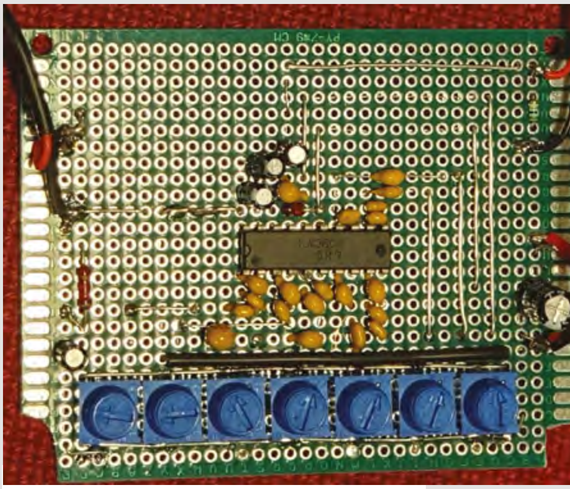
Ben says: There are loads of repair cafés around the country. I have just had a look and there is not (yet) one in Bradford, but if you are able to travel to Leeds, they have them every other month. You can see the schedule of the Leeds cafés here: hsmag.cc/0xNe0I.

HEAR HEAR

Thanks for the article on making an audio equaliser in issue 18. For me, one of the best things about being a maker is that I can adapt things to my (or other people's) needs. We're not all identical, so mass-market products designed for the average person will always leave some people struggling.

Margot
Kent

Ben says: Yes, we completely agree. We've looked at this a few times in the past, and it's a subject we'll certainly cover in the future as well. Watch this space!



OPERATIONAL INFO

Op amps have always been a mystery to me. I've seen them in circuits, doing … something. On one level they sound so simple, yet I never used to understand how they were used in actual circuits. I've been enjoying Dave Astel's articles on electronics generally, but this one was particularly interesting to me. I'm going to order a few and have a play.

Kevin
Utah

Ben says: Yes, do. You can learn a lot from reading, but it's only when you get stuck in with breadboards and wires that you really get to grips with a component. Don't forget to let us know how you get on.

CROWDFUNDING NOW

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.

HestiaPi Touch

Heating controls for your home

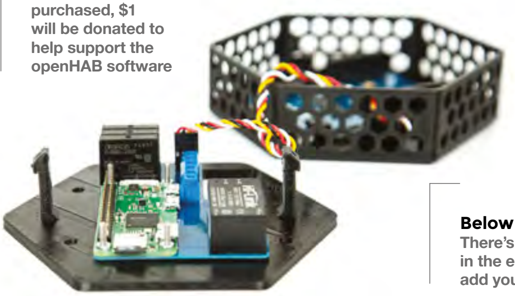
From $95 | crowdsupply.com | **Delivery:** October 2019

The HestiaPi Touch wraps up a Raspberry Pi Zero with a touchscreen, four AC relays, and a temperature sensor. With these, and a little control powered by openHAB 2, you can control many home heating, ventilation, and air conditioning (HVAC) systems. In the US this means systems running on 24 V AC, with five-wire control, while in the EU it means switching mains voltages for four different systems. Everything's housed in a 3D-printed case.

> If it works with your hardware, it should bring all the bits you need to bring **your smart home under your control in one box**

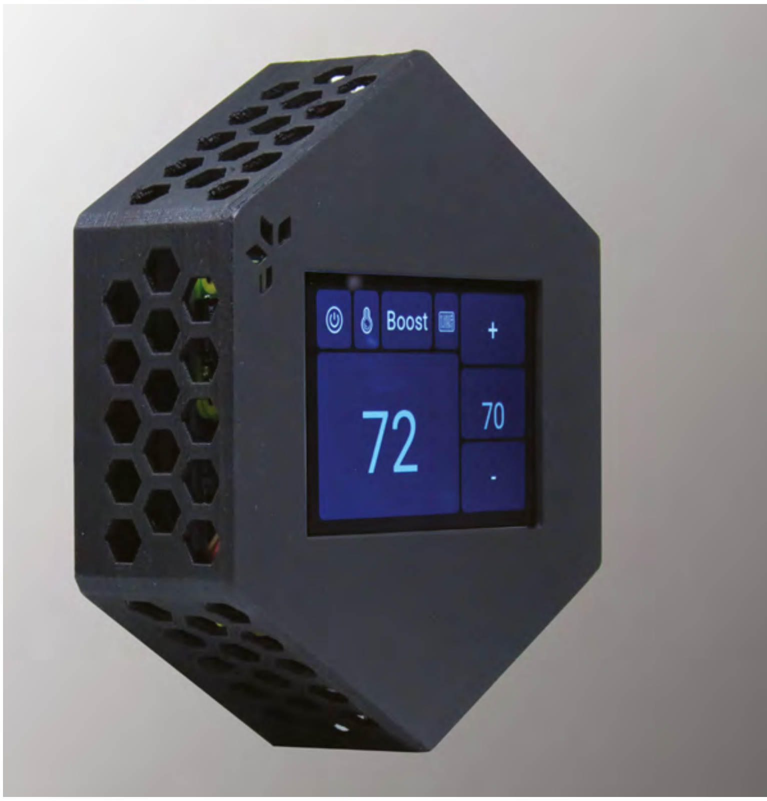
OpenHAB 2 brings in support for many smart home appliances – you can see the full list here: **openhab.org/addons** – as well as a smartphone app for Android, iOS, and Windows 10. As the underlying system is open (and works with MQTT), you can also incorporate your own homemade smart home devices, as well as commercial offerings. Want a set of ESP8266-controlled lamps? This gives you a hub to control them from. A Raspberry Pi-powered doorbell? This user has integrated one: **hsmag.cc/CRwQ0M**.

While we haven't been able to test one out, this product does look more 'tinker-y' than 'plug-and-play', but if it works with your hardware, it should bring all the bits you need to bring your smart home under your control in one box. □



Below For every HestiaPi purchased, $1 will be donated to help support the openHAB software

Below There's enough space in the enclosure to add your own sensors



Space of the month: Prishtina Hackerspace – Kosovo

prishtinahackerspace.org

In Kosovo's largest city, Prishtina, a group of hackers and makers push a vibrant movement in the city that incorporates lots of startup culture and events, activity around free and open-source software such as the Software Freedom Kosovo (SFK) conference and, of course, a hackerspace.

Set halfway up one of Prishtina's San Francisco-style hills, you feel you have earned your visit to the hackerspace if walking up from the city centre. It's a fabulous space set across three floors, with a great garden and various balconies. It's extremely pleasant to have a local weather system that allows the use of these outdoor spaces for projects.

ALL THE USUAL KIT, BUT MORE BESIDES

There are all the usual trappings of a good hackerspace. A large homebrew CNC machine sits in a hallway mid-move from one room to another; a self-built book-scanning machine sits in another room bedecked with various 3D printers. There's a finely stocked electronics workbench, plenty of office-type spaces, and a great kitchen. Hackerspace user Salem tells us that the 3D printing kit has been used for making prosthetics and replacement fingers and proudly shows us the (unrelated!) table saws, MIG welders, drills, and other tools donated from companies around the globe.

THE BACK STORY

The building is owned by some people who left during the war and never returned. It had become a local dumping space and was at risk of becoming a squat, with people reportedly using and selling drugs in it. The hackerspace team took on the building to give it purpose and to tidy it up and keep the main house maintained in exchange for a lowish rent paid to the remote owners. The garden and outside areas filled with trash have slowly been cleared and repurposed, revealing a great space. →

Right
The walls of the refurbished bar have been made using recycled timber, and upcycled motherboard (with an Arduino Nano on the back) acts as an artistic panel with the temperature and humidity displayed

Below
An LED screen keeps track of beer consumption using a flow meter and a WeMos device

HACKERSPACES, BEER, AND LIQUOR!

The standout feature of Prishtina Hackerspace is the basement bar. It had previously been a bar, and the team emptied it and have it fit for purpose again with their own beer. Lis is a founder member of the hackerspace; he's a master brewer and has an IoT-enabled microbrewery set up in the basement space. His microbrewing kit is rigged with sensors and timers, and as he talks about the process, his phone is being sent reminders to add different hops and ingredients to his current brew. They also (this is a tolerated activity in Kosovo) distil raki, and similarly, Lis's condensing and distilling equipment is kitted out with a sensor attached to either WeMos devices or Raspberry Pis so that the process is quantifiable (it's important, for example, to know temperatures in distillation so you don't make a liquor that contains methanol, which may make you go blind!).

FLOW SENSORS, IOT, AND MORE BEER

The bar has flow sensors on the barrels of hackerspace-brewed beer, and a NeoPixel matrix display on the bar keeps a running tally of that day's consumption. The garden has an automated IoT irrigation system that they have just set up and, although only fledgeling, the small hop plants seem to be thriving under its control. Again, many of the projects here create a lot of learning, but also link to the production of beer.

Above
Bleron has become the official icon of the Prishtina Hackerspace

Right
Once filled with fly-tipped waste, the courtyard garden is now a lovely place to hang out and hack

BLERON, THE ICON OF PRISHTINA HACKERSPACE!

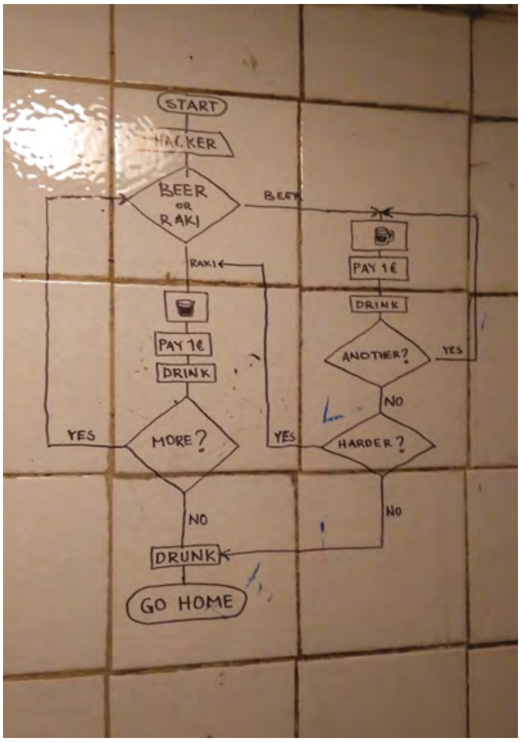
As with many hackspaces, there's fun and games here. Bleron is a member very much involved in the hackerspace; he's become an icon of the space, with a vector art image of his head popping up all over the place, mainly as his image was used as a test piece on the CNC machine when it was set up as a pen plotter. There are lots of members and volunteers who are under 20 years old, and it's great to see there is a real happiness and sense of belonging in this shared workspace, which is so positive in a city that has seen so much hardship and difficulty over so many years. Thanks to all the crew for showing us around. □

Above
Salem teaching some members woodworking skills in one of the two basement workspaces that adjoin the courtyard

Right
A DIY book-scanning machine

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** with an outline of what makes your hackspace special, and we'll take it from there.



#StopWatchingUs
Papa, was hast du getan, als die Freiheitsrechte abgeschafft wurden?
PRISHTINA
HACKERSPACE

meet
build
learn
play
Ultimaker
Ultimaker

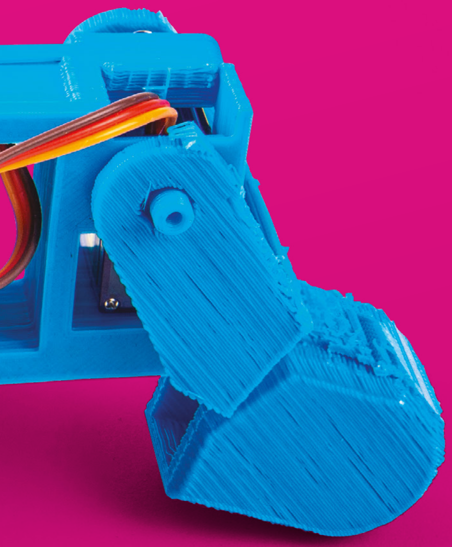
HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

BUILD YOUR FIRST WAL



By Jenny List

We take for granted the ability of mammals and other creatures to walk. It's something that our young minds learn in infancy without understanding what a complex task we have mastered. We are blessed with some of the most intricate and capable actuators imaginable in the form of our arms and legs, but to master them takes a huge array of skills and sensory inputs that we process subconsciously. To try to replicate walking motion in a robot is hard, so it's little wonder that the majority of mobile robots employ wheels or tracks.

KING ROBOT

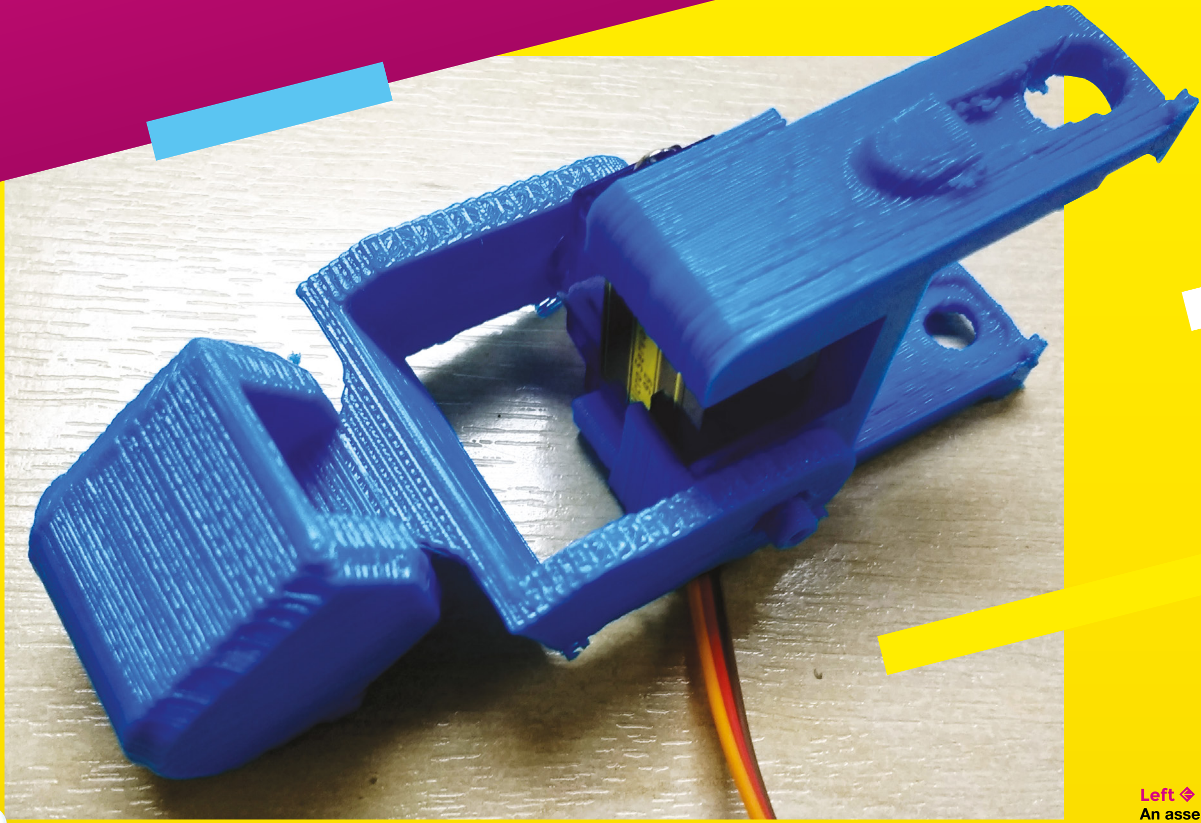
Make a robotic pet, butler or assistant

As humans, we are one of relatively few species that walk upright, on two legs. We are inherently unstable and prone to falling over, so our brains monitor our balance continuously and adjust our muscles accordingly to keep us upright. This is a particularly difficult task in terms of robotic programming, so even the fruits of multi-million-pound research programmes such as Honda's ASIMO or the Boston Dynamics' biped robots are only just starting to walk comfortably. By comparison, four-legged walking in the way practised by most walking animals is a much more stable process, and a four-legged walking robot is well within the capabilities of most people. →

BUILDING THE CHASSIS

We walk by creating a gait using our complex muscles and joints, but there are other motions that can generate movement in a robot. The simplest by far of these use a circular reciprocating motion from a cam or eccentric drive, but there are also robots that use a spring-driven hopping motion. For the purposes of this article, we'll be looking at a robot that mimics an animal leg, as the capabilities of a microcontroller now put its control within anybody's reach.

Our limbs have a range of articulation far greater than is needed for walking, to the extent that fully replicating a human limb makes for an extremely expensive piece of robotics. Happily, the articulation required for each individual task that a human limb can do is only a subset of the whole. So, for a walking leg, the articulation can be reduced to only the axes



Left
An assembled leg

needed for the job. Hip, knee, and ankle joints can be made to only move in one plane, resulting in a leg that only requires three servos. With a rounded foot design, the need for an ankle can be further removed, leading to a leg with only two servos. This is the design followed by the robot we are making here; it has four legs of two servos each – only eight servos for full walking mobility.

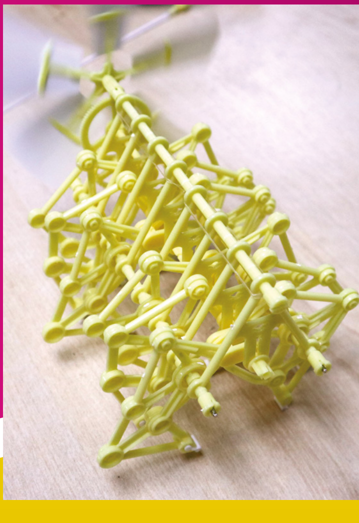
THE MINIKAME, AN INFINITELY VERSATILE ROBOT DESIGN

The Kame series of robots follow a 3D-printed open-source design that has been around for several years now, and which has seen significant refinement and alternative versions of the same basic four-legged robot. Elements from different versions can be combined for custom builds, and a wide array of other controllers substituted for the ESP8266 of the original. A search of the popular Thingiverse 3D model library will turn up a host of different Kame-derived designs. The MiniKame we built is a smaller version originally designed for the HuaDuino, a custom Arduino-based board which required a wait for Hong Kong postage, so we've opted to print a modified version of its body designed for the popular (and readily available in the UK) Arduino Nano expansion boards. If you can wait for a HuaDuino to be delivered, that option is a bit more compact. Or if you really know what you are doing, choose another board entirely, such as an ESP8266 one with WiFi – but, for the purposes of this article, we're sticking with Arduino boards. This is the beauty of open-source: instead of a single take-it-or-leave-it design, there is instead a healthy ecosystem of remixes, meaning that every Kame robot build can be different to suit its owner's needs.

The different 3D-printed parts for our MiniKame came from Thingiverse, and the →

THE STRANDBEEST, A WALKING ARTWORK

Since the early 1990s, the Dutch artist Theo Jansen has created a series of wind-powered walking sculptures. These Strandbeesten (Dutch for 'beach beasts') are typically made from PVC pipe, and are released on the beaches of the Netherlands as autonomous artworks. They are characterised by their multiple legs, using Jansen's own design of linkage, with a pair of rigid triangles linked by a diamond shape that can be manipulated by the crank to produce a practical walking motion in which the foot moves in an approximation to a walking human foot. Strandbeest legs have been enthusiastically taken up by the maker community and turned into all sorts of walking machines; they have even been used to replace the rear wheel on a bicycle frame.



THE PARTS WE USED IN OUR BUILD

The parts used in this robot build are all completely standard and should be available from the usual maker sources. If you opt for a board such as the HuaDuino, it may have to be ordered from overseas, but that is beyond the scope here.

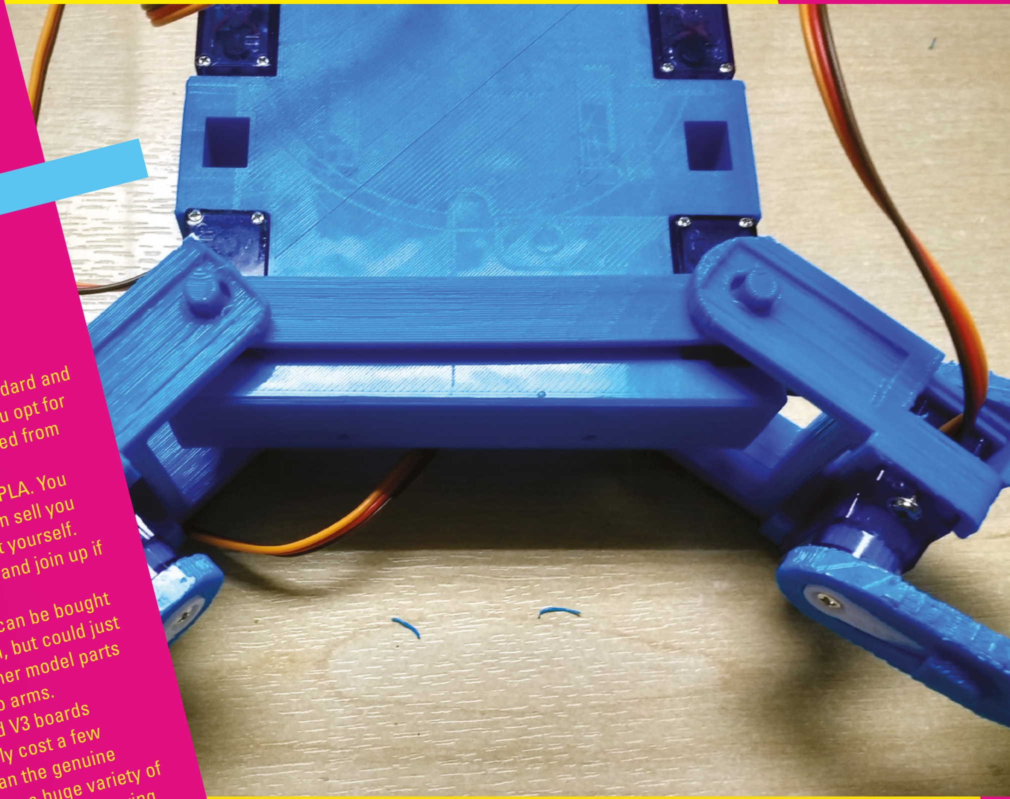
The body and leg components were 3D-printed in PLA. You may find that some commercial 3D-print services can sell you them ready-made, but they are easy enough to print yourself. Your local hackerspace will have a 3D printer – go and join up if you are not already a member.

The legs use eight standard SG90 servos that can be bought from multiple suppliers. Ours came from Amazon, but could just as easily have come from HobbyKing, or any other model parts shop. They should include all screws and servo arms.

The Arduino Nano and Arduino Nano Shield V3 boards are standard commodity items that should only cost a few pounds. Ours were Chinese clones rather than the genuine Arduino boards. Again, they are available from a huge variety of suppliers; ours were already on the bench. We suggest buying quality if you can.

The Bluetooth HC-05 module is yet again a standard component available from many suppliers. Ours came from Amazon.

Finally, the jumper cables are the rainbow ribbon cable variety with single DuPont sockets at each end. All the usual suppliers sell them; the chances are you already have some, but if you don't, you'll find them useful far beyond a MiniKame.



Above ◆
The servo arms fit into the insets in the 3D-printed parts

instructions came from its Instructables page. You'll find the resources at **hsmag.cc/CkdDMO.**

The main body of the Kame is a plastic box sized for its controller board, with a lid, and receptacles for the four SG90 servos that form its hip joints. There are two link bars underneath that clip onto the bottom of the body and locate with the lower part of the hip hinge; in our build, it was these and the main body assembly that both came from the Thingiverse repository for the Arduino Nano board. Meanwhile, the legs are the standard MiniKame items: a thigh assembly, and a combined lower leg and foot assembly that fit together with yet another SG90 servo. We printed four of these assemblies. They, and the body, were printed using PLA in two sessions on a HyperCube at MK Makerspace over the course of an evening.

The first step in assembling a MiniKame is to build all four legs. There are the two larger pieces for the thigh and lower leg, plus a small pin that fits under the knee servo and becomes half of the knee hinge. There is a circular hole in the assembly at the bottom of the thigh piece, into which the pin snaps, then the servo can be fitted above it with its shaft pointing in the opposite direction on the same axis as the pin. All MiniKame servo positions have holes ready for the servo fixing screws; your servo should come with the necessary fastenings in its accessory pack.

The first step in assembling a MiniKame is to build all four legs

There are two distinct mirror-image sets of leg prints; in each case, the spindle of the servo must face outwards away from the end of the robot when it is fitted. In all cases, the lower leg piece has an inset in the shape of a servo arm on the side of the knee which locates with the servo shaft, and a plain hole on the side locating with the pin.

WHAT WENT WRONG?

In this article, we have discussed the building of a toy robot by an engineer with years of experience in creating some extremely complex devices and systems. We've seen quite a few MiniKame builds over the years – it's a popular choice because it works. Despite the descriptions above sounding easy, however, this build turned out to be a difficult one fraught with problems and setbacks. It's important to own your mistakes and shortcomings, and it's also important, if you don't have huge experience, to understand that things go wrong for professional engineers too. Thus we'll run through some of the issues, so that with luck you can avoid them yourselves.

In theory, 3D printing is a press-and-go affair, like using a photocopier. In practice, a lot of care and patience is required, along with some failed prints. Our first set of MiniKame legs didn't have enough support material and were somewhat droopy, so we increased the support and tried again. The result was a perfect print, but the extra support proved very difficult to remove. Thus our feet have a messy remainder of chiselled-off support material on their underside. It's best to own up to these things.

Always read the instructions. We fitted the servo arms, then had to remove them again for the calibration step.

The servo arms can break if too much force is applied. We broke a couple, and had to cut one side off a double-sided arm to make a replacement.

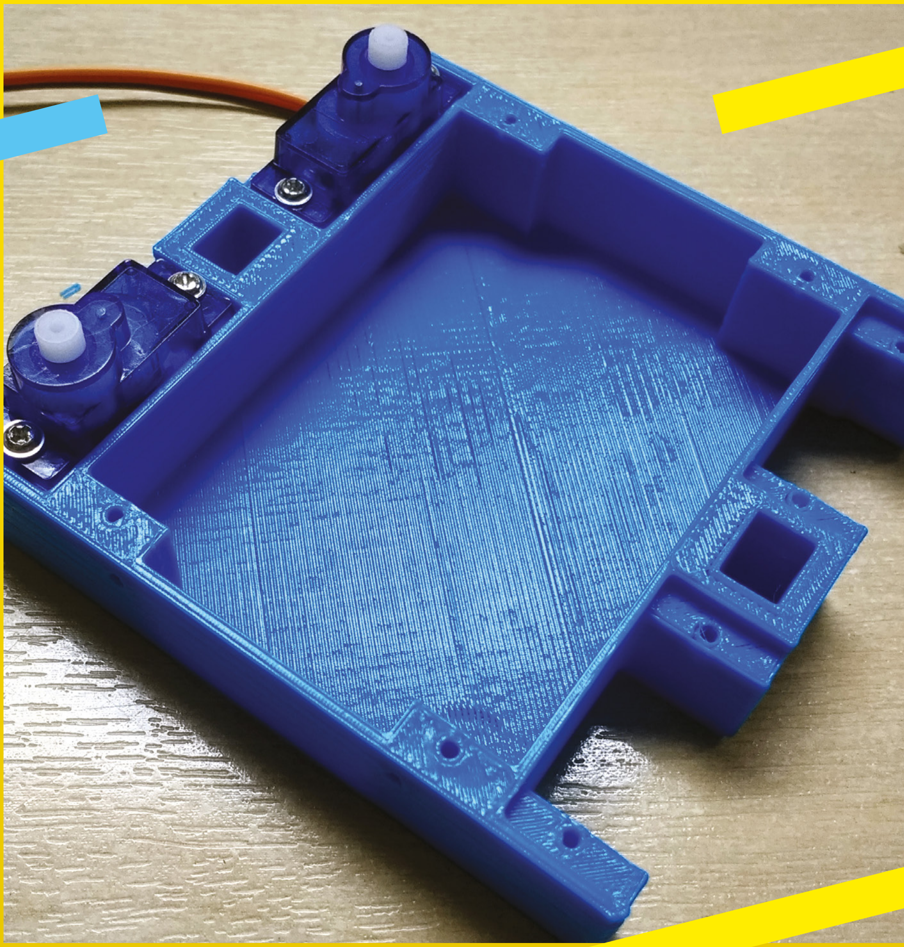
Our Bluetooth module was extremely difficult to pair with our phone. It would appear in the list of devices, then disappear as if by magic when we tried to connect. A lot of time was expended getting it to eventually connect.

We have to admit it, our MiniKame has been temperamental. All the components appear to be fully working, yet sometimes they refuse to work together. A thought was that a power supply might be to blame, but they continue to have problems even when a bench power supply is used.

If there is a lesson to be learned here, it is to always buy good-quality components if you want a robot that walks all of the time rather than some of the time, and be prepared to mistrust a cheap Chinese Arduino clone that has been sitting on your bench for a year or so.

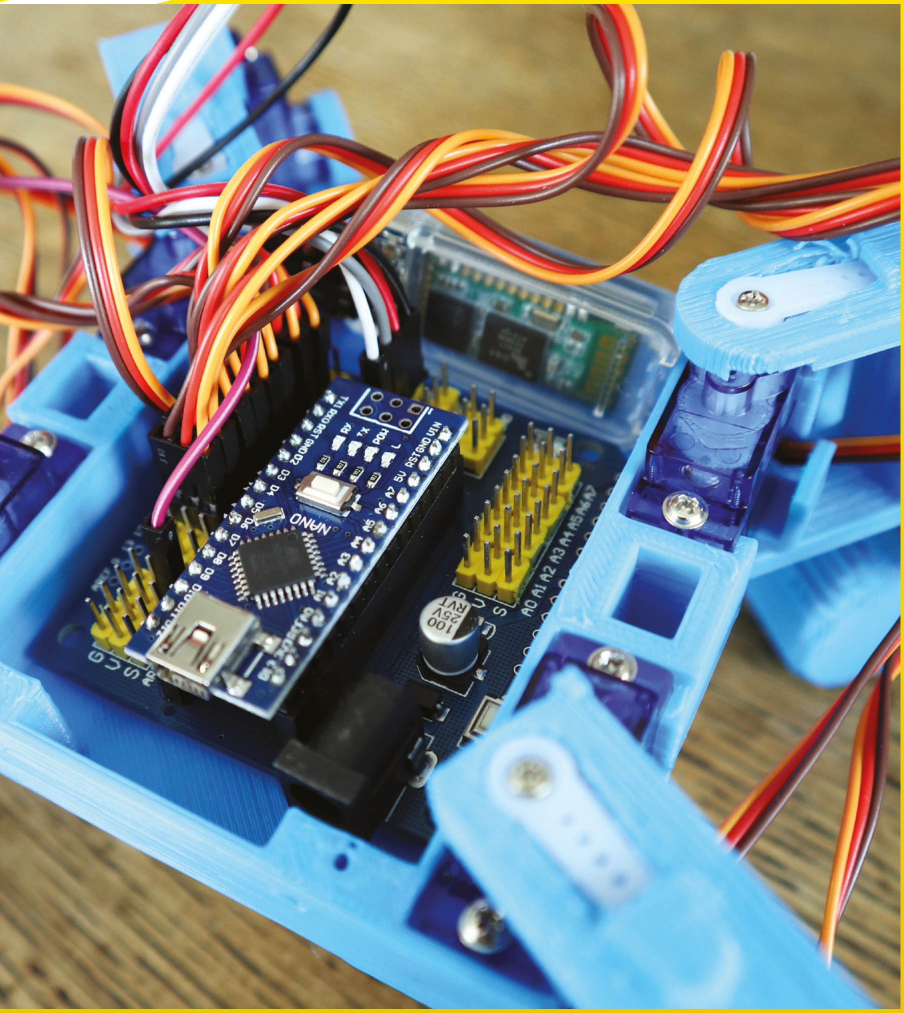
Once the pin and servo are fitted to the thigh piece, the lower leg can be fitted. Slide the pin into its knee joint hole, and then gently ease the other side of the knee over the servo shaft. Don't fit the servo arms yet – we need the servos to be able to move freely for the calibration step which we'll perform later.

The body is little more than a plastic box with mountings for four servos that form the hip joints, which should be easy enough to fit and screw into place. Match up the legs to the body corners, such that each corner has a leg whose knee servo shaft faces outwards. With the link bar held underneath across the robot, such that its pin is in the same axis as the servo shaft, fit the leg in the same way as the knee joint by locating the lower point of the hinge on the pin and easing the upper point onto the servo shaft. Again, don't yet fit the servo arms. Fit all four legs, and route all the servo wires to come together in the centre of the body. That's it: you now have recognisably built a MiniKame robot, albeit one with floppy legs, because the servo arms are not yet installed. →



Below
The thigh servos mounted on the body

MAKING IT MOVE



Left

The Nano and its shield fit neatly into the body. The Bluetooth module is in the background

The next hardware task is to fit the control board and hook up the servos. We used an Arduino Nano clone and an expansion board with an added Bluetooth-to-serial board because that's the most basic MiniKame configuration, but there are versions of suitable software for multiple other controllers. Wiring is straightforward, with the Nano expansion board providing numbered headers for servos that should be connected as follows:

- D2 to front right hip servo
- D3 to front right knee servo
- D4 to back right hip servo
- D5 to back right knee servo
- D6 to back left hip servo
- D7 to back left knee servo
- D8 to front left hip servo
- D9 to front left knee servo

The Bluetooth module is then wired in with four socket-to-socket jumpers: two to 5V and GND pins, and the serial TX on the Nano board to the RX on the module, and the RX to the TX.

There's one further component that we've not yet mentioned: the power supply. There are a huge number of possible ways to power any Arduino project, almost all of which could give a MiniKame the necessary juice. We tried two methods: our robot was able to function from a mobile phone booster pack over a USB cable, or from a pack of AA batteries via the Arduino power jack.

Other options you could consider might be a LiPo battery with a suitable regulator board, or even a long cable from an external power supply. It's worth bearing in mind, though, that the robot may not like too much weight; our MiniKame found the weight of eight AA batteries to be a bit much.

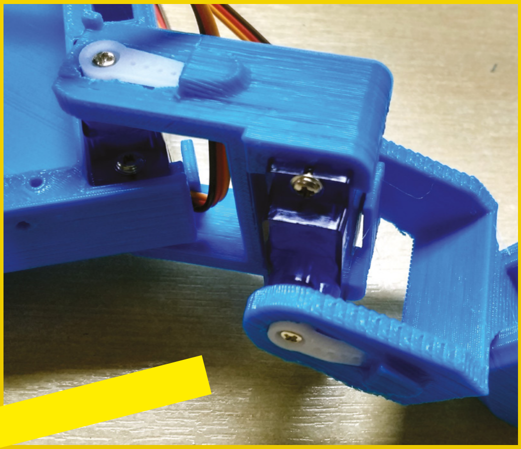
SOFTWARE

We are almost ready to install the software on our MiniKame, but there is one final step before we can proceed. It involves another piece of software, which sets up the Bluetooth module. This can be found on the Instructables page, but is also shown below. You'll need to use the Arduino IDE to load it onto your Arduino Nano and, ensuring that the Bluetooth module is connected, you should then reset the Arduino without USB connected and let it run.

```
void setup() { Serial.begin(9600); //change to fit
your ble initial baud_rate
Serial.println(“AT+UUID0xDFB0\r”); // uuid
delay(50); Serial.println(“AT+CHAR0xDFB1\r”);
// characteristic delay(50); Serial.
println(“AT+BAUD8\r”); // set baud rate to 115200
}void loop() {}
```

The stock software for a MiniKame comes in two halves: an Arduino sketch for the robot itself, and an app for your phone. The result is a simple remote-controlled robot which is fun enough to play with, but the real fun comes in the accessible nature of Arduino coding. You can choose to use it as a novelty robot toy, or you can get inside its mind and hack the software.

Installing the final Arduino sketch is as simple as downloading its repository from GitHub, unzipping the archive, and compiling it to the Arduino itself using the Arduino IDE. Meanwhile, there is more than one suitable app for a Kame in the Play Store and the Apple App Store, which can be installed on your device of choice. Pair with the Bluetooth module and you should be ready to proceed.



You should now have a MiniKame robot with floppy legs

If all went well, you should now have a MiniKame robot with floppy legs, but with all software and wiring in place. The final step is one of calibration, the act of setting all the servos to a known position before fitting the servo arms. This is simple enough: fit a jumper wire between the Arduino Nano's D12 line and its 3.3-volt pin (made easy by the relevant pins being exposed on our Nano expansion board), and power up the robot. You will hear the servos move into their calibrated positions, then you can power down the robot, remove the jumper from D12, and install the servo arms.

Position the robot flat with its legs outstretched at 45 degrees to the body, and carefully snap in a single-sided servo arm into each inset. There should then be just enough room to slide a single-sided servo arm into the inset, and click it onto the servo shaft before screwing it into place. If you're lucky, you should now have a working MiniKame. Enjoy it!

WALKING MECHANICS

To successfully make a robot walk, we have to understand something of the mechanics of walking, both the structure of a leg and the coordinated movements of its joints. One of the best sources for this comes from an unexpected source: not zoologists or roboticists, but animators. A cartoon character walking across the screen has to look like a cat, a dog, or an anthropomorphic standing-up mouse, and one of the key features it must possess to do this comes in walking in a convincing way. Thus animators make an extensive study of walking motion, so if you're interested in walking, a very good place to start is a web search on phrases such as 'animation walking tutorial'. You may not need to produce the natural motion that the animators require, but a grasp of the sequence of movement of a four-legged animal's legs is important in understanding of how your robot can move without becoming unstable.

Copyright, 1878, by MUYBRIDGE.

MORSE'S Gallery, 417 Montgomery St., San Francisco.

THE HORSE IN MOTION.

Illustrated by MUYBRIDGE.

AUTOMATIC ELECTRO-PHOTOGRAPH.

"SALLIE GARDNER," owned by LELAND STANFORD; running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed in each twenty-seven inches of progress during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The exposure of each negative was less than the two-thousandth part of a second.

Above The Victorian photographer Eadweard Muybridge was one of the first people to study walking motion in detail

Below The servo arms fit into the insets in the 3D-printed parts

MORE ROBOTS

You've built your first robot – what next?

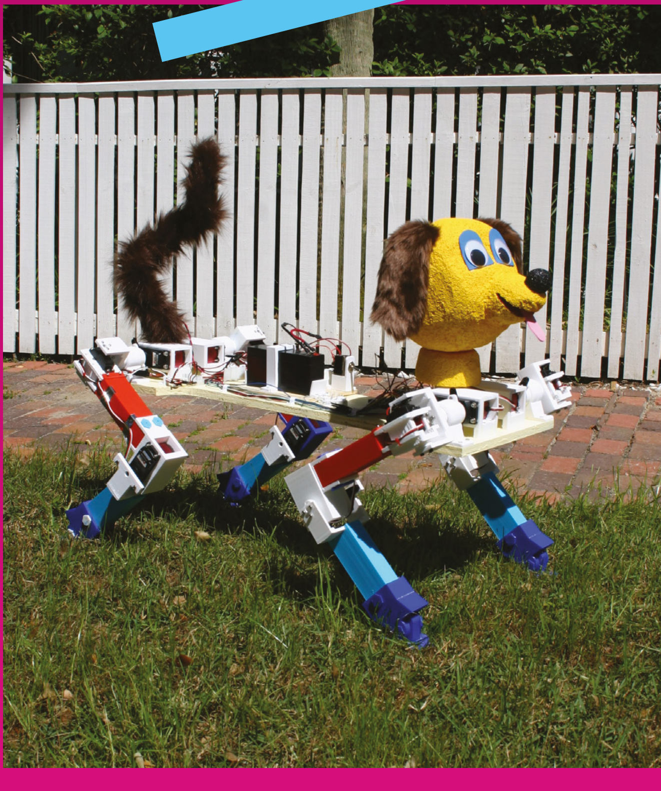
MIKE'S WALKING DOG

Created by Mike Rigsby, this cheap and cheerful hound can be walking around your estate for under £500:

"Walking robot platforms can navigate buildings, climb stairs, enter cars, and traverse farmland. Potentially, they can become elder companions or herbicide-free weed removers. Excessive cost for such a platform – tens of thousands to millions of dollars – discourages students, makers, and startups from advancing the technology. My open, shared walking platform can be constructed using parts and materials that cost less than $600.

"The dog has evolved from a gangly beast that could barely stand to something that can now barely walk. The legs have been shortened, and joints strengthened. The servo motors selected represent the highest torque per dollar that I could find.

"I am a writer and a maker – the best place to keep up with the dog's progress (as well as files and build instructions) is **hsmag.cc/sBBErE**. The best video of the dog moving can be found at **youtu.be/kclfsCcEjcs**.



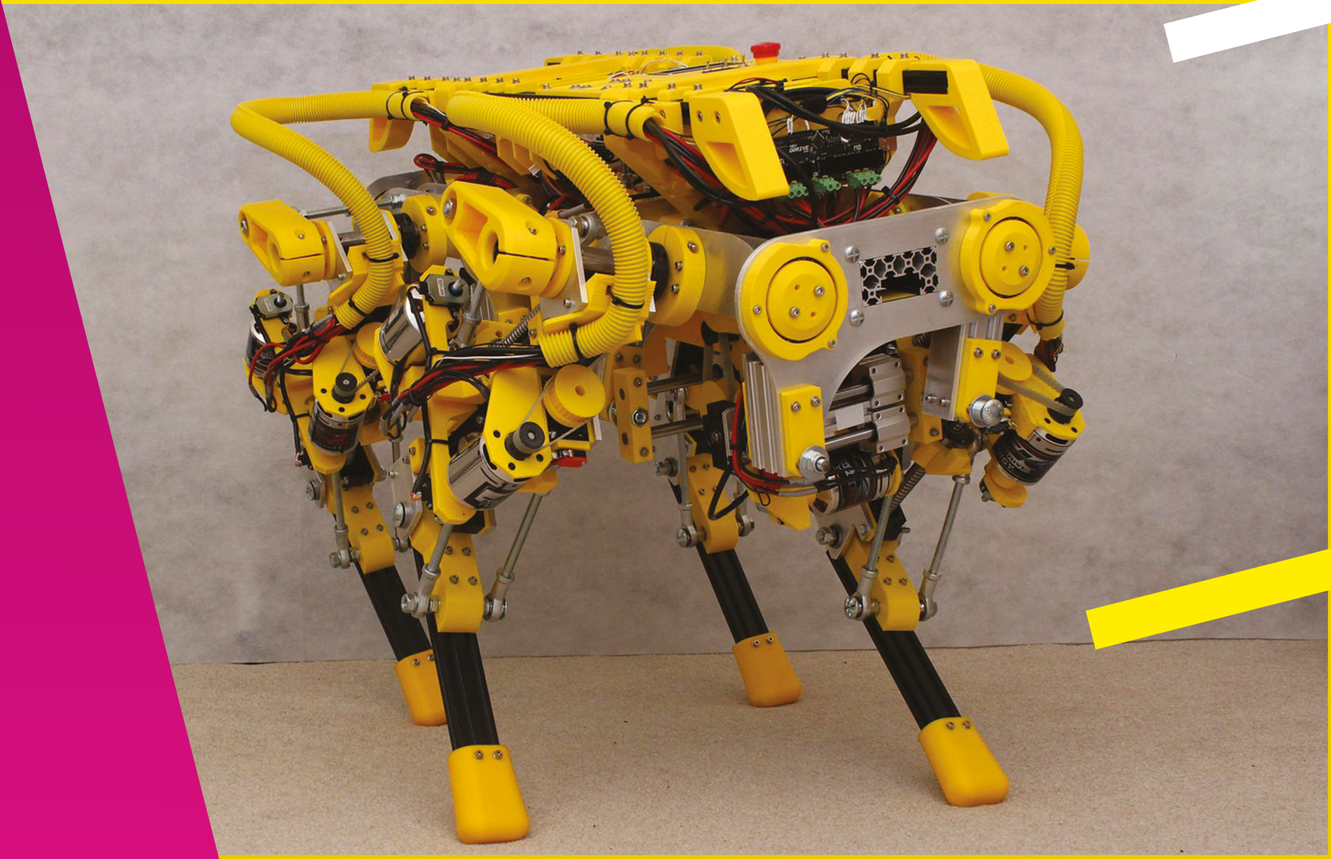
Right ➡
You don't have to add a head and tail to your dog, but you can

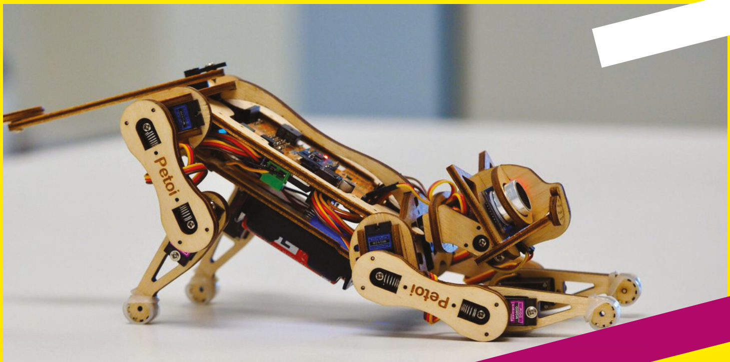
OPEN DOG

The pinnacle of walking robots has to be the work coming out of Boston Dynamics, as we've already mentioned. But what if you don't have millions of dollars to spend on research and development? James Bruton is in the process of building an open-source, four-legged walking robot called openDog, and documenting the process on his YouTube channel so that anyone can follow in his footsteps. So far, it's cost him just over £2000 – you can see for yourself what the robot can do at **youtube.com/user/jamesbruton**

Below
Is this an open-source dog or a terrifying vision of the future?

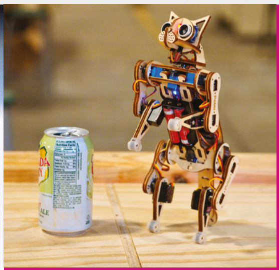
Credit
© James Bruton





NYBBLE

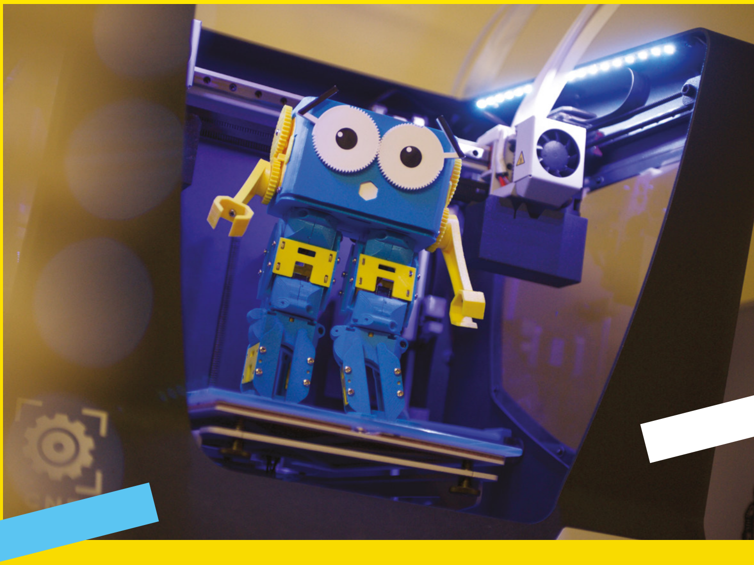
If you're not a dog person and would rather have a robot cat, Nybble is for you. Its body is made of laser-cut wood, so it's easy and cheap to put together. It uses an Arduino-compatible microcontroller with the option of plugging in a Raspberry Pi to make it more intelligent, and can take inputs from built-in ultrasound, lidar, GPS, and more sensors.

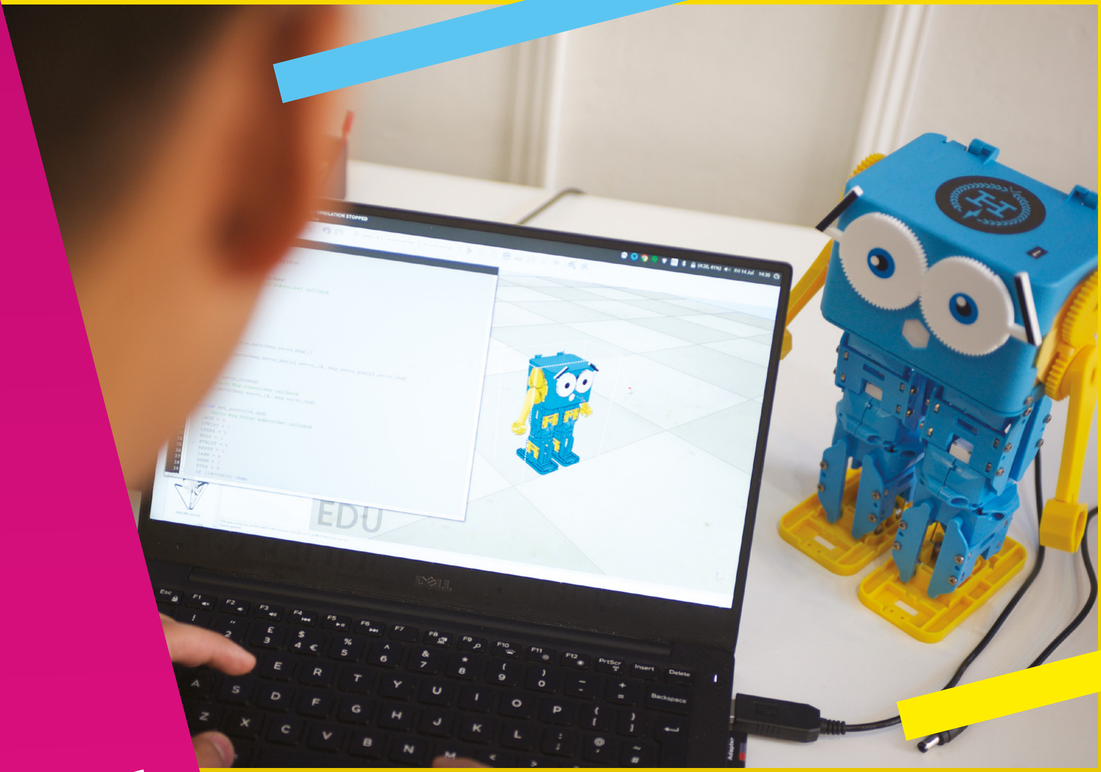


Left ⬘
Just like a real cat, Nybble understands voice commands; it just chooses to ignore them

MARTY THE ROBOT

If you're more comfortable following a script than going off-piste with your own design, give this a try. Marty is a two-legged walking robot which solves the problem of balance with an exaggerated hip sway and big, stable feet. Where it really comes into its own is the accessibility of its programming. It's controllable via Python, JavaScript, and even Scratch, so it's ideal for kids wanting to take their first step [pun intended] into ambulatory androids. □





Above An easy way to wobble along

How I Made A VERTICAL PLOTTER

Building my own drawing machine

By **John Proudlock**

A few years ago, I ended up with a spare Raspberry Pi, and I set about looking for a project to put it to good use. I had a bit of a root around on the internet for inspiration. There was no shortage of ideas, but I wanted something that involved movement – I'd never used the GPIO ports on a Pi. I also wanted something that needed a bit of simple coding – I'd not coded for years, and I wanted to make something that would have some vaguely artistic output.

And that's why plotters caught my eye. I've never been very good at drawing. I always wondered if I could build a machine to help me… It turns out I could. The basic principles are fairly straightforward, and this article explains them.

You'll probably have seen commercial plotters. Typically, the paper is set horizontally, and an armature moves a pen over the surface to draw. There's a much simpler version, known as a vertical plotter (depending on the creator, they might also be called v-plotter, polar plotter, or polargraph). Search for images online, and you'll quickly see how these devices work – but I've marked up the photo of the one I built opposite, and its main parts:

A. Paper attached vertically to a wall (on the plotter easel)

B. At the top-left and right corners, there are two motors; the drive shafts point out into the room

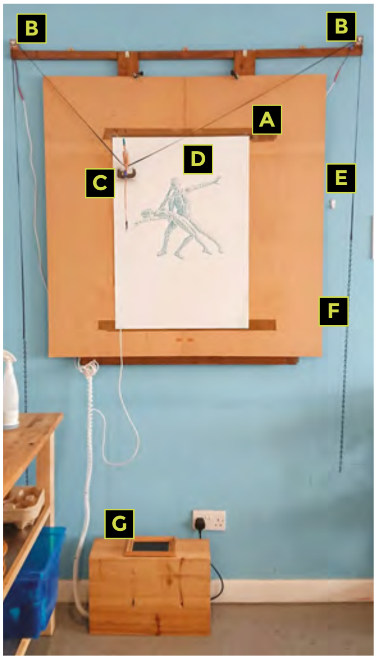
C. Pen holder, suspended on a toothed belt running between the two motors

D. When the motors move, the pen is dragged across the surface of the paper, creating an image

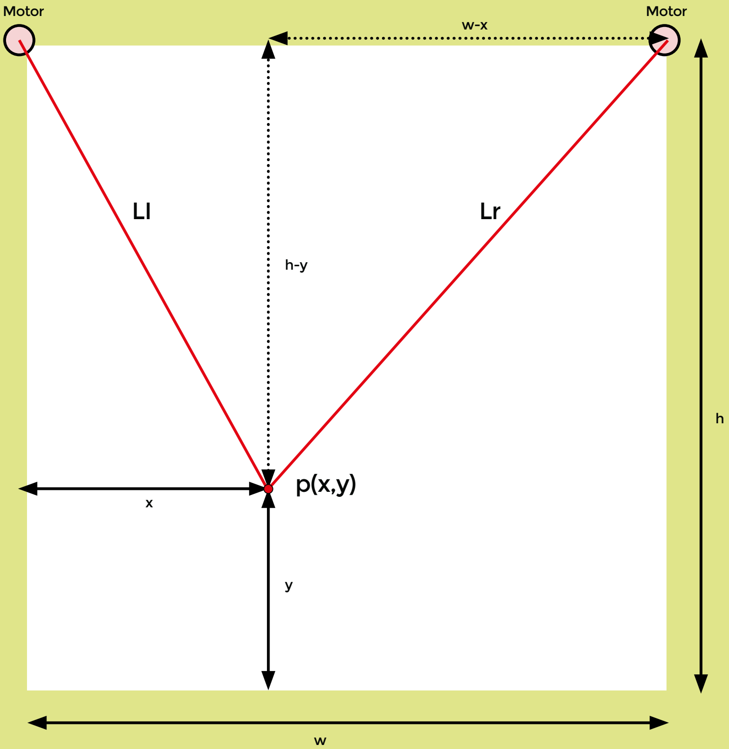
E. Toothed belt moving over the motor sprockets

F. Weights (steel chain) to counter the weight of the pen holder

G. Control electronics with small output display



Above The plotter today



Left Understanding the maths

The control electronics:

A. The power supply unit (from an old desktop PC)

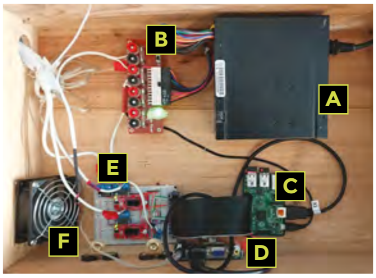
B. PSU breakout board

C. Raspberry Pi B+ and GPIO breakout

D. 7-inch HDMI screen (the back of)

E. Breadboard
- Stepper motor drivers to left and right motors
- Drive for the pen lift servo

F. Cooling fan for motor driver circuits (normally mounted over the breadboard)



Left The control electronics

V-PLOTTER MATHEMATICS

I start new projects by working on what I think will be the hardest part of the project for me to complete. That way, if the project is beyond me, I find that out early on and abandon it! For the plotter project, I wanted to make sure I could understand the underlying mathematics. I sketched out a simple model of a plotter.

I wanted to work in Cartesian coordinates, so if I want the pen at point p(x,y) how long would the belts from the left motor (Ll) and the right motor (Lr) need to be? To know that, I also need to know the height of the paper (h) and the width between the two motor spindles (w).

The v-plotter makes two right-angled triangles, and the values can all be calculated using Pythagoras' theorem:

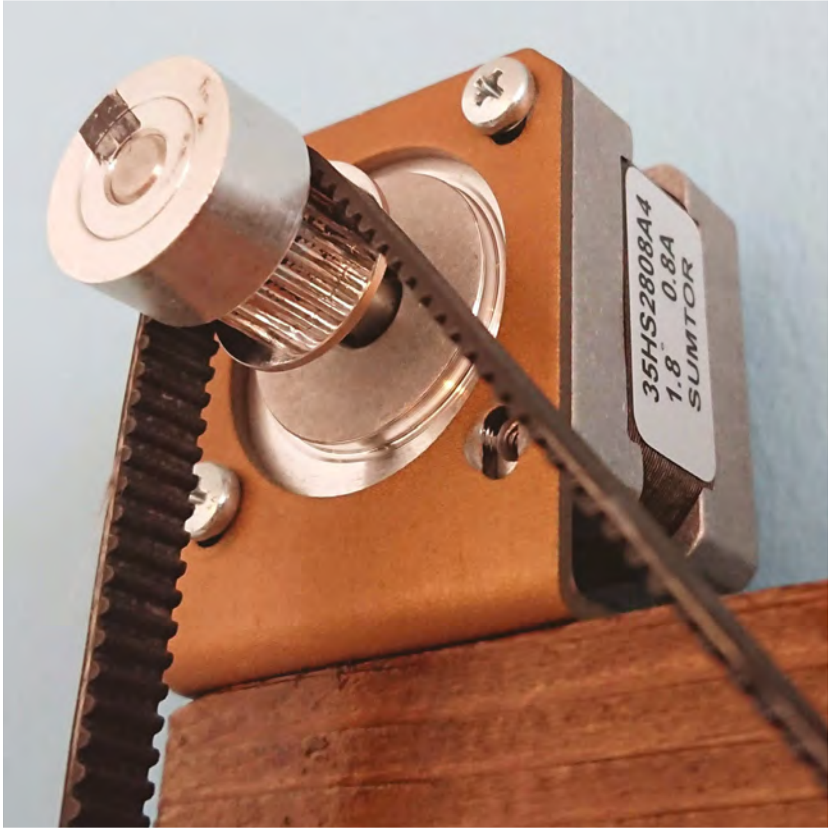
$$Ll = \sqrt{(x^2 + (h-y)^2)}$$
$$Lr = \sqrt{((w-x)^2 + (h-y)^2)}$$

So, if the plotter is 500 mm square, h=500, w=500. And, if I want the pen to be at p(150,200), then x=150 and y=200 so the belt lengths can be calculated as:

$$Ll = \sqrt{(150^2 + 300^2)} = 335.4$$
$$Lr = \sqrt{(350^2 + 300^2)} = 461.0$$

I then started to think about moving the pen across the paper to draw a line. I bought a pair of NEMA stepper motors which are driven by electrical pulses (that would come from the Pi's GPIO). For each pulse, the motor would turn 1.8 degrees, so 360/1.8 = 200 steps per full rotation of the motor shaft.

I then dipped into the world of 3D printers to get a GT2 toothed pulley for the motor shaft, and decided I'd use a GT2 toothed belt to run over each pulley and suspend the pen. This had the advantage that the pulley wouldn't slip or stretch (unlike string). The sprockets I found →



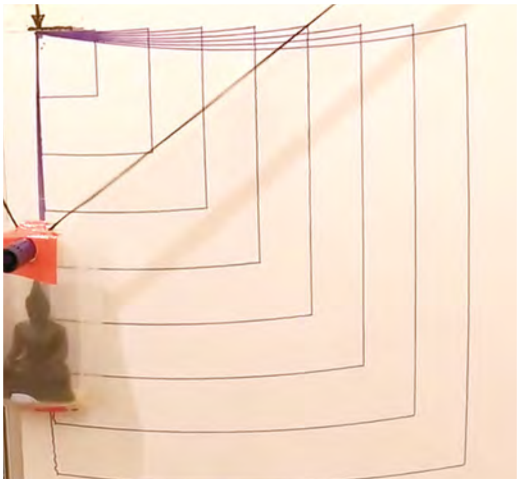


Figure 1
I never did work out the maths to correct distortion in long line

Left
Using a GT5 sprocket to precisely control the plotter

had a circumference of 40 mm. This means one rotation of the motor would take 200 pulses, and would move the pen 40 mm closer/further from the motor. You can see the left-hand motor setup in the image above.

In the example before, if the pen is at point A(150,200) we know Ll = 335.4 mm and Lr = 461.0 mm.

If I want to move the pen to Point B (350,450) then we need to make Ll = 353.6 mm and Lr = 158.1 mm.

Just considering the right-hand motor and Lr, the change in Lr is 461 – 158.1 = 302.9 mm. This requires 302.9/40 = 7.57 rotations of the motor spindle.

As a single rotation requires 200 pulses, we can calculate that the right-hand motor must receive 200 × 7.57 = 1514 pulses.

You can use the same working to calculate the pulses needed to set the new length of the left-hand belt, Ll (the answer is 91 pulses).

> By using short wires, close to the board, the breadboard solution has been robust enough to work reliably

HARDWARE – SETTING UP THE MOTORS

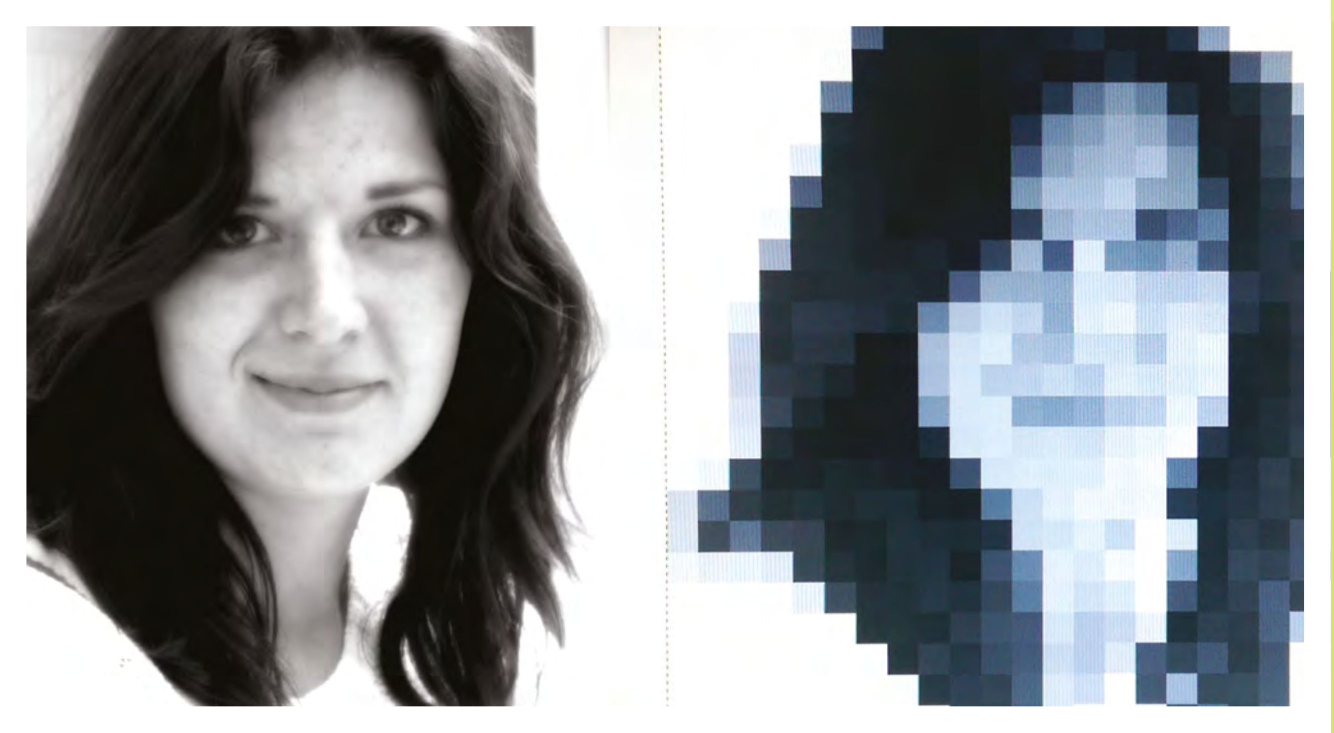
Confident that the underlying mathematics was all manageable, I moved onto the hardware and getting the two motors powered up. The Pi GPIO would provide the signal (the pulses mentioned earlier) as a square wave, but the Pi can't provide the current needed to drive the motors, so you need some additional circuitry to boost the amps.

Firstly, I connected the GPIO from the Pi to a breadboard so that the signals from individual pins were accessible. It was also easier to get pre-built stepper motor driver circuits to save time, and I found the (excellent) 'EasyDriver – Stepper Motor Driver' circuits. I soldered on the headers so the two driver circuits would plug into the breadboard.

I connected the four power lines to each stepper motor and powered up the driver circuits. The motor spindle went from being loose to fixed firm (stepper motors draw current and lock without an input signal).

I always intended to transfer the breadboard circuit onto stripboard, but never got around to it. By using short wires, close to the board, the breadboard solution has been robust enough to work reliably.

There were two inputs required to make the motor move: the 'drive' signal, which would be a square wave where each pulse would move the motor spindle 1.8 degrees, and a 'direction' signal as the motors would need to rotate back and forth depending on the direction of pen line required. Armed with this knowledge, it was time to write the first code for the project.



Above Portraits, starting with the basics

SOFTWARE – DRAWING A LINE

I used Python and worked in the IDLE environment to simply set a GPIO pin high, then low, 50 times a second. I did this for four seconds, and I got a motor spindle to turn one rotation clockwise. I then set the 'direction' pin to high, and repeated the four-second cycle. The spindle turned one rotation anti-clockwise.

From there, I duplicated the code for the other motor and clamped each motor to a shelf: one on the left, one on the right. Under the shelf, I propped up a whiteboard. I took the toothed GT2 belt and draped it across the two motors. I gaffer-taped an old takeaway carton in the middle of the belt and glued a dry marker to it. With the left motor turning anti-clockwise, and the right motor turning clockwise, the pen was dragged up the board and made a neat vertical line.

I wasn't going to meet any artistic needs with a single vertical line, though, so I wrote a function that would be passed an (x,y) location and would move the pen from its current location to the one I requested.

The main steps to this are:

- Knowing your current pen position
- Knowing the position you want to move to
- Calculating the change in the left belt, and the right belt
- Calculating how many steps are required for each motor to make the change in belt lengths
- Moving the motors

I had the maths for this from earlier in the project, so coded it up. The first stumble was that if you move one motor, then the other, you don't get a straight line to the destination, you get a sort of L shape. So you have to use two processor threads, one for each motor. Both motors have to start moving at the same instant.

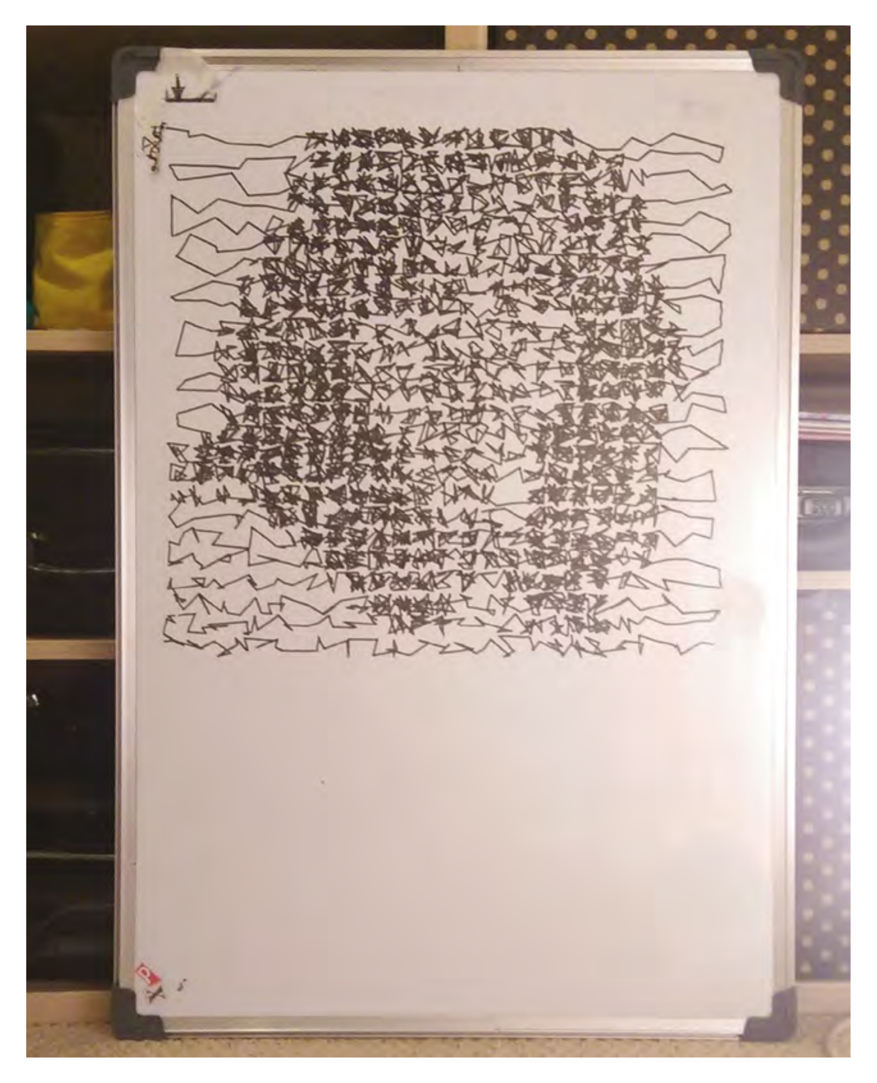
I found that I needed a more sophisticated development environment than IDLE and moved the project into PyCharm. The PyCharm debugging and refactoring tools were essential as the code got more sophisticated.

Unfortunately, that still left me with a kinked line because (unless it's a vertical line in the centre of the paper) the L belt change and the R belt change are different lengths, so one motor finishes first and you end up, again, with a curved L shape instead of a straight line.

To fix this, the motors have to run for the same period of time, starting together and →

Version control

As the code became more complex, I learnt to use the very basics of Git, a code repository tool. This made it easy to go back to known 'safe' versions of the software if I needed to. I went on to use Bitbucket (an online repository). This let me work on the software on a desktop PC, then 'push' the code up to Bitbucket. On the Pi (controlling the plotter), I'd simply 'pull' down the latest version of code that I wanted to use.



Right Portraits, looking basic

finishing together. To do this, you need to have set your fastest motor speed. I found 50 pulses per second worked, and used that as the 'maximum motor speed':

- Identify which belt length change is the biggest. In the example from earlier, this is the change to Lr, which requires 1514.32 pulses. At 50 pulses per second, this will take 30.29 seconds
- The left motor must change the Ll length by just 91 pulses. These must be delivered through the 32.29 seconds that the right-hand motor is running, which requires a pulse rate of 91/32.29 = 3 pulses per second

I coded this so that both motors ran for equal periods and… it worked, mostly. Short lines of about 5 mm appeared straight. Long horizontal lines would curve slightly, like they were sagging. You can see this in **Figure 1** from early in the plotter's development. The plotter is drawing squares of increasing size. As the horizontal lines get longer, you can see them dip in the middle. There was distortion on the vertical lines too, but it was not as noticeable.

The cause is that in my implementation, the pulse rate sent to the motors is constant, and each motor rotates at a fixed rpm as the pen moves. Instead, the motor rpm should vary constantly during pen motion. I haven't attempted to understand the mathematics required to fix this (I'm not sure I could), so instead, I found a workaround. Any long line is broken into segments, each no longer than 5 mm. In this way, a long line is drawn in several short segments. Although each segment bows slightly, it's not visible and the line appears straight.

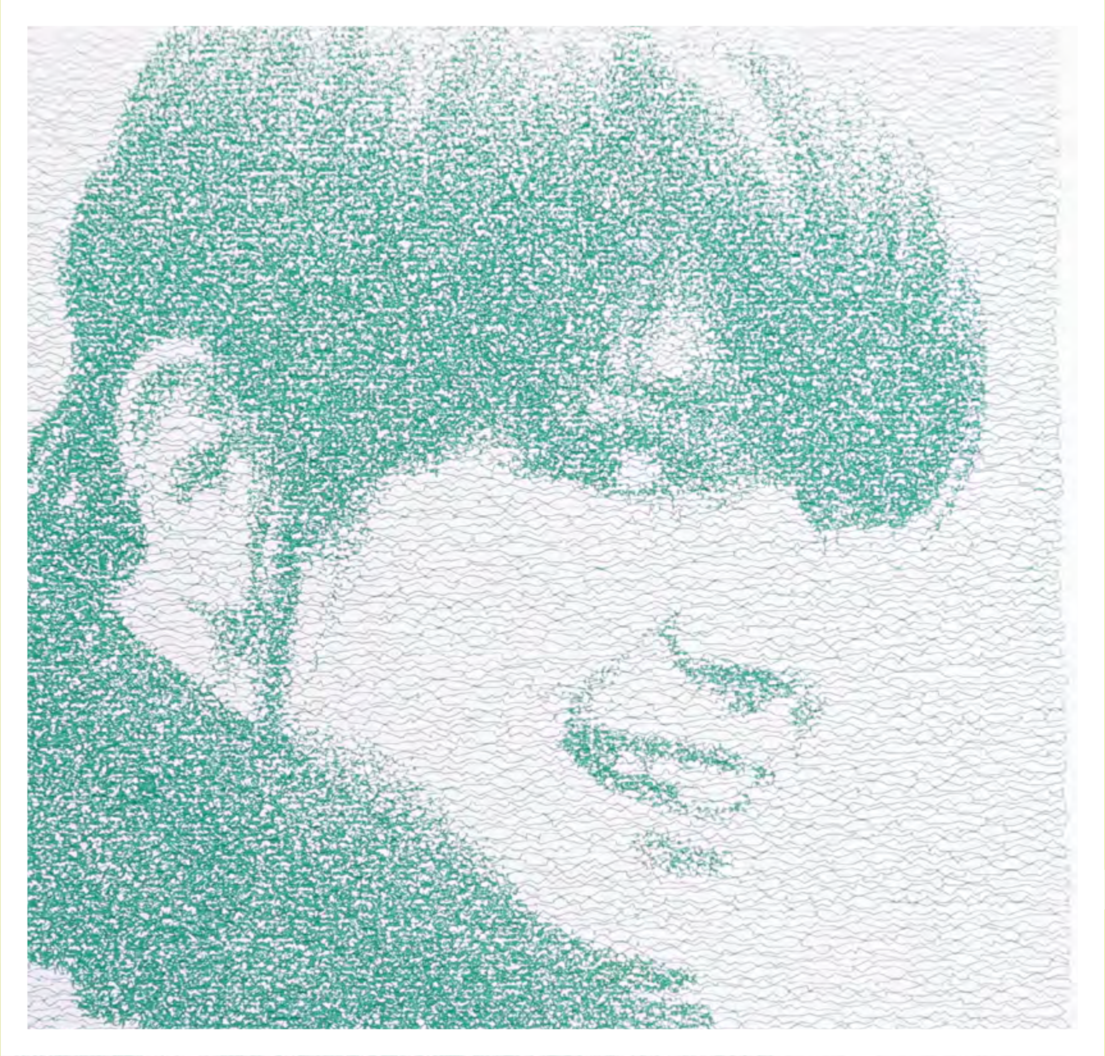
The plotter is drawing squares of increasing size

In this prototype version of the plotter, you can see that I'm plotting on a whiteboard and that the marker pen is held in an old takeaway container. The container holds my trusty Buddha paperweight to stabilise the pen.

I really wanted to draw something a little more inspiring than squares and started to experiment with drawing bitmap images, all with a single line. I started by considering a photograph I had and simplifying it into a small bitmap of 23×23 pixels.

I then mapped these pixel values to a 'darkness level' by equating black to a darkness level of 100 and white to level 0. In the plotter control code, I stored these values in a two-dimensional array. The plotter was then configured to represent each pixel in a 20 mm square and started with the pen in the top-left corner of the image. The latter is a white pixel (level 0), so the pen moves to a random point in the next adjacent pixel. If the next pixel is black (represented by 100), the pen must keep moving to random points in the pixel's square until 100 mm of line has been drawn. After that, it moves to the next adjacent square.

This simple algorithm would 'colour in' a 20 mm square with an amount of line represented by a pixel's darkness level. In this way, graduating tones would be



Left Portraits, a 100×100 pixel bitmap, drawn with a single, unbroken line

represented too. A mid-grey (value 50) would cause 50 mm of line to be drawn in a pixel's square.

Once coded up, I set the plotter running and waited to see how close the plot was to the image I had passed in.

With some squinting, I could just about make out the original image but could see that to make an engaging image, I would need a much higher resolution. This would need a bigger drawing space and a smaller pen tip. In any case, the shambolic 'tape a couple of motors onto a shelf' approach was not proving particularly reliable. This triggered the building of a simple easel that I could hang on the wall. In turn, I also upgraded the drawing algorithm to work in 5 mm squares per pixel, and this allowed the re-creation of 100-pixel images, and much better image creation.

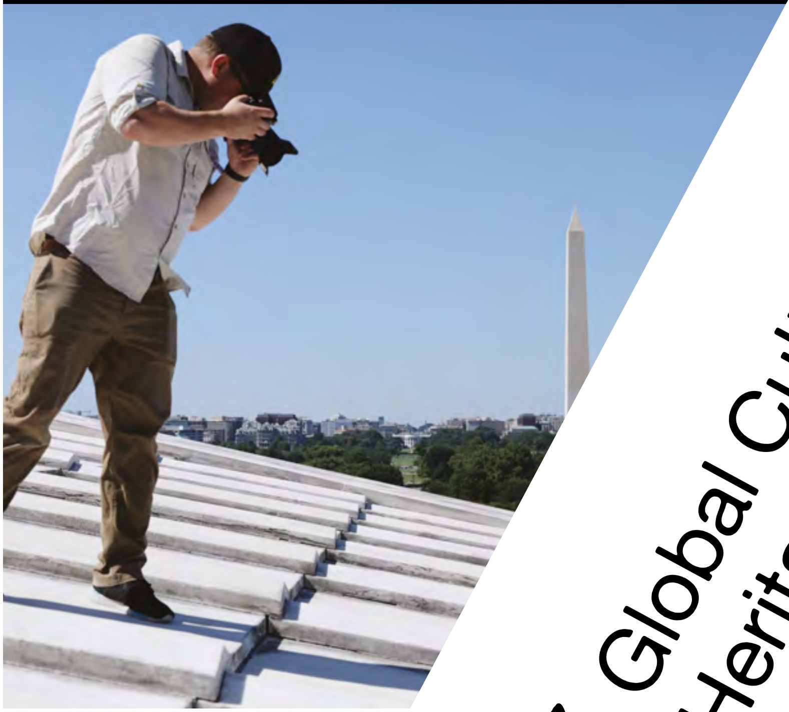
WHAT NEXT…?

This is a 'weekend project' that's now been running for about four years, and some extensions I've found fun to add to the project are:

- Plotting bitmap images with a single line
- 'Generative Art', which basically means plotting the results of equations. It may sound dull, but the results can be stunning:
  - Plotting spirographs (the equations are scary-looking, but not that hard to replicate in code – even if you don't understand them)
  - Plotting fractals
  - Plotting modulo arithmetic
- Adding a mechanism to lift the pen off the paper
- Plotting G-code, an industry standard file format for CNC machines
- Drawing bitmap files with multiple colours
- Writing an emulator so you can predict how the plot will look on screen before you waste another set of pens and a sheet of paper

Each of these hacky endeavours would be an article in itself, but you can see the results from this plotter project at **instagram.com/inkylinesplots** and **inkylines.blogspot.com**. □

NEW CHRISTMAS LACE
BELIEVE IN YOURSELF



CyArk

Global Cultural Heritage Preservation

Discover the tools being used to preserve and explore the **world's cultural heritage**

Cameron Norris

@CameronSNorris

Cameron is a technology and communications specialist, passionate about the use of open-source hardware for social innovation.

As flames ravaged the 850-year-old Notre-Dame Cathedral just days before Easter Sunday, there was little the world could do besides watch in grim horror. "Disasters such as this have a huge impact, not only on those who work and worship in a church, but also the wider community for whom the building is a symbol of identity," explains Dr Kate Giles, a Senior Lecturer from the University of York's Department of Archaeology.

The overnight destruction of one of France's most iconic landmarks – a building that took nearly 200 years to construct – provided another harrowing reminder of the fragility of cultural heritage sites around the world.

"Our heritage is much more than our collective memory. It's our collective treasure. We owe it to our children, our grandchildren, and the generations we will never meet to keep it safe and to pass it along," says Ben Kacyra, an Iraqi American civil engineer who founded CyArk, a non-profit organisation dedicated to digitally recording, archiving, and sharing the world's most important cultural heritage.

Since officially launching in 2003, CyArk has documented over 200 cultural heritage sites on all seven continents, including the Sydney Opera House and Skara Brae, a 5000-year-old stone-built Neolithic settlement in Scotland. Ben and his wife Barbara set

Image credits CyArk & Partners

up CyArk after witnessing the destruction of the magnificent 5th-century Bamiyan Buddhas in Afghanistan. "They were gone in an instant," says Ben. These ancient sandstone carvings, once the tallest statues of Buddha in the world, were destroyed with dynamite by the Taliban in 2001. "Both my wife and I were so touched by this that we decided to expand the mission of our foundation to include digital heritage preservation of world sites. We called the project CyArk, which stands for Cyber Archive."

PRESERVING OUR SHARED CULTURAL HERITAGE

Using high-resolution digital imagery and laser scan data, CyArk began working methodically to quickly produce accurate 3D models of cultural heritage sites that could serve as a record to aid recovery efforts in the case of natural disaster, vandalism, war, or simple neglect.

Cultural heritage can be defined as monuments, buildings, or landscapes of outstanding universal value to history, art, or science. "In the beginning, CyArk worked closely with the 3D scanning community to collect datasets around the world," says CyArk CEO and Chairman, John Ristevski. "Later we began to incorporate photogrammetry into the process using high-resolution cameras and drones to capture the colour and texture of surfaces alongside the geometry captured by the laser scanners."

Last year on World Heritage Day, CyArk and Google Arts and Culture launched the Open Heritage initiative to provide open access to 3D cultural heritage datasets for education and research. Twenty-six high-resolution digital datasets of heritage sites were released, including Mexico's famous Chichén Itzá, one of the Seven Wonders of the Ancient World, and the iconic moai statues carved from volcanic stone on Easter Island.

Above ↑
CyArk took to the air with the help of the US Park Police and their helicopter to capture the roof of the Jefferson Memorial

This year, CyArk is building on the initiative even further with the Open Heritage Alliance, a collaboration with Historic Environment Scotland and the Digital Heritage and Humanities Center in the University of South Florida Libraries, to release 36 additional datasets through Open Heritage 3D, a dedicated portal for sharing 3D cultural heritage data. The collection represents heritage from four continents and twelve different countries, spanning over 5000 years of human history.

DIGITISING ICONIC MONUMENTS

"Our goal isn't just to digitally preserve heritage sites at risk, but to make their stories and the data we collected available to future generations of researchers, educators, and students," explains John. From an ancient city carved out of a mountain top at Mexico's Monte Albán, to isolated expedition huts that sheltered Shackleton and his teams in their Antarctic explorations, these datasets can be downloaded by anyone, and the stories they tell touch on the diversity of the human experience as well as our shared histories. "Since we began this project last year, the data has been downloaded thousands of times, used to create 3D-printed temples from Thailand, and incorporated into VR experiences that let you explore an ancient city," he adds.

To access the data, provide your name and email address after hitting the download option on the relevant project page, and you will be emailed the links to download the datasets. However, be warned, the average download size per site is approximately 25GB. "There is incredible power in sharing this information with a broad community, and we can't wait to see what they do with it." →

LASER SCAN

In 2015, art historian Andrew Tallon created a 3D reconstruction of the Notre-Dame Cathedral using more than one billion laser-measured points of data. Andrew, who sadly passed away last year aged just 49, described himself as "obsessed" with the cathedral and hoped to reveal secrets about the building's unique architectural history through his research. Many believe his laser scan data will prove invaluable in guiding the upcoming restoration process.

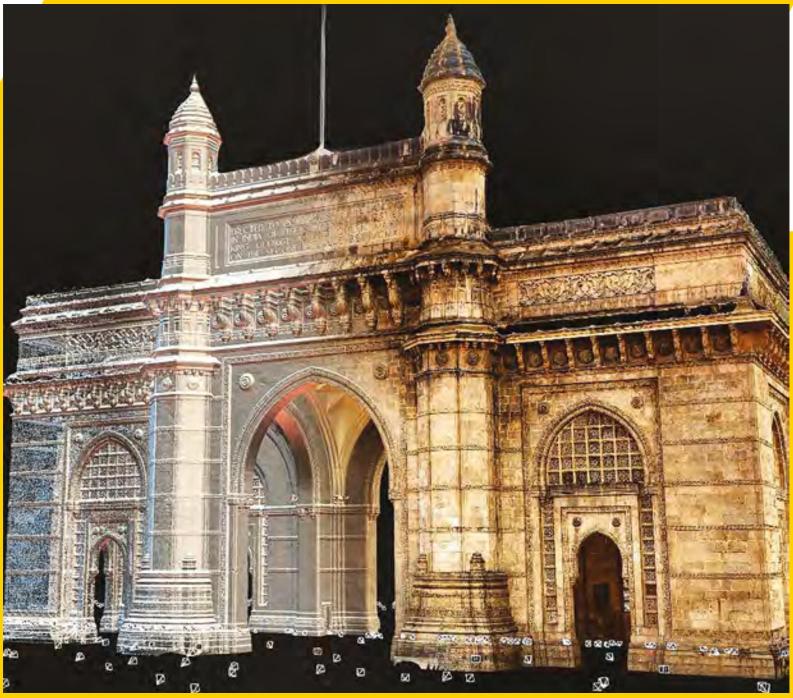
BACKUPS

CyArk keeps an archive of all raw and processed data to ensure a record is available for the future. Data is backed up in the field before travelling back to the office, where it is transferred to the local server and written to a tape drive for an off-site backup. At the end of a project, LiDAR scans and photogrammetric images are converted to non-proprietary and open exchange formats. The complete dataset is then sent to a secure underground storage facility in Pennsylvania.

What is photogrammetry?

Photogrammetry is the science of measuring physical objects and the environment using photographs. By taking overlapping photographs of the same object, but from slightly varying angles, a 3D representation of that object can be obtained using a method known as 'triangulation'.

To determine the spatial dimensions and geometry of an object using triangulation, a 'line of sight' or 'ray' is drawn between the camera location and a specific point on the object. These lines are drawn on multiple photographs, with each photograph overlapping its neighbour by around 60%. When these images are superimposed, the crossing or 'intersection' of these rays determines the x-coordinate, y-coordinate, and z-coordinate of the specific point of interest in three-dimensional space relative to all other point locations. A collection of these points is known as a 'point cloud'. A point cloud can be used to generate a 3D model of the external surface of a target object. The greater the number of points, the more accurately the object can be captured and replicated. This process is also referred to as '3D stereophotogrammetry'.



Below ↓ **CyArk's 3D representation of the Gateway of India in Mumbai**

150 YEARS OF PHOTOGRAMMETRY

The expression 'photogrammetry' was first used by the Prussian architect Albrecht Meydenbauer in the Berlin Architectural Society Weekly Journal in 1867. Albrecht was experimenting with photographic images as a cost-effective means to store detailed information about monuments and historic buildings. He aimed to create a national 'Cultural Heritage Archive' that would enable objects of importance to be accurately reconstructed if damaged or destroyed.

By taking indirect measurements of these structures on photographic images, he believed he could replace the expensive and often dangerous task of measuring buildings and monuments directly. However, it would take him more than 25 years to develop the methods and instruments necessary to realise this idea.

During his initial experiments, Albrecht believed that any camera could be used for photogrammetry, but it quickly became apparent that a purpose-built system would produce much better results. The first of Albrecht's photogrammetric systems combined a measuring instrument and photographic camera into a single device. This new camera was equipped with

Above
Using 3D laser scanning, photogrammetry, and drones, CyArk digitally captured details of the Mexico City Cathedral

a Pantoskop wide-angle lens (with a fixed focal length of 25 cm) and captured 30×30 cm images onto glass plates. Crosshairs were added to each image during exposure to form a simple coordinate system, and the camera itself was mounted to a tripod, which enabled highly accurate horizontal adjustments to the camera and image coordinate axis, as well as the vertical image plane.

With this new instrument, Albrecht performed practical tests in the town of Freyburg. Here, he recorded metric images from the church as well as a section of terrain for topographic purposes. Thankfully, Albrecht's experiment was a success and enabled him to secure valuable evidence on the suitability of photogrammetry for both architectural surveys and topographical mapping.

Having silenced his critics, Albrecht founded the Royal Prussian Photogrammetric Institute in 1885, which became the first photogrammetric institution in the world for the documentation of cultural heritage. Between 1885 and 1920, the institution recorded over 2600 objects in approximately 20,000 photogrammetric images.

PRESERVING CULTURAL HERITAGE

Fast-forward over 130 years, and CyArk has replaced Albrecht's early photogrammetric camera and tripod with a Nikon D810 digital SLR and GigaPan EPIC Pro robotic camera mount, to form a vastly updated version of the original 19th-century system. The non-profit also makes use of a DJI Phantom 4 Pro in 'Point of Interest' mode to capture overlapping photos of roofs and other areas that are impossible to document from the ground.

CyArk's team has also been developing a low-cost heritage documentation kit, which includes a digital camera, laser rangefinder, tablet computer, GPS receiver, compact tripod, and video tutorials on how to digitally document a site.

"A key component of the kit was its use of off-the-shelf components that could be easily obtained by anyone, and the methods and take-away resources that can easily be adopted and referenced in the field," explains CyArk's Heritage Program Manager, Kacey Hadick.

At under £1200, this relatively inexpensive solution relies on advances in photogrammetry software, combined with laser distance measurements to quickly and accurately document heritage sites, while requiring only moderate technical ability from the operator.

DIGITAL DOCUMENTATION TOOLS

Photogrammetry images are shot as bracketed HDR to capture details in both light and dark areas. Three photos are taken: one using the camera's recommended settings, one intentionally underexposed, and one intentionally overexposed. The resulting images are then processed using Lightroom, X-Rite, and Photomatix software before RealityCapture automatically extracts 3D models from the images. The result is an incredibly complex digital model with a hyper-realistic photographic texture. →

ORIGINS

Photogrammetry is derived from three Greek words: 'phos' or 'phot' meaning light, 'gramma' meaning letter or drawing, and 'metrein', the Greek noun of measure. Although Albrecht Meydenbauer coined the term, French military engineer Aimé Laussedat is credited with the invention of photogrammetry in 1861, which he used to produce highly detailed topographical maps.

TRY IT YOURSELF

If you're interested in quickly creating 3D models from images captured on your digital camera, we suggest trying the 'ARC 3D webservice' by the University of Leuven in Belgium. After registering, upload your images to their server and the 3D spatial data will be extracted and emailed to you automatically. For more information, visit: **hsmag.cc/IOqMBh**

Below
CyArk documents St Sophia, an 11th-century church in Skopje, the capital of North Macedonia

What is LiDAR?

Below ↓
CyArk returned to Rapa Nui in 2019 to provide a more intensive workshop where attendees were able to share the methods they use to process the data

LiDAR stands for light detection and ranging. It describes the process of rapidly firing pulses of laser light at a solid surface to measure its distance from the laser source. By calculating the time it takes for light from each pulse to return, the distance of the target surface can be measured within fractions of a millimetre.

The formula used to calculate how far a light photon has travelled to reach the object or target surface is relatively simple. Multiply the speed of light by the time it takes for the laser light to return to the LiDAR sensor (known as 'time of flight') and then divide that number by two.

The raw data collected by LiDAR sensors typically contains millions of measurements and produces a much denser point cloud than those collected using traditional photogrammetry techniques. These high-density point clouds enable not just the distance of the object to be determined, but its size and shape as well.

Left

Following CyArk's visit to Easter Island, the Community began documenting cultural resources

FIRE THE LASER!

"3D scanning is growing within the heritage field as an acceptable standard for site documentation," says Justin Barton, Chair of CyArk's Technology Center Advisory Board. "3D laser scan data can easily be converted to CAD and other imaging programs for conservation, management, and restoration works, as well as virtual tourism, education, and information dissemination."

The term LiDAR was first introduced by William Middleton and Athelstan Spilhaus in 1953, to describe attempts made by the Bureau of Intelligence and Research in the United States to measure cloud base heights using pulses of light in the late 1930s. However, the rapid development of modern LiDAR technology would not begin until the first laser was built in 1960 by Theodore Maiman at Hughes Research Laboratories in California. A decade later, early LiDAR systems were used to successfully measure lunar surface height and topography during the Apollo 15 mission in 1971, piquing global public interest in this exciting new technology.

VIRTUAL EXPERIENCES

Today, CyArk uses LiDAR to document and record a large variety of objects, structures, buildings, and topographies. Everything from small archaeological finds to vast geo-referenced landscapes. Making this data viewable on the web is at the heart of CyArk's mission. The Internet Archive houses the media collected from each of its projects, providing free access to all project data. "The mystery of what utility people might find in the data is part of the excitement," says John Ristevski. "Some of the exciting frontiers are in virtual reality and augmented reality, and we are excited to see what type of experiences people can build around heritage data."

To find out more about CyArk's work and to explore their vast online archive of heritage data, visit **cyark.org**. □

BACK STORY

Before launching CyArk, Ben Kacyra founded Cyra Technologies, which developed the first truly portable and 'eye-safe' long-range laser scanner in the late 1990s. Leica Geosystems acquired the company in 2001, which left Ben free to pursue his passion for archaeology and, through the Kacyra Family Foundation, put the laser mapping tool he developed into the hands of as many cultural heritage professionals as possible.

Below

In 2017, CyArk travelled to Easter Island to document sites in the Rapa Nui National Park

HackSpace magazine meets...

Liverpool MakeFest

Showing off the best makers in the UK

The news that UK Maker Faire was to shut its doors came as a bit of a surprise to many. This vibrant weekend of makers meeting, sharing, and learning was absolutely brilliant, and left us fizzing with ideas after our visits there. We're sad that it's gone.

Makers being makers though, if there's demand, it will be filled. And that's exactly what's happening in Liverpool with Liverpool MakeFest. Now in its fifth year, this is the UK's biggest free maker event, attracting thousands of visitors, and its vision of a free maker-focused festival is spreading far and wide.

We visited the mid-Victorian splendour of Liverpool Central Library, the home of MakeFest, to talk to the founders – Denise Jones, Mark Feltham, and Caroline Keep – to find out what makes this event special. →

Above ⇧
Caroline Keep, Denise Jones, and Mark Feltham, the founders of Liverpool MakeFest

Images
Mark McNulty

HackSpace Hello! Thanks for having us over here. How did the three of you come together to start Liverpool MakeFest?

Caroline Keep I was a geotechnical engineer, Mark's an academic, and Denise is a librarian. We bumped into each other watching a workshop in lantern making. Mark had all the academic experience. When I came to work with Mark on his makerspace, I was the geeky maker – he didn't even have a smartphone at that time. I got the education bug and then moved into secondary school teaching.

Mark Feltham It all started over there, as a chance meeting. We bumped into each other and got chatting. Within six weeks, we'd filled the library. What we thought would be a one-off, since then it's taken off.

HS So no business plan, no franchising fees, no world domination?

CK We've just winged it. We made all the banners, bunting. The first year my PGCE fund paid for MakeFest! This building reopened again in 2013, and in 2014 we were lucky that they were running a programme of events and initiatives to make it a really vibrant building, so it was the right time as well. We thought we'd have a little room off to the side and get maybe six tables. We'd already done a Mini Maker Faire, and we've always been good friends with [local makerspace] DoES Liverpool, so we were confident we'd get at least a few people turning up. And in six weeks we were full.

MF We pulled the first one off, we're talking the first three floors of the library and 60 makers, for £850. And that included feeding them and making badges as well.

HS For context, this building is huge. It's bigger than most libraries; it's probably about the same size as the Life Centre in Newcastle, where UK Maker Faire was held until recently. It must have helped having a librarian on-board to negotiate with the powers that be?

Denise Jones I had to sell it to the people in charge back then, which was the head of service and manager of this building. The Department for Culture, Media and Sport has a Taskforce for Libraries, which is funded until next year. We're close to finishing the national guidance now for the Taskforce – the idea is to get one of these in every library. We wanted the guidance doc to be inclusive of museums and libraries, because we knew that Manchester had opted to put their MakeFest in a museum. We've got Chester and Stoke MakeFest, and there's one in the pipeline in Wrexham. We were having the same conversations over and over again, so we decided to write a document – how to run a MakeFest.

> Everybody at some point in their life will set foot in their **local library, so everybody thinks it's theirs**

I'm the chair of the North West Universal Offer Group, which is associated with Libraries Connected, which is the main lead that's funded by the Arts Council on public libraries.

After the second year of MakeFest, we went in on funding from the Arts Council and we got MakeFest funded. I try to find other pots of government money, so that we can do things together with local makerspaces.

I bought a few mobile carts that have tech in – Raspberry Pis, Arduino, stuff like that – so you can move them around depending on where you're having a session, and we got 3D printers as well. Other librarians, such as Amy Hearn [Digital Inclusion Coordinator at Leeds City Council], got hold of a load of micro:bits, and she's sent them out to other library services.

With the pot of money we ended up with, we commissioned different makers to get going with workshops, and gave them the kick-start they needed.

HS It may be a basic question, but why libraries?

MF Shared ethos. No politics. Makerspaces require a community of makers. You need a community of makers before you even get to put a makerspace anywhere. We knew there were loads of makers around Liverpool, we were lucky with that. But they're all really disparate. We wanted to find one place to bring them all together, and the library was the obvious place.

DJ This has been happening for a long time in Canada, the US, Australia. We were slow to pick it up. The public library has always been involved in self-directed learning initiatives – it's a people's university. Recently it's been at the forefront of digital inclusion, giving access to tech and IT for people who would otherwise not be able to afford it. If a person hasn't got a computer, hasn't got WiFi, the first place they can come is a library. It's open to all, you don't have to identify yourself.

It's a public space, and it's non-threatening. You don't have to be a member to come in. Everybody at some point in their life will set foot in their local library, so everybody thinks it's theirs. So you've got open access, shared public information, shared communications, all of that.

CK We wanted a space that would make people feel equal, that no one maker was more important than any other maker; that each person could bring the thing that they wanted to showcase. We've had tech companies with quite a lot of money alongside scholastic kids. It's mix and →

Above ↗
One of the spin-offs that have come out of MakeFest is Little Sandboxes, which takes making out to deprived areas of the city

Above
This year's MakeFest includes an exhibition of women in science – come and meet some real-life knowledge expanders!

match, and it's important that anyone can be part of that community.

MF And not something that was going to be branded by a company that wanted to raise their profile. In the first year, when we didn't know if we were going to do this, we did have offers of several thousand pounds from different sources, but they came with strings attached. They wanted certain things included, and they wanted their branding everywhere. We didn't want that. That's one reason we didn't have a Mini Maker Faire, because you've got to pay thousands for the branding, and then you have to pass on the costs.

DJ For us, our USP is that we're free. We never thought that any maker who comes and volunteers their time for free should really feel that the public would get charged for attending. Nor should they pay to set up stalls or anything else. I can understand why Maker Faire went that way, and I'm not saying it's a bad thing, but it's not for us.

CK It made us feel like we had, from all the disparate places around the city, there's this one day we come together somewhere that's in public. From that, it built collaborations from one maker to the next, which meant that loads of stuff came out of it that never would have. So it kind of spun off.

MF The same time the first MakeFest started, I was doing a roll-on, roll-off makerspace in Liverpool John Moores University. We were trying to build making into the curriculum, learning in different kinds of ways. It became apparent that, as the national curriculum changed, the creativity of the students coming through was waning. They were being taught to pass tests. It was too late to instil any creativity in them by the time they were coming to me, so I thought that I needed to get in earlier.

I started doing outreach into schools. And it's gone further and further back. It started with later secondary years, and it's now gone back to year 8 and 7. And now it's even down to primary years, through code clubs. So my maker stuff now is a little bit bespoke within modules at a university degree level for biologists.

CK We kicked off MakeFest when I was in my PGCE year, so I went about trying to find a school to put a makerspace in. I picked a brand new school, new build, all new tech, and it was absolutely beautiful, but I didn't realise the politics of schools. When I got there, I didn't realise that a brand new school has a long way to go. There was no safeguarding policy, no plan for homework, no curriculum, nothing. So it took two years to get it into shape, and the year after I got it into shape the Conservative government came in, policies changed, and they shut my school, practically overnight. We had all this kit in the building that was ready to set up a makerspace that then got literally locked in a building, and you couldn't get to it.

> "I just did what every makerspace does – went and asked everyone **for stuff they had. Parents, teachers, universities… everybody**"

I then said to these two, "You know what? I've had enough. I can't face schools. I can't face the education system, it's so ingrained and so difficult," and I decided to take my redundancy money and go back to science.

At my school I had a lot of people interested because they knew that I was a very tech teacher; that's where my expertise was. By then, I'd already won the engineering award and the digital excellence award [Caroline won't bring this up, but she is the current reigning TES New Teacher of the Year].

My current school emailed me about eight or nine times, but they said: "How are you? Are you OK?" The head, John Carlin, who had just taken over, was one of the youngest heads in the country. He'd been at the school 14 years and decided to go for the headship himself, because they'd had several head teachers and nothing had got any better.

He really sold it to me. He cared about the school, he cared about the kids. I thought I'd work there for six months and see how I felt. By that point, I'd taken everything I knew about MakeFest out of my CV. He knew I was a physics teacher, but I'd taken everything to do with making off my CV.

Six months in, I sat him down and asked for a room to build a makerspace. Denise and Mark came in, and he took us into an old DT room. There were these buildings that had been left to rack and ruin. I took it upon myself to clean out this room over the course of three days, then set about getting stuff to go in it. I had no funding, no budget, and no plan, so I don't know how I pulled it off. I just did what every makerspace does – went and asked everyone for stuff they had. Parents, teachers, universities, other makerspaces, everybody. Now I reckon it's the best-stocked makerspace in the area. On the back of that, I started to run workshops and lessons.

HS What have we got to look forward to this year and beyond?

CK That's a good question. We've got some corking stuff coming this year. We've given it the theme 'space and time – creativity in the making'. We've got events planned for the Apollo anniversary, and [just] before MakeFest we're going to kick off with a music day, showing people how to make music, and making the instruments to make music. That's another spin-off that's come out of MakeFest: the MakerNoise Unconference at Edge Hill University. →

Right ↗
Liverpool MakeFest 2019 is taking place at the Central Library, Saturday 29 June 2019, and it's completely free to attend

MF We've always felt that we hold MakeFest in trust for makers. In terms of where it goes long term, I don't see it ever becoming more than a one-day event here, because one day is good. It gives people Sunday to get over things, and get home because they have day jobs on a Monday. We're always very sensitive about that, that we don't want to take up too much of people's time. The other thing is that I don't see it spilling out into a bigger building; it's always going to be in the library. But the way to grow it is to put it in other libraries. Not to make this one, Liverpool, bigger and take over. Then each maker community gets its own feel, and its own vibe – Stoke MakeFest has a very different feel than ours, because their maker scene is different than ours, and their city is different than ours.

The other way to expand it is that, rather than by just expanding to other cities, you can have more events on throughout the year. Rather than being solely a one-day event, you can have all these spin-offs, so once a month there's something going on. Rather than it just being about tech and digital, we've always liked to have some sort of fantasy element. Things like *Doctor Who*, *Star Wars*, Darth Vader, K-9 – the kids love that. We have a lot of friends who are into steampunk; they get roped in to do front of house duties. You know what the funny thing was from the first one? Not only did the public enjoy it, but also the makers. It's kind of like a musician playing an acoustic set. We've got a get-together on the Thursday before, we've got a Friday night party going, we always do an after-party. The public come on the Saturday, but there's always stuff going on that week for makers.

> "The way to grow it is to put it in other libraries. **Not to make this one, Liverpool, bigger and take over**"

In addition to always wanting it to be free for the public and for the makers to not have to pay for their stand, we feel very strongly that we should give something back. We always give them lunch, we always give them a badge, and there's always a party. We can't pay them, but it's our way of showing our appreciation to the makers who come and make it what it is. The celebration and sharing are big parts of the maker ethos.

People like to show [their projects] not to show off, not to say 'Look at how clever I am', it's more to say 'Look at this awesome thing, isn't this cool'? Trying to explain that to people can be tricky. You can make this: here's how you do it. That's the ethos.

CK I always feel with MakeFest – you said it's like an acoustic gig. I always envisioned it as Liverpool's party for makers. It's our little get-together, and that's how I like it. □

Above
Liverpool Central Library was renovated a few years ago – the precious books went into temporary storage in a salt mine in Cheshire to keep them dry

NUTS & BOLTS

IMPROVISER'S TOOLBOX

Keeping the world from falling to pieces since time immemorial

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.

You don't have to look too far to find a bolted connection. From holding the wheels on your car, to securing Tower Bridge, nuts and bolts are one of the most common elements in construction. Despite their simple appearance, there are several elements and a lot of physics involved behind these simple-looking tools.

You can trace the threads on the bolts to around 400 BCE, but a threaded bolt with a matching nut that serves as a fastener only dates back to the 15th century. That's when Johannes Gutenberg started using screws in his printing presses. By the end of the 16th century, the Besson factory in France began manufacturing bolts and screws that were mostly used to hold together sleighs and carriages. Besson is also famous for coming up with a screw-cutting lathe that was later improved by Henry Hindley of York.

It was the Industrial Revolution that sped up the development of the nut and bolt and put them firmly on the map as an important component in the engineering and construction world. Back in those days, making bolts was an excruciating process. The problems compounded when the process was industrialised, as each company manufactured its own threads, nuts, and bolts. Joseph Whitmore helped standardise the size of the screw threads in the UK, and all was good until William Sellers proposed what evolved into the American Standard Coarse Series and the Fine Series. The inconsistencies were only ironed out after the lack of standardised screw threads resulted in damages to the Allied forces in World War I. The Unified thread was first selected by all countries that used imperial measurements; it was later refined to the ISO metric thread, which is what we continue to use today.

Stroll down the fixings and fittings aisle in the hardware store, and you'll know that nuts and bolts have come a long way since the days when they were fabricated by hand. As they cemented their role in the construction process, people started putting a lot more emphasis on analysing joints. There are plenty of ways for bolted joints to fail, but vibration-induced self-loosening is surely one of the most dangerous ones. Over the years, engineers have come up with dozens of ways to make sure the world doesn't fall apart.

As you can imagine, nuts and bolts also play a crucial role in a DIYer's toolkit. But our ingenious makers need little else to cobble together the awesome builds featured over the next few pages.

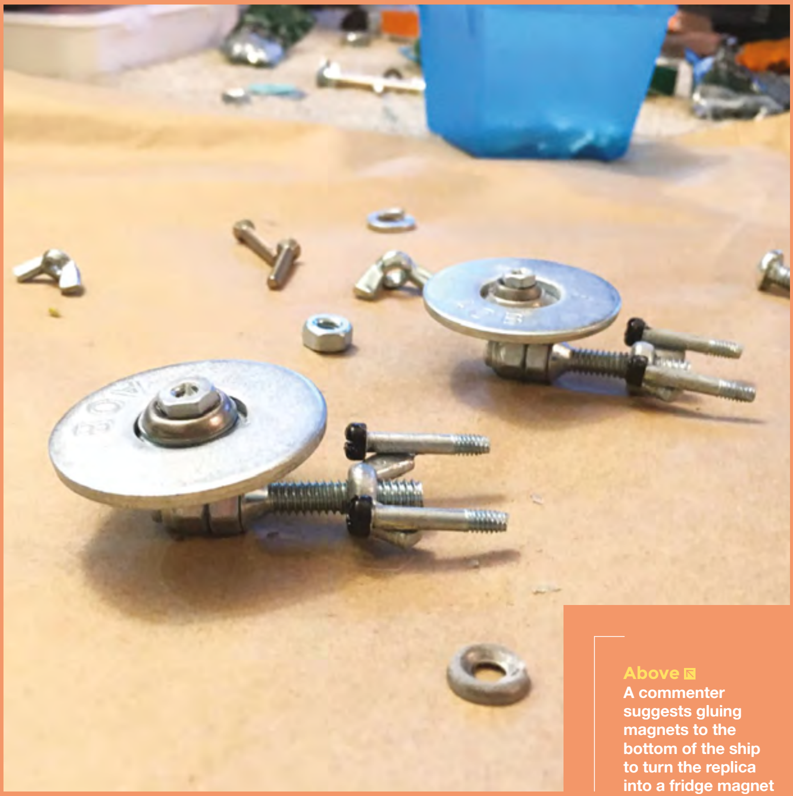
ENTERPRISE REPLICA

Project Maker
Gaia Liu

Project Link
hsmag.cc/UlvGUw

Nuts come in all shapes and sizes. There are wingnuts, hex nuts, slab-weld nuts, T-nuts, and several others. A wingnut, lodged in the gap of the carpet, inspired multidisciplinary designer Gaia Liu to use it, together with some other knick-knacks, to assemble a replica of the original Star Trek Enterprise 1701-A. He glued two carriage bolts, with pan heads, to a wingnut to create the pair of engine nacelles. A couple of hex nuts and a screw completed the secondary hull. Similarly, he used a screw and some glue to fuse a couple of washers into three separate components that act as the bridge, the primary hull, and the main phaser. Gaia has detailed illustrations of the entire build process, and it doesn't take much time once you've assembled the correct components. And therein lies the catch. While it won't be too much of a chore to round up the nuts, bolts, washers, and screws used in the build, Gaia hasn't published a list of materials, despite repeated requests. □

> " IT DOESN'T TAKE MUCH TIME ONCE YOU'VE ASSEMBLED THE CORRECT COMPONENTS "



Above A commenter suggests gluing magnets to the bottom of the ship to turn the replica into a fridge magnet

NUT FLASHLIGHT

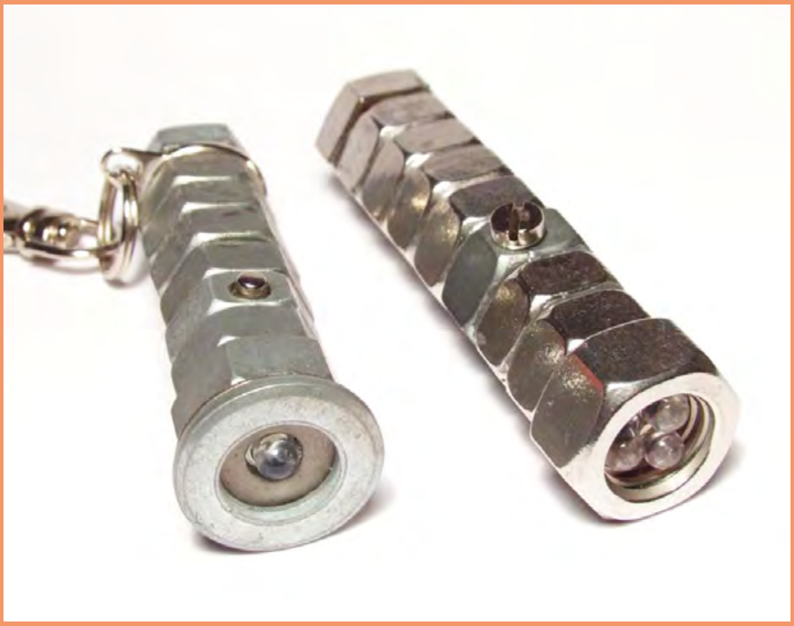
We aren't really sure what prompted Swiss engineering student Silvan Roth to create a flashlight purely out of hex nuts, but we are glad he did. Not only does it have a unique appearance, but it's also sturdy and lightweight. He's created two versions of the flashlight with the second one having a more robust construction. Besides the hex nuts, you'll need some drilling skills to make room in the side of a nut for the switch. You'll also need to saw a small groove into the nuts to pass a wire through the battery compartment. The rest of the construction isn't too cumbersome, but the process is fairly involved and well worth the end product. In fact, Silvan almost became a victim of his own success: "A few months after publishing the original flashlight Instructable, I came across a Google forum where a group of hobbyists were discussing selling a DIY kit to let others make their own nut flashlights based on my concept." Silvan was flattered but also annoyed that people were commercialising his idea without ever contacting him. He shadowed the forum for a few months, but the group eventually seemed to have lost interest in their project. □

Project Maker
SILVAN ROTH

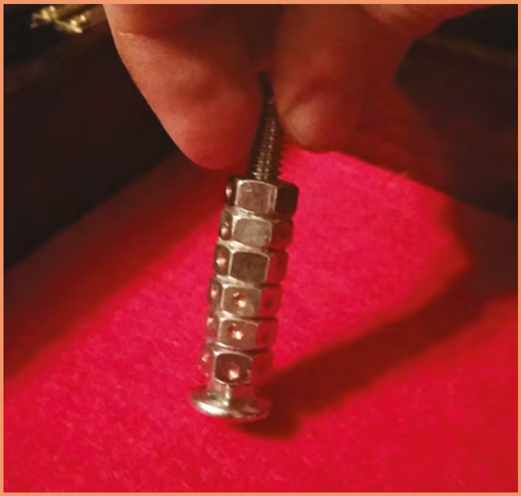
Project Link
hsmag.cc/f0aaMt

> "BESIDES THE HEX NUTS, YOU'LL NEED SOME DRILLING SKILLS

Below
Scroll through the comments section for an informative discussion on the arrangement of the LEDs (serial vs parallel) and the benefits and drawbacks of each approach



HEX NUT DIE



Travis is an avid tabletop gamer: "One day when I was putting something together for work and a co-worker dropped a nut, and out of habit I said runaway dice." Going along with the thought, it took some tinkering for Travis to come up with the perfect Christmas gift for his gaming group – a die made of hex nuts. The steampunk die has six hex nuts on a carriage bolt, and Travis says you can either roll it on the table or spin it like a top with a little practice. The construction isn't complicated. First, glue the nuts on the bolt and tape them up. Next, mark the pips with a highlighter before drilling them, in any order as long as the opposite sides add up to seven. Travis has shared the order that he's used. After drilling the pips, spray paint the die, let it dry, and you're done. □

Project Maker
Travis Peterson

Project Link
hsmag.cc/fjTAOG

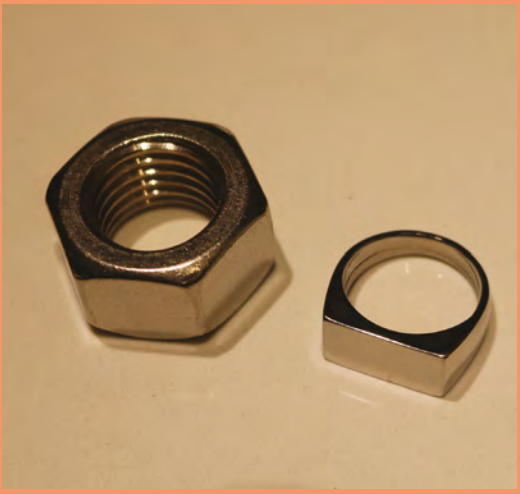
Left
A commenter warns that the die isn't balanced and will tend to roll a six more often, but it's still quite functional

NUT RING

Project Maker
Andrew W

Project Link
hsmag.cc/stcjGd

Andrew has been making rings from nuts since he was 14. He's refined the process since that time when he spent a week filing away the nut. Decades later he can now make a ring in a couple of hours using proper power tools: "The reason why I revisited this ring was I thought it would make a great gift for my uncle Bob. He wears it every day, and loves it so much he even got it engraved with his name." Andrew first marked out the shape of the ring on the nut and then used the flap disc to remove the excess. He then reamed out the centre of the nut with a rotary tool, but suggests you can use a die grinder or even a round file. Once the rough shape is done, use the grinders to smooth out any rough sides. The process is tool-intensive, but Andrew's video does a nice job of visualising the process. □



Right
Andrew has sanded the inside to make the ring comfortable to wear, and used some metal polish to give it a high-gloss finish

MAKE BUILD HACK CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
BUILD A ROBOT
SPACE

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

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

Electronics 101.11
The 555 timer

Delays, pulses, buzzes, and beeps

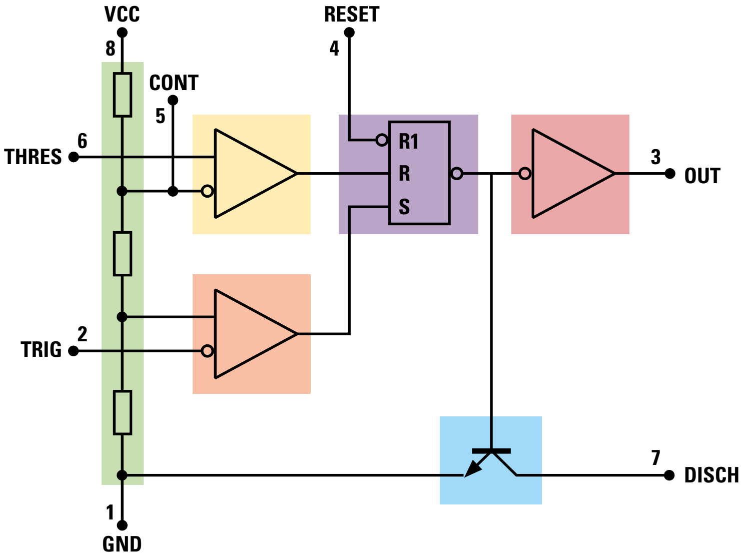
Dave Astels

daveastels.com

Dave's career started in the 8-bit days, with the Z80 and 6502, and he's been working with computers ever since. Check him out at: **daveastels.com** and **learn.adafruit.com**

Few integrated circuits have been used as much, for as long, and for such a variety of applications, as the humble 555 timer. You can find it in devices ranging from toys to spacecraft. The 555 has been in continuous production since the early 1970s. It is available in several varieties: there are bipolar and CMOS versions that are much the same, but vary in small (but sometimes significant ways); we'll just be dealing with the basic bipolar version in this article. The 555 is also packaged in single (555), dual (556), and quad (558) packages. The 558 is no longer in production, but you can easily get the 555 and 556 in both through-hole and surface-mount packages.

Let's peel back the black plastic packaging and take a peek at how this ubiquitous ship operates behind the scenes.



HOW IT WORKS

The 555 can operate with supply voltages between 4.5 and 15 volts. The voltage has no impact on timing characteristics.

Figure 1 shows an abstract view of the internals of a 555. We'll look at each block in turn, most of which we have already discussed in previous issues.

Voltage divider

Shown in green in **Figure 1** is a voltage divider. This is constructed from three identical resistors and stretches between VCC and GND. It provides two reference voltages: 1/3 VCC and 2/3 VCC.

This divider is the key to the VCC-independent behaviour of the 555. The circuit operates relative to the voltages from this divider, which are relative to VCC.

Comparators

The voltages provided by the voltage divider (1/3 and 2/3 VCC) connect to a pair of comparators.

The 1/3 VCC connects to the non-inverting input of the trigger comparator (highlighted in orange). Its inverting input is connected to the TRIG input (pin 2). The 2/3 VCC connects to the inverting input of the threshold comparator (in yellow). That comparator's non-inverting input connects to the THRES input.

When the TRIG input drops below 1/3 VCC, the trigger comparator's output goes high. Conversely, when the THRES input goes above 2/3 VCC, the threshold comparator's output goes high.

There's one other wrinkle: the inverting input of the threshold comparator (which is connected to 2/3 VCC from the voltage divider) is also connected to the CONT input (control) which allows external circuitry to directly override the voltage THRES is compared against. This also impacts the voltage to which TRIG

Figure 1
A functional view inside the 555

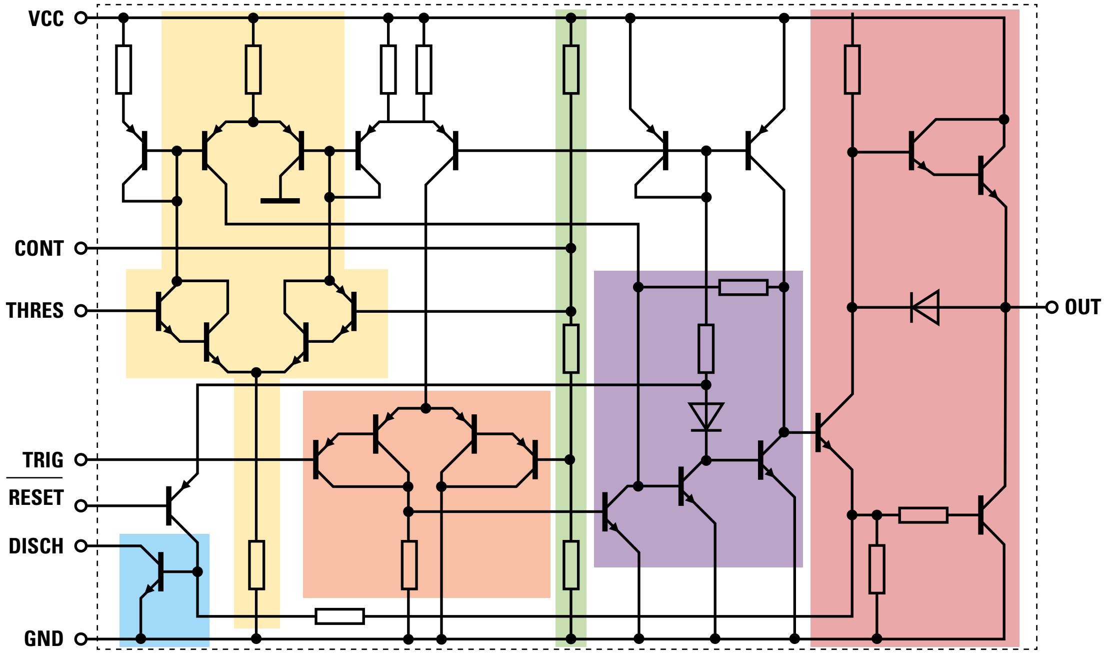


Figure 2 The schematic of the 555

QUICK TIP

The 555 is one of the most produced and used integrated circuits in history. It first appeared in 1971, and in 2003 it was estimated that one billion were being made each year.

is compared; since the resistors are of equal value, TRIG will be compared to half the voltage on CONT. In most circuits, CONT is not used and is connected to ground through a 0.01 µF capacitor to shunt any random noise to ground.

> "This divider is the key to the VCC independent behaviour of the 555"

RS flip-flop

We haven't talked about flip-flops in previous issues. They are a very basic digital logic memory element. They can have their state set to high or low (i.e. true or false) and they retain it until it is otherwise set. There are various types of flip-flops that vary as to how their state is changed. In the case of the 555, the flip-flop is a simple reset-set style, aka an RS flip-flop. When the R(eset) input goes high, the flip-flop's state switches to low. When S(et) goes high, the state switches to high. Note that the state changes when the inputs change from low to high, not when they *are* high. As noted in **Figure 1**, the output of the flip-flop (highlighted in purple) is inverted, so it's the opposite of the flip-flop's internal state. The third input to the flip-flop is R1, which resets the flip-flop to a low state regardless of other inputs. The circle on it denotes that it is active low. When it is low, the flip-flop is forced to a low state and remains so until the reset input goes high again. When it is high, the R and S inputs control the state. The R1 input connects directly, and only, to the RESET input of the 555. This is normally connected to VCC if it's not being used.

The R input is driven by the threshold comparator, while the S input by the trigger comparator. The result of this is that the state switches to high when the TRIG input drops below 1/3 VCC, and to low when THRES exceeds 2/3 VCC.

Output buffer

The inverted output of the flip-flop is again inverted and buffered to drive the OUT pin (pink in diagram). Thus OUT mirrors the internal state of the flip-flop.

Transistor switch

The final piece of the 555 is an NPN transistor (cyan in diagram) whose emitter is connected to GND, and whose collector connects directly (and only) to the DISCH pin. As we will see, this is used to discharge the timing capacitor. The transistor is controlled by the signal between the flip-flop and output buffer, i.e. the inverse of the flip-flop's internal state. So, when the flip-flop is reset, the transistor is on, and DISCH is connected to GND.

Schematic

Figure 2 drills in further, going from block diagram to discrete components. Each block is highlighted using the same colours. For all of its usefulness and flexibility, it is actually a rather simple circuit. If you want to explore further at this level, Evil Mad Scientist (**evilmadscientist.com**) sells a 555 kit using →

YOU'LL NEED

- Solderless breadboard
- Jumper wires
- 555 ICs (at least two)
- Various resistors and capacitors as shown in the circuits, as well as more to experiment with; getting a variety pack is best
- 1 100 kΩ variable resistor (potentiometer)
- 4 or 8 Ω speaker
- Piezo buzzer
- Several push-button switches
- Arduino Uno, or similar board
- 220 Ω resistor and LED to connect to the output of various circuits

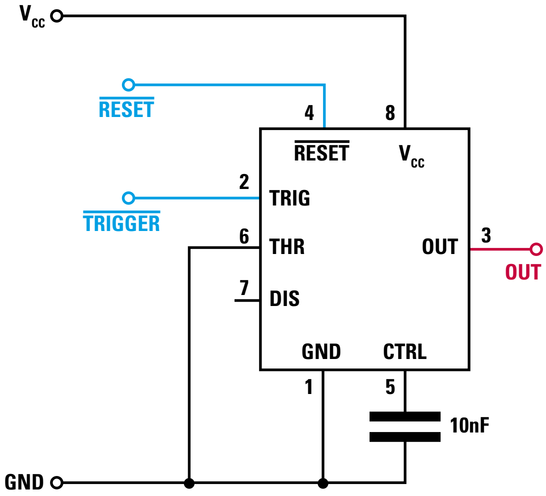


Figure 3 A 555 used in bistable mode

discrete, through-hole components. It's available directly from them, as well as some other sellers of electronics kits and parts.

The 555 supports three primary modes of operation: bistable or switch, monostable or one-shot, and astable or oscillating.

BISTABLE

We will begin with the simplest mode. Bistable mode uses the 555 as a simple switch by using the flip-flop more or less directly. This is done by rendering the THRES input useless by connecting it directly to GND. We then add a pull-up resistor to each of RESET and TRIG. Making RESET low resets the flip-flop, taking OUT low. Making TRIG low (which will be less than 1/3 VCC) switches the flip-flop's state to high and takes OUT high. Note that the inputs aren't completely symmetrical: RESET will take priority, despite what happens on the TRIG input.

Figure 3 shows an example circuit. Pushing the buttons controls the 555's state.

QUICK TIP

A 555 can be used as a small flip-flop with a flexible power supply.

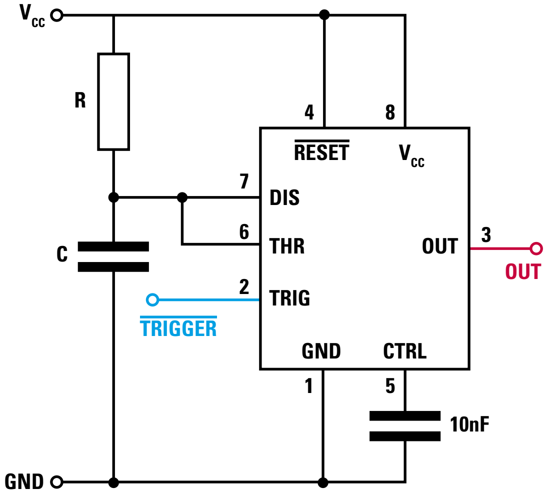


Figure 4 A 555 used in monostable mode

MONOSTABLE

Monostable means that the circuit is stable in one state. We can force it into the other state, but it will return to the stable state as soon as it can. **Figure 4** shows the circuit. Notably, THRESH and DISCH are connected together and to VCC through a resistor (R) and to GND through a capacitor (C). When TRIG drops to low (i.e. below 1/3 VCC), the flip-flop's state changes to high. One effect of this is to turn off the discharge transistor. That allows C to charge through R. When it has charged to the point that the voltage across it is 2/3 VCC, the threshold comparator flips its output, resetting the flip-flop. That takes OUT low (back to the stable state) and turns on the discharge transistor connecting DISCH to GND, which drains the charge from C and takes the voltage at THRES to 0 (since it's connected to DISCH). The 555 is now happily back in its stable state until TRIG once more switches from high to low.

The time that the output stays high is the time it takes the capacitor to charge enough to change the voltage across it from 0V to 2/3 VCC. Recall our discussion of capacitors back in issue 10. There, we said that, based on the charging curve of C through R, in time R × C the voltage across the capacitor will be at 63% of VCC. That's close enough to 2/3 for most purposes, so we say that the pulse length is RC seconds. Notice that the time is dependent completely, and solely, on the resistor and capacitor and is completely independent of VCC. **Figure 5** shows the signals of interest.

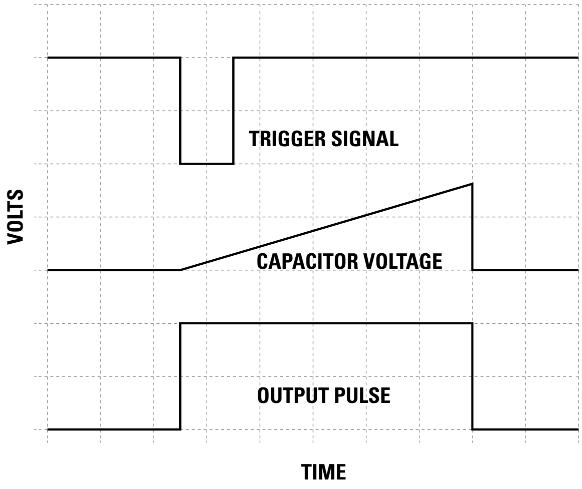


Figure 5 Significant signals in the monostable circuit

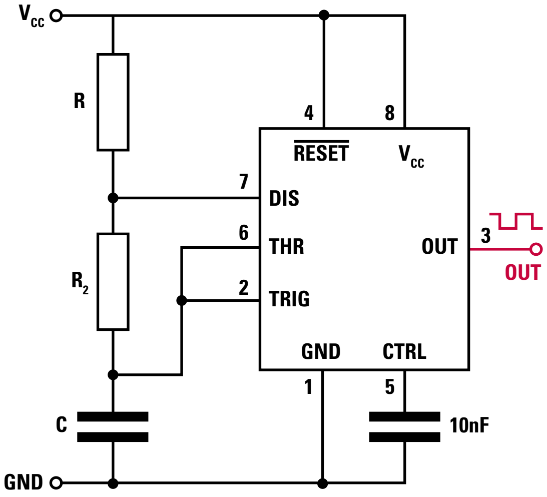


Figure 6
A 555 used in astable mode

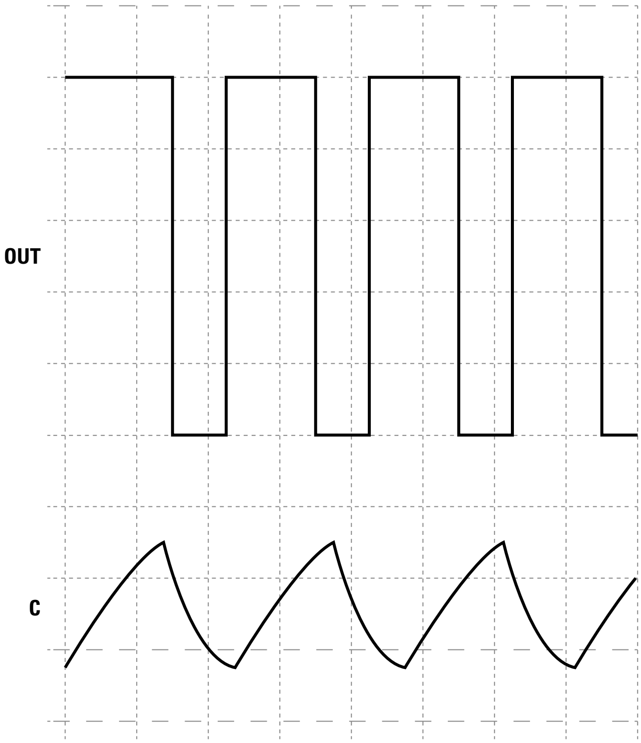


Figure 7
Significant signals in the astable circuit

ASTABLE

Astable means that the circuit doesn't have a stable state. It's continuously switching between them. **Figure 6** shows the basic circuit. There are two simple changes from **Figure 4**:

1. TRIG is connected to THRESH and the timing capacitor instead of being an external signal. This means that the 555 will be triggered when the voltage across C drops below 1/3 VCC. In the monostable configuration, when THRESH reached 2/3 VCC, C was immediately discharged (by the transistor connected to DISCH).

2. Since an alternating output is desired and immediate discharging would result in the 555 being triggered immediately, we need to slow the discharge. We can do that by adding another resistor (R2) between C and DISCH. C will then slowly discharge through R2, delaying the retriggering. This charge to 2/3 VCC, then discharge to 1/3 VCC cycle, will continue indefinitely, creating a square wave at OUT. Remember that OUT will be high when the capacitor is charging through both resistors, and low while it is discharging through R2. So the time it will be high is 0.693(R1 + R2)C seconds and it will be low for 0.693(R2)C. We have the expected RC products, but where did that 0.693 factor come from? That accounts for the voltage on C going between 1/3 VCC and 2/3 VCC rather than 0V and 2/3 VCC. Note that the relationship between R1 and R2 determines the duty cycle of the output (that is, the relative time it is high vs. low) independent of C. Combined with C they determine the frequency. Also note that the duty cycle will be above 50% since C charges through R1+R2 but discharges through R2 alone. Attaining a 50% duty cycle would require R2 = R1+R2. That's not possible because when the discharge transistor was on, VCC would be connected to GND. This is generally considered a bad idea.

Figure 7 shows the important signals.

EXAMPLE CIRCUITS

Since the 555 has been around for so long, you can find projects using it in many places online and in books. The following are a handful of useful examples.

Mail alarm

We can use a 555 in bistable mode to monitor your mail slot. A switch on the mail slot's flapper is connected to TRIG. This could be a lever switch that is pushed by the flapper or a tilt switch mounted on the flapper. RESET is connected to another switch. If there's a lid on the mailbox, it could be connected to that, or it could be a manually activated push-button. In either case, it is pushed/closed when the mail is retrieved. OUT could be connected to a piezo buzzer or light. A buzzer is shown in **Figure 8** (overleaf). →

QUICK TIP

For some applications, a 555 or two can do as good or better of a job than a microcontroller and code.

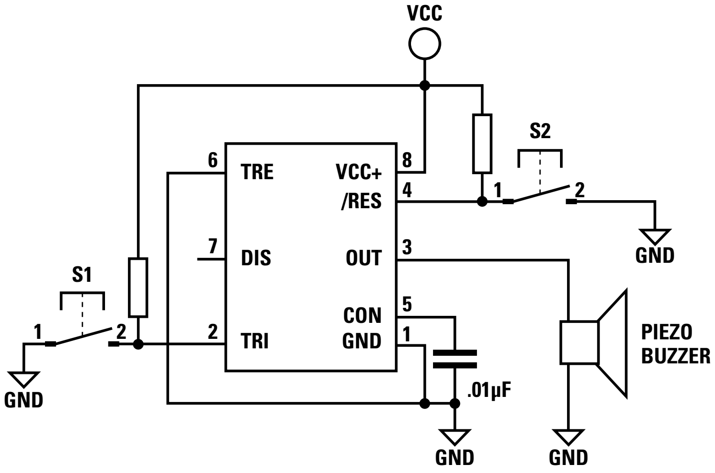


Figure 8 Mail alarm circuit

This could be done with a flip-flop IC, since that's what a 555 in bistable mode is. Using a 555 provides some benefit for a simple circuit like this, however. First, it's far more flexible in terms of power voltage. That means it can be directly battery powered. As the battery drains and its voltage decreases, the circuit will continue working until the voltage drops too low. Also, it only uses a single flip-flop. Flip-flop ICs are typically 14- or 16-pin chips containing at least two flip-flops. Contrast that with the 8-pin 555.

Toy organ

You can create a simple monophonic (one note at a time) toy organ by connecting a number of switches to select the frequency of a 555 in astable mode. You could use the switches to select one of the timing resistances, but that would affect the duty cycle (and thus the sound) as well as the frequency. A better approach is to use the switches to select between timing capacitors. This varies the overall frequency while keeping the duty cycle the same. As we saw earlier, duty cycle is only dependent on the resistors. See **Figure 9**. Capacitor values can be selected to generate desired frequencies:

.22 µF	52 Hz
.15 µF	78 Hz
.1 µF	111 Hz
.068 µF	170 Hz
.047 µF	230 Hz
.033 µF	348 Hz
.022 µF	490 Hz
.015 µF	718 Hz
.01 µF	1173 Hz
.0068 µF	1670 Hz
.0047 µF	2240 Hz

.0033 µF	3252 Hz
.0022 µF	4671 Hz
.0015 µF	6336 Hz
.001 µF	9237 Hz

Delayed signal

Sometimes you want an alert delayed a bit from the trigger event. To do this, you can use two 555s (or alternatively a 556). See **Figure 10** for the circuit. The first 555 is triggered by the external trigger signal. Its output will be a high pulse, whose length is determined by R and C. When it switches back to low, the second 555 is triggered, generating the required output signal pulse. OUT from the first 555 is coupled to the TRIG of the second through a capacitor with a pull-up resistor. This converts the change of OUT from high to low into a short low pulse for TRIG. The timing of the second 555 is set to generate the desired length of pulse.

A possible use for this circuit would be a tea timer. Trigger the first 555 when the water is poured, and have it set to delay for the steeping time required. You could use a rotary switch to select the timing resistor for different delays appropriate for different teas. Once that delay has completed, the second 555 turns on an indicator for some period of time. This could be a buzzer to alert you that your tea is ready.

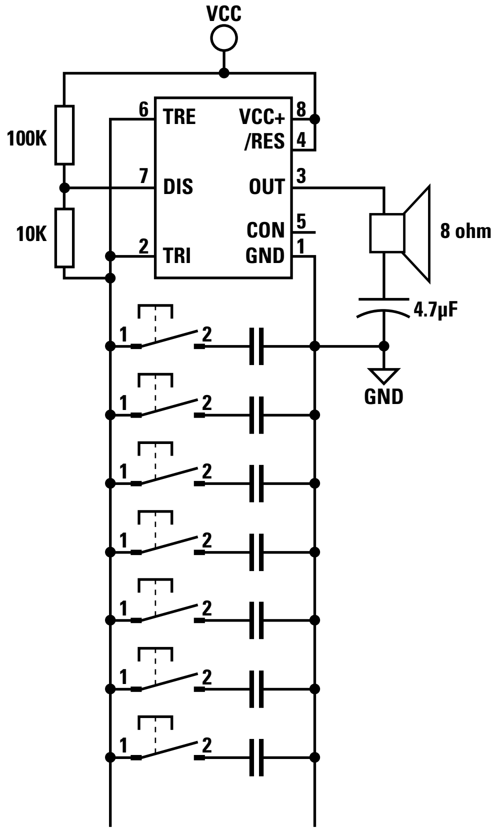


Figure 9 Toy organ circuit

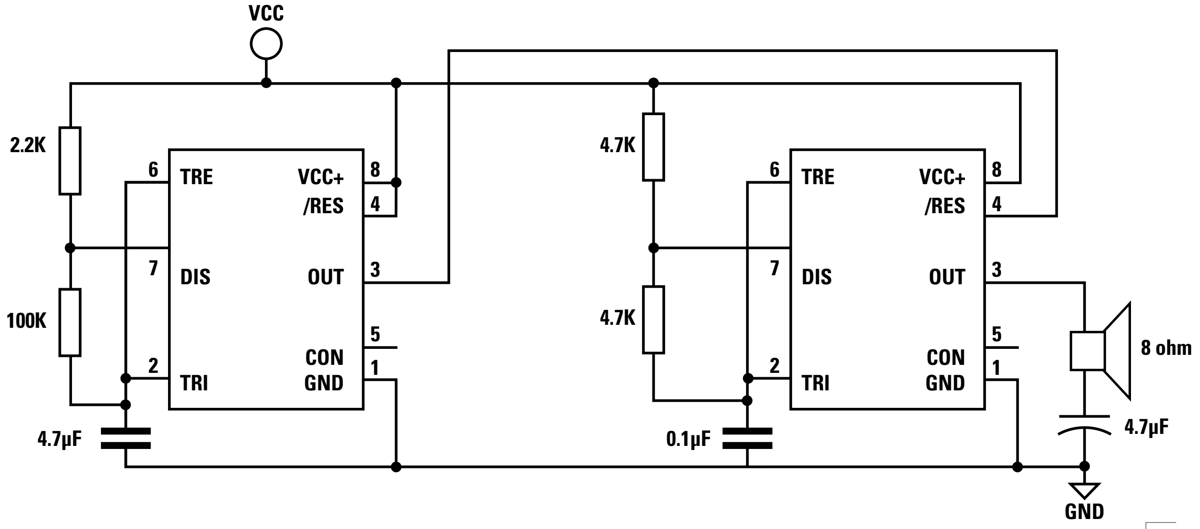


Figure 10 The delayed signal circuit

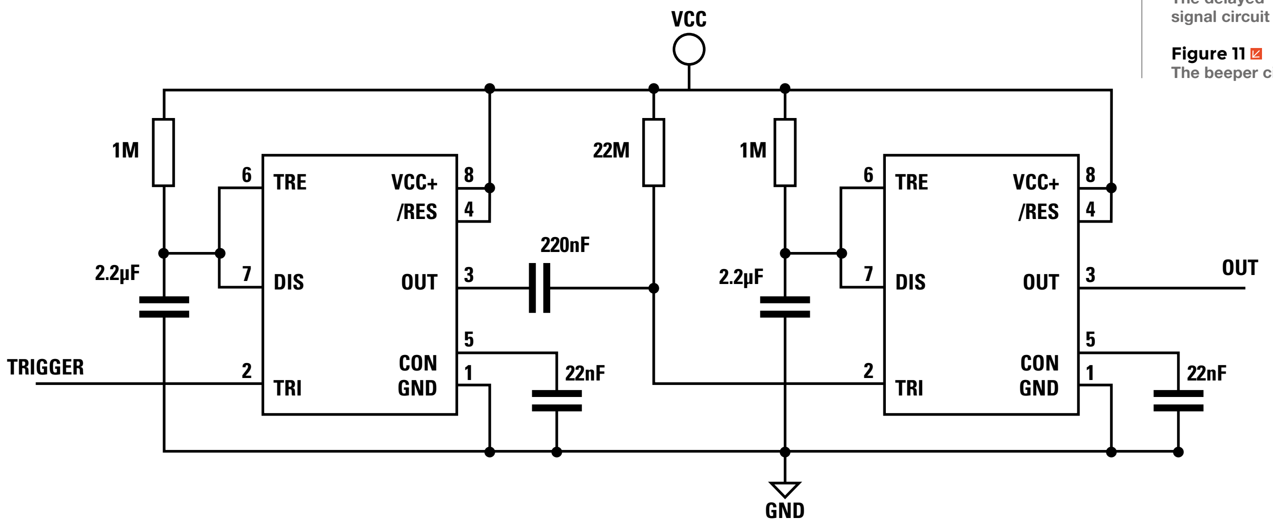


Figure 11 The beeper circuit

Beeper

We've seen how a 555 can generate a tone. We can use a second 555 to turn it into a beeper by using its OUT to control the RESET of the tone generator. **Figure 11** shows the circuit. When RESET is driven low (the stable state), the tone generator is disabled (quiet). When OUT takes RESET high, the tone generator operates. The result is a series of beeps. The pitch is determined by the second 555, while the interval and duration are set by the first.

Warbling tone generator

Instead of controlling the RESET of a tone generating 555 as above, we can use the OUT of the first 555 to provide additional current to the tone generator's timing capacitor, affecting how quickly it charges and thus the frequency of the tone. As the value of OUT changes, so will the tone's frequency (**Figure 12**, overleaf). →

LISTING 1

Arduino code for reading the pulse width

```
const int trigger = 8;
const int out = 9;

void setup() {
  Serial.begin(9600);
  pinMode(trigger, OUTPUT);
  digitalWrite(trigger, HIGH);
  pinMode(out, INPUT);
  delay(50);
  Serial.println("Starting");
}
```

```
void loop() {
  long count = 0;
  digitalWrite(trigger, LOW);
  digitalWrite(trigger, HIGH);
  while (digitalRead(out)) {
    count++;
  }
  Serial.println(count);
  delay(250);
}
```

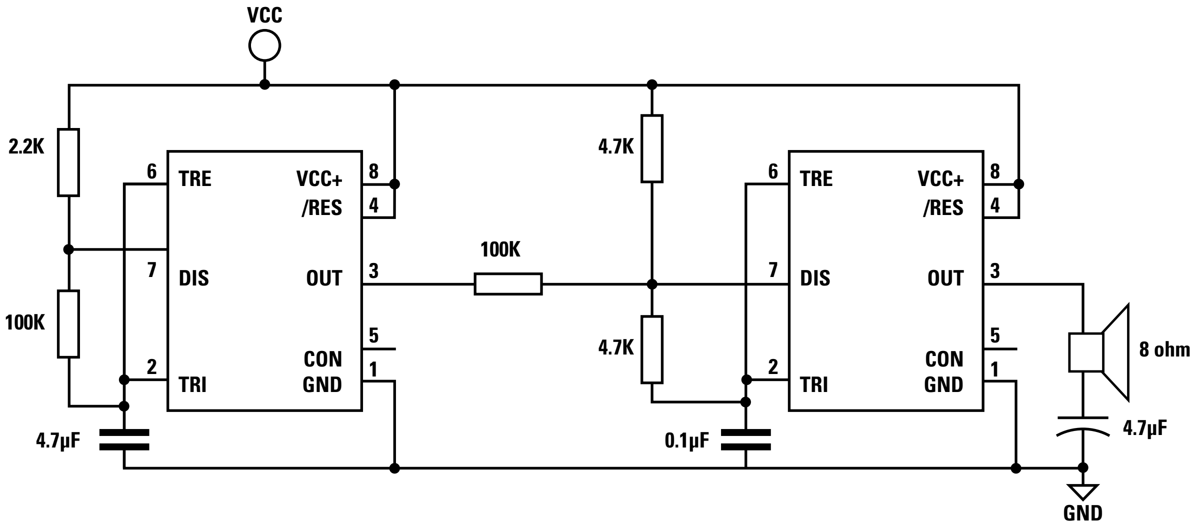


Figure 12 The warbling tone generator circuit

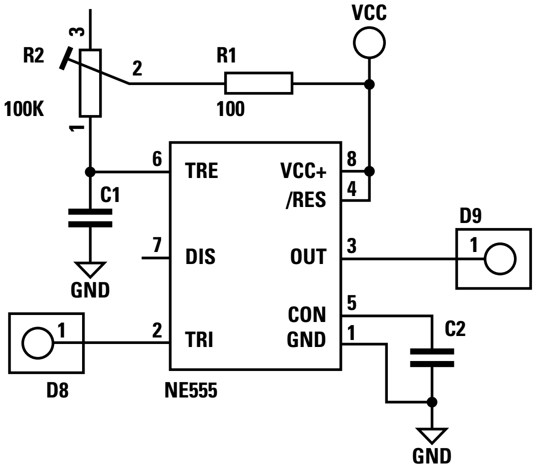


Figure 13 Circuit for reading a potentiometer position

GOING FURTHER

Here are some books and sites with more information and circuits:

- en.wikipedia.org/wiki/555_timer_IC
- *Mini Engineer's Notebook: 555 Timer IC Circuits* by Forrest M. Mims III (Radio Shack, 1984)
- *IC Timer Cookbook* by Walter G. Jung (Howard W. Sams & Co, Inc., 1977)
- *The Art of Electronics, 3rd Ed.* by Paul Horowitz and Winfield Hill (Cambridge University Press, 2015)
- *Practical Electronics for Inventors, 4th Ed.* by Paul Scherz and Simon Monk (McGraw Hill Education, 2016)

Above The 555

Potentiometer reader

The 555 was used in the Apple II family of computers to implement its joystick/paddle interface. Specifically the 558 chip, which contained four mostly independent 555s; some pins were shared to fit four of them onto a 16-pin chip. The circuits on the 558 were a derivative of the 555 that had reduced functionality, but work in monostable mode. The 558 is no longer in production and can be hard to find, so we'll build this for one potentiometer and use a 555. Since this uses software for measuring the pulse length, the ubiquitous Arduino Uno will be used.

The potentiometer is used as the timing resistor, which determines the length of the output pulse. It's read by triggering the timer and measuring how long the output is high. The length of a pulse is proportional to the value of the timing resistor, i.e. the position of the potentiometer. **Figure 13** shows the circuit, while Listing 1 shows the Arduino code.

We've looked at the 555's design and some examples of how it can be used, but that merely scratches the surface of what can be done. It's a chip that has been in production and use for close to 50 years. In that time, this little chip has been used and abused in a myriad of ways. It would take a series of books to discuss all of the applications to which it has been applied. Such a history for such a diminutive IC. □

Make your mark!

Let's learn how to mark out metalwork accurately

Right Our test piece with some accurate marks and guidelines. It's handy to practise marking out on scrap pieces before setting off on stock you want to use to make a component

Jo Hinchliffe

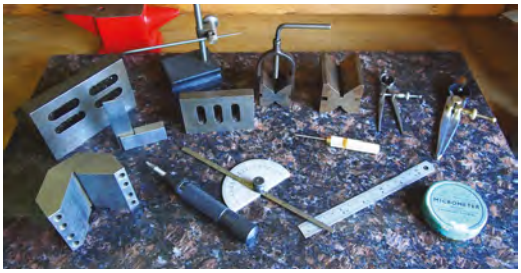
@concreted0g

Jo Hinchliffe 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.

The ability to accurately mark out metalwork for accurate machining or bench fitting (hand-cutting and filing etc.) work is a vital skill if you want to make metal components to an accurate tolerance. In this article, we will explore some of the tools and techniques used to mark out materials.

The most fundamental and useful tool in accurately marking out small metalwork is a flat surface onto which we can place the material to be marked, and on which the marking tools can move smoothly. When we are trying to make something to a high tolerance, we need the surface that we mark out from to be as flat as possible, as it is our reference point. Machining/engineering workshops usually have surface plates or surface tables. These tables are highly finished to a very flat and stable surface. They used to be made from cast iron, but it's becoming more common now for them to be made from thick and highly polished pieces of granite (**Figure 1**). However, an AAA grade granite surface plate can be expensive to buy for home use, and so a lot of home engineers use cheaper options. If you can find a thick piece of plate glass, for example, this will usually be manufactured to a highly flat surface finish, and likewise for memorial granite or even granite polished

Figure 1 A very flat, highly polished piece of granite will work well as a surface plate. A wooden cover can be made to protect the surface when not in use



STARTING FROM ZERO

The other fundamental is to work from a good-quality zero point or 'datum' on the material. If, for example, you mark out some lines for cuts or points for holes to be drilled, but you use a non-straight edge as your reference/zero/datum, then you are unlikely to gain accurate results. For instance, if we want to mark a vertical line along the centre of this stock, and then some accurately spaced points for hole drilling, if either the base or the side of the stock aren't flat then this will throw the positional accuracy. Spending time with an engineer's square and a file to ensure your material stock is square is time well spent!

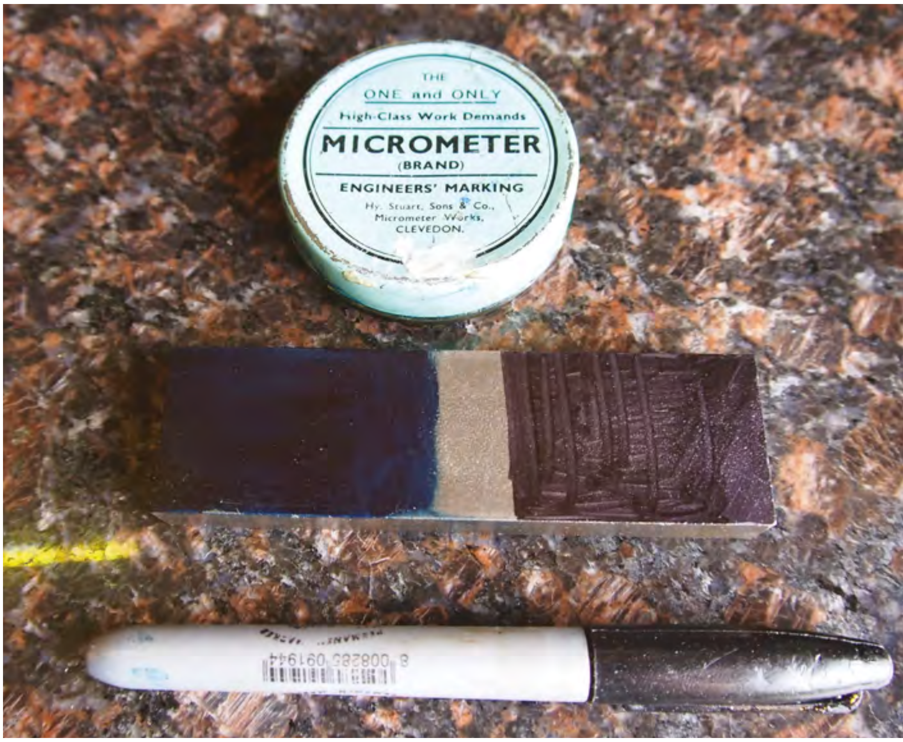


Figure 2 A tin of engineer's blue versus the cheaper and more available permanent marker

up for kitchen work surfaces. Using a known good straight edge and a very fine feeler gauge, a highly polished off-cut of granite may vary very little over its surface area (less than 0.01 mm). Plenty good enough for most needs.

FEELING BLUE?

Marking metal accurately is a fine balance in that we don't want to scratch or mark the surface to the point where it damages the finish of the final component, but also it needs to be clear enough to be seen. To help with this, it is common to cover the material stock with a fine layer of engineer's blue (**Figure 2**), which is a thick inky paste. Then, when we scribe a line, the metal is revealed in high contrast beneath the engineer's blue. Engineer's blue is common in machine shops as it's used for a variety of tasks; however, it's less common for home use.

To make the marks, a commonly used item is a scriber – a sharp tool made of hardened steel that can impart a very fine scratch onto metal. Care must be taken to keep the scriber end in good condition, and that it should taper to a fine angle so that the tip can reach into the corner when, for example, used with a ruler to draw a line. A classic accessory for a scriber is a scribing block; this allows a scriber to be held at a constant height and slide across the surface plate, allowing for straight lines of a known height to be scribed.

> "Care must be taken **to keep** the scriber end in good condition"

KEEPING THINGS VERTICAL

It's important, when working on a surface plate, to maintain accuracy when we need the workpiece to be positioned either vertically or horizontally in relation to the plate. Using vee blocks or angle plates makes this extremely simple to achieve. Vee blocks and angle plates are made to very high accuracy with precision ground surfaces so that if a piece of work is aligned with them, it ensures that the work is held at a precise angle. As seen in **Figure 3** and **Figure 4** overleaf, they have various uses in both supporting measuring tools and materials perpendicular to the surface plate. A height gauge is an excellent tool that is similar in some ways to a scribing block, but much →

YOU'LL NEED

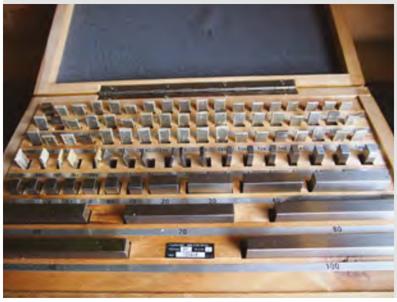
- **Very flat surface** (a piece of plate glass or granite)
- **Set of engineer's dividers**
- **Metal ruler**
- **Permanent marker**
- **Engineer's protractor**
- **Scriber**
- **Engineer's square**
- **Centre punch and hammer**

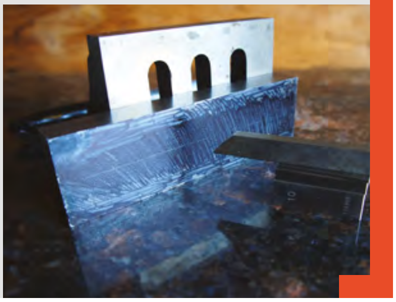
OPTIONAL

- **Scribing block**
- **Height gauge**
- **Set of slip gauges**
- **Automatic centre punch**

A BIT SLIPPY

Slip gauges can also be used for marking out, and are often used in many other areas of machining practice – they're a worthy addition to any home machine shop. Slip gauges are available in sets and are produced to a very high tolerance, often to thousandths of a millimetre. Although it isn't the best practice, a common workshop trick is that the surface finish of slip gauges are so precise that if you 'wring' them together, they will stick together. For marking out, they can be stacked to a desired height, and by using a flat-based scriber which comes to a precise point, they may be used to scribe lines at accurate coordinates. They are also used to set the travel in say a machining operation on a lathe or in a pillar drill, and can be used to gauge distances and much more.





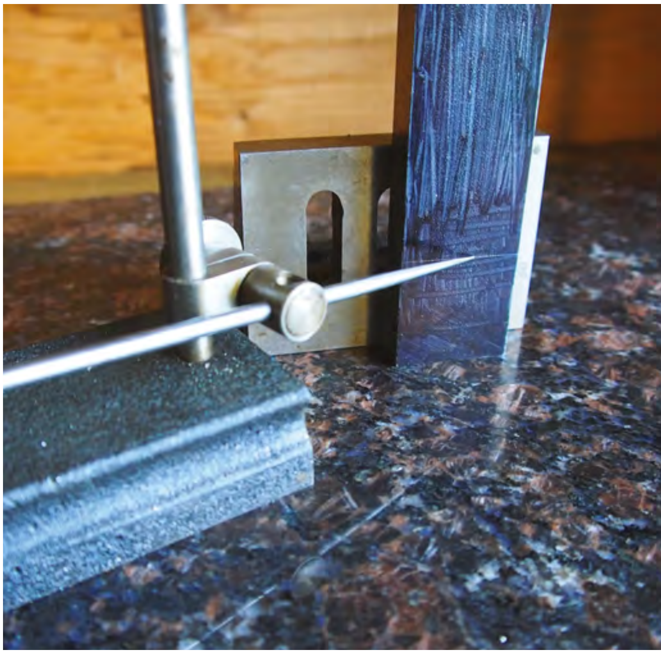
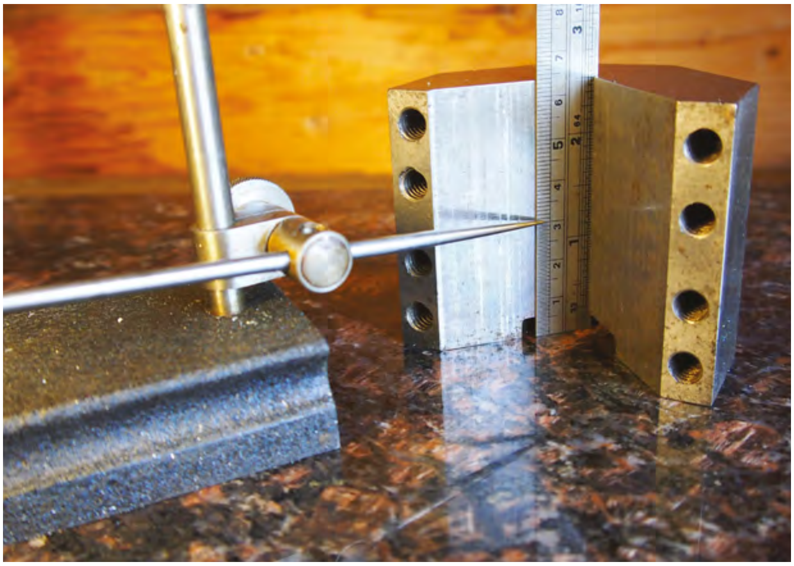


Figure 4
Scribing the line at 30 mm height onto the stock material using an angle plate to keep the stock vertical

Figure 3
A scriber held in a scribing block being set to 30 mm height using a ruler held perpendicular to the surface plate with a vee block



more accurate. Available in differing forms with analogue vernier scales, analogue dial scales, but also now with digital readouts (DROs) they represent an excellent, highly accurate method of marking out, and can also be used for many metrological checks on dimensions. Similar to the scribing block, they have a sliding base precision ground to lie extremely flush to the surface plate, and then the scribing head can be moved up and down the graduated column. The beauty of a digital height gauge is that it can be zeroed. This means, for example, that measuring the height of our stock to draw a horizontal line through the midpoint is a simple task: we zero the height gauge on the surface plate, then move it to contact the top of the metal workpiece and note the height. Dividing this height by half and moving the height gauge back down to the midpoint coordinate, we then lock it and draw the centre point line. If we wanted to add a line 3 mm above and below the centre line (to mark the position of a slot, for instance), we can then simply add the lines by moving the scriber tip to +3 mm and -3 mm from the centre line point, as seen in **Figure 5**.

Sometimes it's desirable to create equally spaced marks on a single line to, for example, create a workpiece with regularly spaced drilled holes. The first point and the centre line may be created using the height gauge or scriber block methods as above, but it's useful to mark the sequence of holes using the first hole mark/point as a datum. As the points will be drilled, it's useful as part of the drilling practice to centre-punch the hole centre mark coordinates to ensure the drills pick up exactly on the desired centre point. Using a centre punch and a hammer to create a small punch mark is a simple operation, but requires some care and practice. It's important to not risk damaging the surface plate or granite block, so centre- punching should be carried out on an anvil. Small anvils can be found for purchase, but often a small anvil area may be found on a metalwork vice.

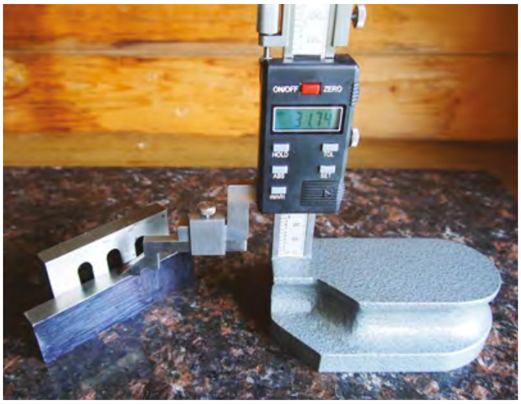


Figure 5
A height gauge is a versatile tool – it simplifies creating accurate parallel lines at known coordinates on a workpiece, as well as a host of other workshop tasks

If necessary, use a loupe or magnifier to ensure you position the centre punch accurately at the intersection of scribed lines, and then give the centre punch a firm tap to create the small indentation (**Figure 6**). Alternatively, automatic centre punches can be bought that have an adjustable spring inside which can be set to deliver a blow before the spring releases. They work well and reduce the chances of the operator bouncing the punch across the work due to an awkward hammer strike. They are also very useful if you need to punch a large number of marks, as they can deliver the same consistent blow each time, resulting in the same depth of mark.

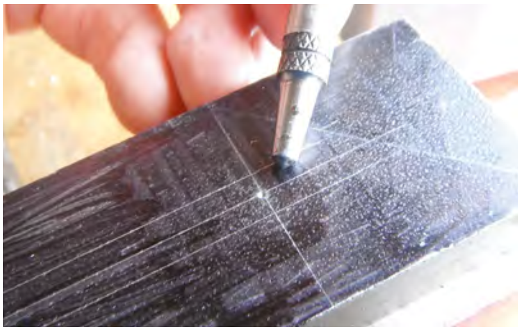
YOU SPIN ME RIGHT ROUND!

So, having punched the first mark of a regular series (**Figure 7**), it can be useful to use a classic engineering tool called 'dividers' to make further regular marks. Setting the dividers using a ruler is similar to setting a scribing block: we can insert one leg of the dividers into an indent on a rule and then lengthen the dividers, taking the other leg to the desired indentation on the rule. Then, to scribe the desired distance between the desired hole centres, we can then place one 'leg' into the mark we just centre-punched (**Figure 8**).

Rotating the other leg across the marked centre line will create the coordinate mark for the next centre punch. Repeating this will give the accurately spaced centre marks we desire.

Next month we will look at drilling and tapping operations using both hand tools and power tools, before moving on to a small toolmaking project that combines these tutorials. □

Figure 7
The first centre-punch mark made



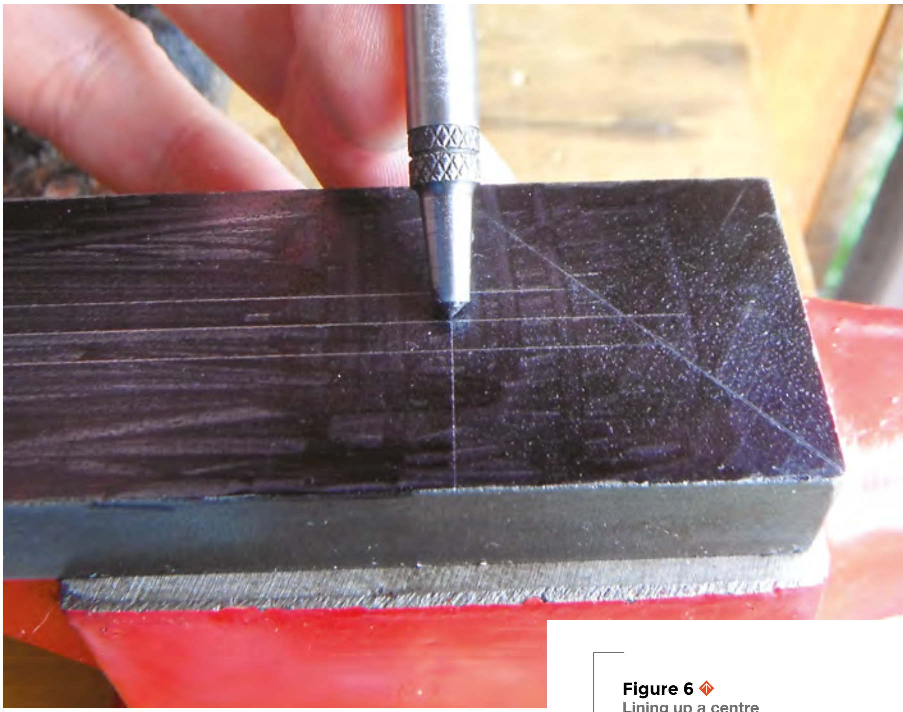
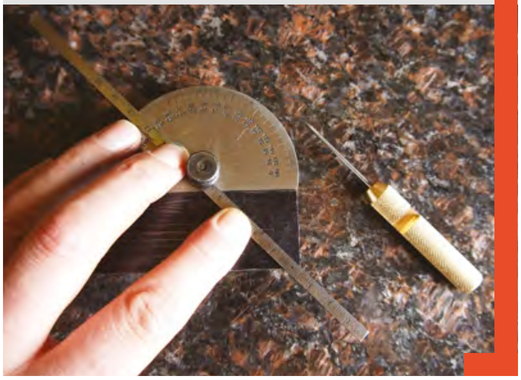


Figure 6
Lining up a centre punch takes some practice, decent lighting and, if needed, some magnification to ensure accuracy

ANGLING

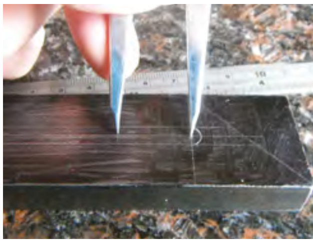
An engineer's protractor can be clamped to accurate angles and used for scribing angled lines using a scriber (the scriber pictured was made as a tutorial in issue 16) They are a useful tool for also setting angles on machines and work, and can be purchased quite cheaply.



QUICK TIP

Due to the taper on the leg of the divider, and the angle of the punch mark, the divider leg will automatically find the centre of the mark.

Figure 8
Having again used a rule to set the desired distance, dividers are a great way of creating equally spaced marks and arcs



Control games with a steering wheel

Emulate a keyboard for a custom controller

Left
We used this controller to play a racing game, but with a bit of imagination, you can take on any game you like

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.

YOU'LL NEED

Circuit Playground Express

Many microcontrollers, including the Circuit Playground Express, can emulate a class of USB peripherals known as human input devices (HIDs). These are the things that you use to control your computer – mostly mice and keyboards, but also trackballs and other controllers. Your operating system should already have drivers to handle this input, so we can send data to our computer without needing to do any programming on the computer side. What's more, this can control any game we like – there's no need to modify the game in any way in order to control it with our microcontroller. Let's take a look at how this works.

The game we will use here is Pico Racer (**hsmag.cc/DWQrXI**). It's a simple racing game, controlled using the arrow keys.

We're going to use the accelerometer to gauge the orientation of the CPX. Accelerometers measure any acceleration on a device – if it's sitting on a table, this will just be the acceleration due to gravity. It's returned as the amount of acceleration in each of the three dimensions: x, y, and z. If all the acceleration is in one dimension, that dimension will read 9.8. If the device is turned, that will be shared between more than one of the dimensions. We can get into trigonometry to work out the exact angle the device is held at, but we don't need that. For our purposes, we just need to know roughly what direction the CPX is held in as the controls are on/off rather than graduated.

This all sounds quite complicated, but you can get an idea of what sort of data we're going to get from the accelerometer by just printing out a stream of data.

```
from adafruit_circuitplayground.express import
cpx
while True:
    x, y, z = cpx.acceleration
    print((x, y, z))
```

Plug the CPX into your computer and open the Mu code editor. Enter this in the **code.py** file on the USB device (take a look at the Getting Started guide at **hsmag.cc/IoNNGG** if you've not done this before). Save this code and open the serial monitor and you should see a stream of numbers.

As you turn the CPX around, you should see these numbers change. The logic for our steering wheel is:

- If the X value is greater then 2 turn right
- If the X value is less than -2 turn left
- If the Z value is greater than 3 accelerate
- If the Z value is less than -3 brake.

Before we can get onto this, though, we need to work out how to send data to the computer. There are two modules we need to import:

```
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keycode import Keycode
```

We can use these in a few different ways, but with this game, you need to hold down keys to turn, accelerate, or brake, so it's not just key presses we need. For example, the following creates a keyboard object then presses and immediately releases the up arrow:

```
kbd = Keyboard()
kbd.press(Keycode.UP_ARROW)
kbd.release(Keycode.UP_ARROW)
```

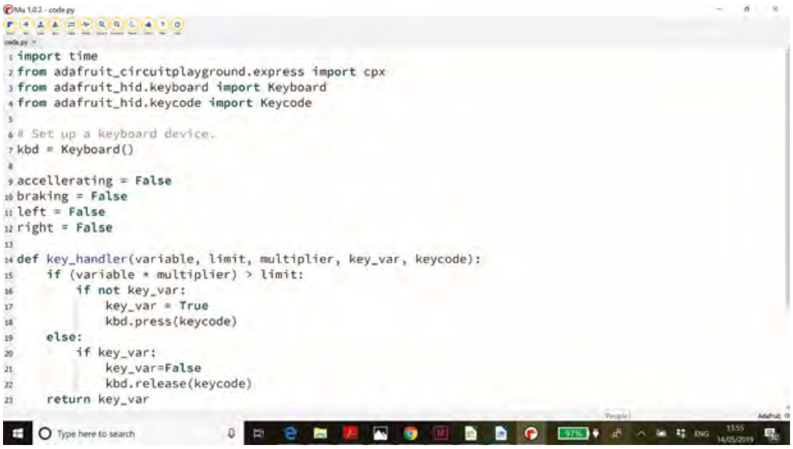
The only thing needed is a bit of code to work out what keys need pressing or releasing depending on the current orientation of the CPX. We have variables for each of the four actions that hold False if the key isn't pressed, and True if it is. We also have a function – called **`key_handler`** – that looks a little complicated, but it just stops us repeating similar code for each action. It takes five parameters; the first is the value in the x, y, or z dimension that's important to this action. The second is the amount that the value has to be larger than to trigger the action. The third is either -1 or 1 that is used to make sure we're always doing 'greater than' comparisons. The fourth is the variable that holds whether or not the key's currently being pressed, and the fifth is the key that should be pressed or released. There's then an **`if`** statement that just checks if the key should be pressed or released (or neither).

> **You'll be able to control the game by** tilting your CPX like a steering wheel

You can load this code onto your CPX, and you'll be able to control the game by tilting your CPX like a steering wheel. Be careful though: as soon as you load the code, it'll start trying to move your cursor!

```
import time
from adafruit_circuitplayground.express import cpx
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keycode import Keycode

# Set up a keyboard device.
kbd = Keyboard()

accelerating = False
braking = False
left = False
right = False

def key_handler(variable, limit, multiplier, key_
var, keycode):
    if (variable * multiplier) > limit:
        if not key_var:
            key_var = True
            kbd.press(keycode)
    else:
        if key_var:
            key_var=False
            kbd.release(keycode)
    return key_var

while True:
    x, y, z = cpx.acceleration
    print((x, y, z))
    accelerating = key_handler(z, 3, 1,
accelerating, Keycode.UP_ARROW)

    braking = key_handler(z, 3, -1, braking,
Keycode.DOWN_ARROW)

    right = key_handler(x, 2, 1, right, Keycode.
RIGHT_ARROW)

    left = key_handler(x, 2, -1, left, Keycode.
LEFT_ARROW)

    print(accelerating, braking, left, right)
```

Below
Create your own games controller with just 32 lines of CircuitPython



```
import time
from adafruit_circuitplayground.express import cpx
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keycode import Keycode

# Set up a keyboard device.
kbd = Keyboard()

accellerating = False
braking = False
left = False
right = False

def key_handler(variable, limit, multiplier, key_var, keycode):
    if (variable * multiplier) > limit:
        if not key_var:
            key_var = True
            kbd.press(keycode)
    else:
        if key_var:
            key_var=False
            kbd.release(keycode)
    return key_var
```

How to make a cassette synthesizer

By altering the speed of a cassette player's motor, you can turn it into a musical instrument

Matt Bradshaw

@nerdonthewire

Matt Bradshaw is a maker and musician from Oxford. He builds instruments to play with his band, Robot Swans. You can find more of his projects at **mattbradshawdesign.com**

Magnetic tape can be used to record and play music, but there is also a rich history of musicians using this technology to create music. In this tutorial, you will learn how to convert an old cassette player into a musical instrument capable of playing ethereal melodies.

Anyone over the age of about 30 will remember cassette tapes. They were hugely popular in the 1970s and 1980s, largely because you could record onto them. Their popularity waned in the 1990s as CDs became more popular, but this means there are now a lot of cassette players lying unused in people's attics, ready to be hacked.

An old cassette player can be turned into a musical instrument by controlling the speed of the motor. Because of the way that sound works, halving the speed of the tape player's motor will lower the pitch by an octave, and the speeds in between will correspond to other notes. This isn't very helpful if you're listening to an album, but if you record one long, constant note onto a cassette, it becomes possible to play a melody simply by altering the motor's speed.

Right Keep track of which screws go where, particularly if they are different lengths

CHOOSE YOUR WEAPON

This project should work with a wide variety of cassette players, but it's worth having a think about exactly what type of player you want to use. There are many types of cassette player, and there are no right answers as to which one is best for this hack, but for this tutorial we've used a Sony TCM-939.

This is a sensible choice for a number of reasons. It is quite a chunky design, making it easy to hack. It has a built-in speaker, which is good for live music performance. It has the ability to record as well as play, and features both a built-in microphone and an audio input jack. Crucially, it can also be powered by either batteries or a 6V DC power supply, making it nice and safe. There are plenty of similar designs which will work just as well, so have a look around charity shops and on eBay, but feel free to experiment – it's even possible to do this project with a Walkman.

Once you've found a suitable cassette player, you need to become familiar with it. First, try out all its functions – if you know that everything works to start with, it'll save you a lot of time troubleshooting later on. Try playing a couple of different tapes. Try pausing, rewinding, fast-forwarding, stopping, and ejecting. Try recording a few different sounds through the mic and the input to see if the quality is OK. Try both the speaker and the headphone output. Check that the volume control works. This all sounds very basic, but it's definitely worth doing.

WHAT LIES BENEATH

Next, take a look inside. Use a screwdriver to open the case, and examine everything closely. You should be able to see a circuit board, a selection of gears and belts, and a motor. The mechanical workings of a cassette player are pretty complex and fiddly, so it's best to leave them untouched, although it's interesting to press 'play' with the innards visible,

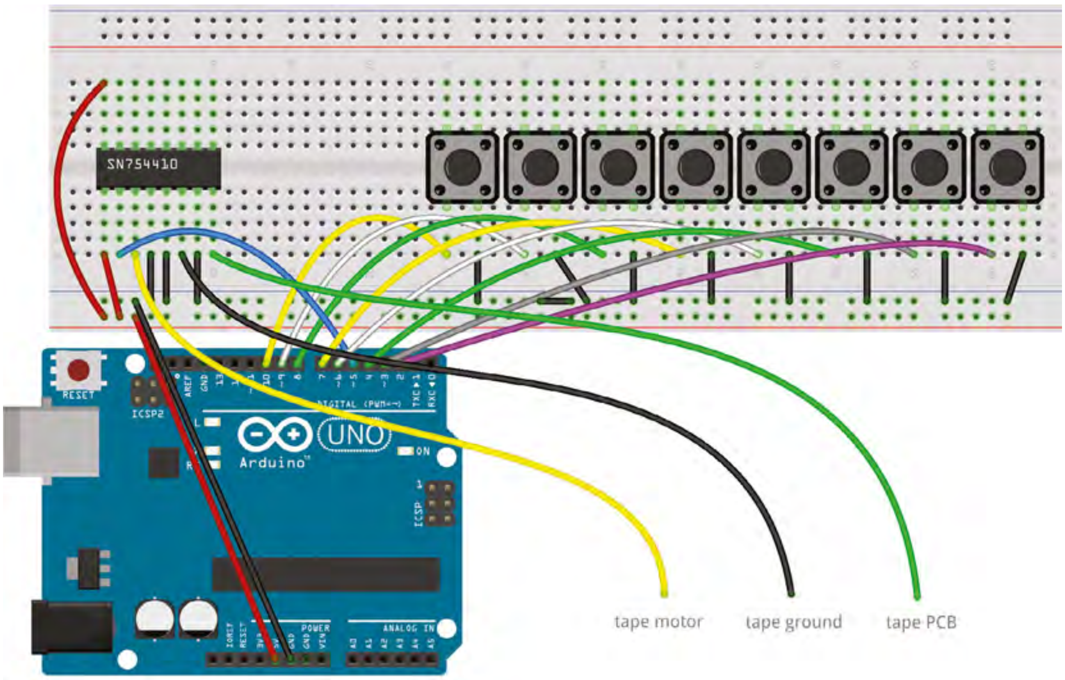


Figure 1 The wires that you add to the cassette player connect to a motor driver chip on the breadboard, which is controlled by an Arduino

in order to get a better understanding of how the gearing works.

To execute our musical hack, we need to interrupt the signal between the circuit board and the motor. If you're lucky, there will be two obvious, accessible, colour-coded wires going to the motor: a power wire (often red) and a ground wire (often black or white). Cut these wires in half, taking care to leave a decent length of wire going to both the motor and the circuit board if possible, then strip about 1 cm of the plastic covering to reveal the wire underneath. You should now have four exposed wires.

> " **The mechanical workings of a cassette player** are pretty complex "

In order to connect the tape machine to a breadboard, we want to end up with three long-ish wires protruding from the back of the machine: a ground wire, a wire from the circuit board, and a wire to the motor. Cut three lengths of single-core wire, each about 20 cm long, making sure they are different colours so you can identify them easily (we've used black for ground, green for the circuit board, and yellow for the motor). →

THE **BEGINNINGS**

One famous instrument to use a similar principle to this is the Mellotron, a keyboard from the 1960s which Paul McCartney describes as "a forerunner of the synthesizer". Like our instrument, different recordings could be used to simulate different instruments, but unlike our instrument, it had a different length of tape for each key, allowing a musician to play chords. You can hear the Mellotron on epic tracks such as The Beatles' *Strawberry Fields* (the flute sound in the intro) and Radiohead's *Paranoid Android* (the choir sound in the breakdown).

YOU'LL NEED

- **A suitable cassette player** (such as Sony TCM-939)
- **A music cassette for testing**
- **A blank cassette**
- **Arduino Uno**
- **Breadboard**
- **Jumper wires**
- **Single-core wire or crocodile leads**
- **Insulating tape or heat-shrink tubing**
- **SN754410 motor driver chip**
- **Eight large tactile buttons**
- **Wire strippers**
- **Soldering equipment** (optional)

Above The Sony TCM-939 is a good candidate for hacking

First, twist the cassette player's two ground wires back together, then solder these two wires to your long ground wire, giving a Y-shaped connection. Next, solder the new long circuit board wire to the relevant wire in the tape machine, and do the same for the motor wire. Cover the connections with electrical tape or heat-shrink tubing to prevent short circuits.

ON AND OFF

We now have access to the signal that the cassette player's circuit board sends when the play button is pushed, and we have separate access to the power input of the motor. This means that we have already executed a simple hack, in that we can turn the motor on and off at will. Try this by pressing 'play', leaving the ground wire unconnected for now, and touch the ends of the signal and motor wires together. The motor should run when the wires are touching and stop when you break the connection.

Getting the Arduino to change the speed of the motor is a bit trickier. The Arduino can't provide enough current to power the motor on its own, and besides, the voltage is wrong, so we want to use the power coming from the cassette player's circuit board. To do this, we're going to use a chip called the SN754410. This integrated circuit is known as a 'quadruple half H-bridge', which means it can either control four motors going in a single direction, or two motors going both forwards and backwards. It's a little bit overkill for our needs, since we're only aiming to power one motor in one direction, but it's a nice simple way to control the motor. The chip allows a 5 V signal from the Arduino to switch the motor on and off using the power from the cassette player.

To vary the speed of a motor like this, you don't tend to alter the voltage. Instead, you send a series of quick pulses to the motor, essentially turning it on and off very fast. Varying the length of these pulses affects the speed of the motor, which is why this technique is called pulse-width modulation, or PWM – it's worth understanding this technique, so have a look for some information about PWM on the web if you haven't come across it before.

LET'S GO

Build the breadboard circuit as shown in **Figure 1** on the previous page. The eight buttons can be read individually by the Arduino, and will serve as a miniature piano keyboard, representing a full octave of notes. Pin 5 of the Arduino is connected to the motor driver chip, and will send a PWM signal which controls the speed of the motor.

Before we start using the buttons, let's try uploading a simple sketch to make sure the motor driver is working properly. The code (overleaf) sets pin 5 as an output and sets the motor power at 255, which is 100%. Upload this code to the Arduino, press 'play' on the tape machine, and the tape should play as normal.

QUICK TIP

If you don't like soldering, you can join the wires using screw terminals, or even just use crocodile clips.

GOING FURTHER

This tutorial describes one method of generating melodies from magnetic tape, but there are lots of other ways to achieve the same thing, each with its pros and cons. A more precise approach, for instance, might involve replacing the existing motor with a stepper motor. You would probably need to do some tricky modifications to the gear system, but the upshot might be a better-sounding instrument with a bigger range. Try building this design first – if it interests you enough, you can always try a different, more ambitious method next time.

```
void setup() {
  pinMode(5, OUTPUT); // set pin 5 as output
  TCCR0B = TCCR0B & 0b11111000 | 0x01; //
increase PWM frequency
}

void loop() {
  analogWrite(5, 255); // set motor speed (PWM
duty cycle)
}
```

Try re-uploading the code with a few other values between 0 and 255, and you should hear the tape play at a lower speed. Below a certain value, the motor probably won't work at all – try not to leave the motor in this state for long in case it causes damage (you can always press 'stop' to kill all power to the motor).

TUNING UP

By now you should be able to digitally control the speed at which the tape is played, so it's time for your machine to level-up from cassette player to musical instrument. Find a blank tape, rewind it to the beginning, and hit 'record' (if you're hacking a machine without a record function, you'll obviously have to do this with a separate cassette recorder).

Now you want to record one long, unwavering note, which will serve as the base sound source. Don't worry too much about it sounding great at the moment, this is just for tuning the instrument. It doesn't matter exactly what note you play, but a high note is easier to calibrate the instrument with than a bass note. You can play the note with any instrument that will reliably stay in tune for a few minutes – an organ, an open violin string, or even a single piano note played repeatedly would be fine. A pure tone from a synthesizer (or synth app) would be ideal. The longer you can be bothered to record the note for, the better – about three minutes is a good starting point.

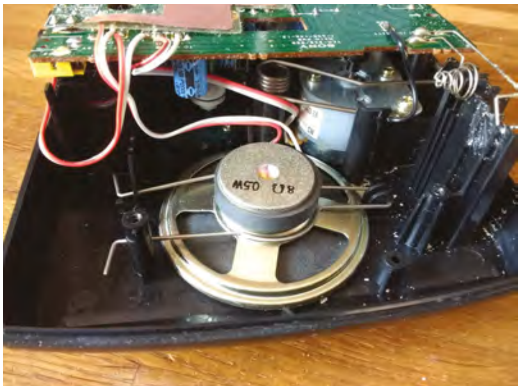
Once you've recorded a few minutes of a single note, rewind the tape. Before you can start playing melodies with your instrument, you'll need to know which **analogWrite** values will produce which note. A value of 255 (the highest possible value) will give the highest note, and lower values will give lower notes, but the relationship is non-linear and unpredictable, so a bit of trial and error is required. →

TIMING TROUBLES

There is a line of code in the sketch (beginning `TCCR0B =` ...) which looks a little bit odd. This line alters the Arduino's internal workings and increases the PWM frequency. You can try the sketch without this line, but (depending on the tape machine) you might hear an unpleasant buzzing sound through the speakers. This occurs because the motor is being switched on and off at a frequency within the human range of hearing. Raising the PWM frequency above the speaker's audio range fixes the problem. The only issue with doing this is that it also affects the Arduino's sense of time – if you later decide to add time-based functions like **delay** or **millis** to your sketch, you will have to multiply or divide the results by 64 to get the intended outcome. To learn more about what's going on, try searching the web for something like 'Arduino timer prescaler'.

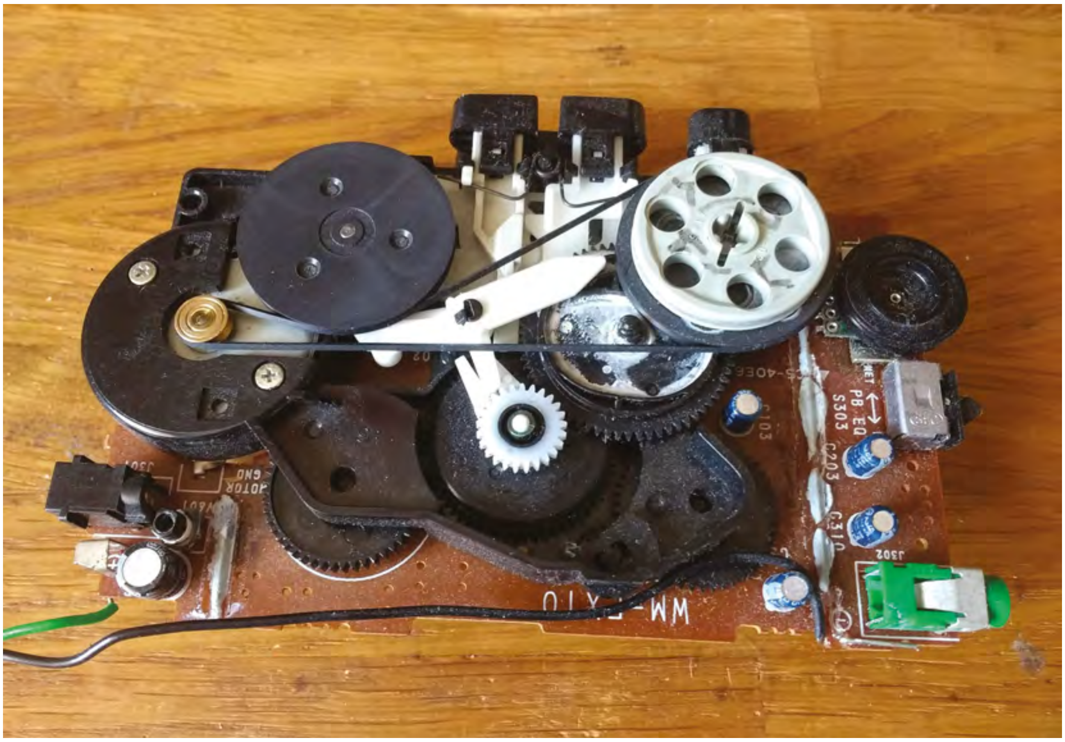
QUICK TIP

To get a 'sped-up' rather than 'slowed-down' sound, record your long note at half speed by pressing the lowest button before hitting 'record'



Above
The motor can be found behind the speaker, under the PCB

Below
If you're lucky, the tape machine should come apart easily



Right As an example of an alternative approach, this hacked Walkman was driven with a Lego motor

Try re-uploading the sketch with different values in the **`analogWrite`** function, and make a note of which notes they produce. Unless you've got perfect pitch, you'll need a musical instrument or possibly a guitar tuner to help you identify each note, bearing in mind that only a few values will produce notes that are in tune, rather than sharp or flat. As a rough guide to get you started, some typical values for the highest five notes (semitones) might be 255, 138, 123, 116, and 110, but it will be different for every cassette player.

With the TCM-939 it's possible to get at least an octave of notes (i.e. a full major scale, or 13 semitones), but the range will depend on the type of cassette player and motor. The higher notes are likely to sound better than the lower notes, because the motor is being underpowered to produce the lower notes and is therefore likely to be a bit wobbly.

QUICK TIP

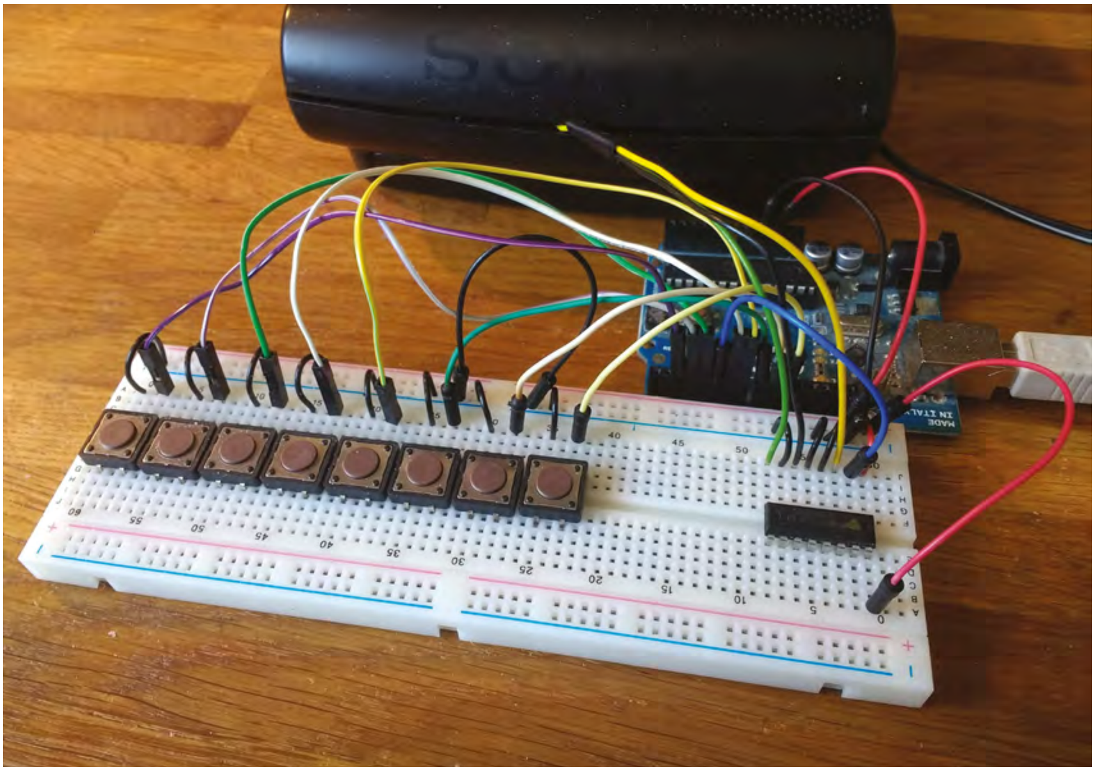
Only certain Arduino Uno pins are capable of pulse-width modulation, and pin 5 has the highest maximum frequency, which is best for this project.

IT'S PLAY TIME

Once you've made a note of the tuning data, you can upload the main sketch, shown opposite, making sure to substitute your own tuning values into the **`notes`** array. In this sketch, the buttons are initialised as **`INPUT_PULLUP`**, meaning that the pins default to 'high' using the Arduino's internal pull-up resistors. When a button is pushed, its pin is connected to ground, and its reading becomes 'low'.

When a button push is detected, the sketch outputs the relevant PWM signal to the motor driver chip, using **`analogWrite`**. Only eight of the 13 possible **`note`** values are used, creating a major scale.

```
const int buttonPins[] = {2,3,4,6,7,8,9,10};
const int notes[] = {90,92,94,96,98,101,104,107,1
10,116,123,138,255};

void setup() {
  pinMode(5, OUTPUT);
  for(int i=0; i<8; i++) {
    pinMode(buttonPins[i], INPUT_PULLUP);
  }
  TCCR0B = TCCR0B & 0b11111000 | 0x01;
  analogWrite(5, notes[0]);
}

void loop() {
  if(!digitalRead(buttonPins[7])) analogWrite(5,
notes[12]);
  else if(!digitalRead(buttonPins[6]))
analogWrite(5, notes[11]);
  else if(!digitalRead(buttonPins[5]))
analogWrite(5, notes[9]);
  else if(!digitalRead(buttonPins[4]))
```



QUICK TIP

Keep track of which screws go where, particularly if they are different lengths.

Left Using larger buttons makes the instrument easier to play

```
analogWrite(5, notes[7]);
  else if(!digitalRead(buttonPins[3]))
analogWrite(5, notes[5]);
  else if(!digitalRead(buttonPins[2]))
analogWrite(5, notes[4]);
  else if(!digitalRead(buttonPins[1]))
analogWrite(5, notes[2]);
  else if(!digitalRead(buttonPins[0]))
analogWrite(5, notes[0]);
}
```

If all is well, you should now have a working instrument. Try playing a few melodies to get the feel of it. The choice of single-note base sound is important, so try recording a few different ones. Experiment with weird combinations of instruments, or sounds that evolve as the tape plays – there are no rules!

There are loads of ways for you to improve the project from here. You could design a 'gate' circuit so that the audio is only heard when you are holding down a button, giving you more control over the sound. You could add a second or third tape machine and create a polyphonic instrument. You could add a MIDI input and control your instrument from a computer or keyboard.

> **Experiment with weird combinations of instruments, or sounds that evolve as the tape plays** – there are no rules!

Possibly the most useful next step, though, might be to find a way to make the instrument more permanent, perhaps by moving the breadboard circuit to stripboard, mounting the whole thing on a piece of wood, or even squeezing your extra circuitry inside the existing case of the cassette player. It's up to you! □

LOOPING AROUND

Cassettes can hold quite a bit of audio, up to an hour per side, but you'll still likely find yourself having to stop and rewind quite a lot while you're playing around. One way of avoiding this issue is to use an endless tape loop, where the two ends of the magnetic tape are joined together, meaning any sound recorded to the tape will loop forever. There are numerous online tutorials on creating your own basic tape loop cassettes, or you can buy long, ready-made ones from companies such as Tapeline.

Making a coin ring

Make jewellery with currency and fire

Mark Simpson

Mark is a generalist maker who hammers, solders, welds, and saws his way to fantastic creations.

The most important part of picking your coin is knowing what metal it's made from. For our process, it needs to be soft enough to work with hand tools and annealable – this means that we can use heat-treatment to soften it. The two most common metals in coins that work are copper and silver. Aluminium also works, but the only common aluminium coins are too small for our use.

Silver is a bit expensive for learning with – though can work well once you've mastered the skills – so that leaves us with copper. In the UK, 1p and 2p coins are both copper-coloured, but have been made of steel since 1992. Since the 1p is too small, we'll work with a pre-1992 2p coin. These are still commonly in circulation, so should be available.

Drilling British coins is technically illegal in the UK, so while 2p coins are a great choice if you're not based in the UK (this author is US-based), anyone in the UK wishing to follow the law will need to use a different country's currency. Pre-1989 Irish 2p coins should also work. If you can't get an appropriate coin, you can cut a disc out of sheet metal.

The process is, in principle, very simple. We'll drill a hole in the middle of our coin, then fold the sides up to make a ring. Let's start with the hole in the middle. Sizing the ring is a bit haphazard, and you'll probably need a bit of practice to get it right as the hole will stretch as we fold the size up, and the amount it stretches will depend on the particular mandrill and hammer, as well as your techniques. Start by drilling a hole smaller than your

ALUMINIUM SHEET

Aluminium is softer than copper, and easily worked with hand tools. With appropriate cutting tools (we used a Dremel and a series of straight cuts), it's easy to shape a 'coin' to start with. It doesn't even have to be circular, so you can be a bit more creative.

The one risk with annealing aluminium is accidentally melting it. To avoid this, you can apply a little bar soap to the ring, and once this turns brown, the aluminium will be hot enough. Unlike copper, you need to let it cool slowly, so leave it to air cool until it's cold enough to handle.

Above Drilling will leave a rough edge that you need to file off

Left The basic tools you'll need: a mallet and mandrill

Far Left Cut and stamped aluminium, ready to be made into a ring if you'd rather not use a coin

finger in the centre of a coin. You'll need to centre-punch the middle of the coin and then mount the coin securely in a vice (the coin will try to spin as you drill it and it can be tricky to stop it from doing this). We used a stepper drill bit which allowed us to experiment with different sizes, but any appropriately sized metal drill will do.

Once you've got the hole in the middle, it's time to start working the edge round. While copper is quite soft, it'll be too hard to work initially so you'll need to anneal it. This is simply the process of heating and cooling the metal to soften it. As you work the metal, it'll gradually get harder, so you'll need to reanneal intermittently as you work the metal.

Place the coin on a heat-proof surface (such as a fire brick), and blow-torch it until it glows a dull orange. Use pliers to pick up the coin and quench it in water until it's cool enough to handle. You can now slot the coin onto the mandrill and use the hammer to tap the side of the coin over.

Keep tapping and rotating the coin until you feel it harden, and then reanneal it. It can help if you put the ring a different way up on the mandrill each time. After a few cycles, you should have the side of the ring folded over. The more you hammer away at it, the larger the ring will be, so if it's a little small, you might want to keep working it, while if it's the right size, you can stop.

> "After a few cycles, you should **have the side of the ring folded over**"

At this point, the basic ring is almost finished. You'll need to tidy up the drill-hole by using a file and sandpaper to remove any burrs. Beyond this, what you choose to do with the ring is purely down to aesthetics. You can leave the imprint of the coin on, you can make it stand out with a patina, or you can sand it off and put on whatever surface finish you like, from hammered texture to highly-polished copper. □

SAFETY

Blow-torches are hot, as are things that have had blow-torches pointing at them. Be careful and make sure you've got some water handy in case you do accidentally pick something up too soon. It's always a good idea to wear protective goggles when working with metal, and this is especially true when the metal is hot.

Remove rust from old tools

Turn forgotten tools into workshop favourites

Dr. Andrew Lewis

@monkeysailor

Dr. Andrew Lewis is the owner of **Shedlandia.com**, a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.

When rust sets into an old tool, lots of people just buy a new one, or grab the nearest brush and tin of rust-stabilising paint to smother the affected part. There are better ways of dealing with the problem of rust and, in this tutorial, we'll look at some of them.

If you discount expensive lasers and electrolysis, there are two main categories of rust removal. Chemical removal uses acids or other chemicals to etch the oxide layer away, or stabilise it so that it doesn't spread any further. Mechanical methods of rust removal involve physically attacking the rust with abrasives to scrub back the rust until clean metal is revealed. The method of rust removal you will use depends on the object you are trying to restore, the time you have available, the amount of money you're willing to spend, and the amount of rust that you are dealing with.

MECHANICAL REMOVAL

Mechanical rust removal is usually messy, can be noisy, and typically requires a fair amount of physical strength and personal protection equipment to do it safely. The advantage of mechanical rust removal is that it can be very fast, and very thorough. If you need to clean a tool quickly, this is the best way to proceed.

Lightly rusted tools are the easiest to deal with, and can be brought back into condition in a few minutes with no personal protection equipment. To restore a lightly rusted tool, you simply spray the rusted parts with a water-repelling oil, and use an ordinary green kitchen scourer to rub the rust away. Wipe any residue away with a soft rag, and the tool is as good as new. It helps if you spray the scouring pad with oil to keep the surfaces lubricated while you are working. If the rust is a little bit more stubborn, and doesn't yield to the green scouring pad, then you can use a contour sanding pad (also known as a Scotch-Brite pad), or wire wool to be a little bit more aggressive.

The key to removing rust is to use the minimum amount of abrasive possible to do the job. The more abrasive a substance you use, the more chance you have of scratching good metal away with the rust. Most brands of abrasive pad use a colour-code to indicate their grade, with dark green being very coarse, light green being fine, and white classed as a non-abrasive pad.

Right A neglected vice gets the benefit of a wire wheel

Far right Some basic supplies like light oil, a few abrasive pads, and some vinegar are enough to tackle most restoration projects

Left
Rusty tools abound in sheds and car boot sales across the land

Below
This leg vice has been protected with a layer of Danish Oil, which will prevent it from rusting in the future. The screw of the vice has been coated with water-repellent grease

If your tool has rusted to the point that the surface of the tool is scaly or pitted, then you will need to take more aggressive action to remove the rust. You can use a wire brush to remove scale and loose rust. If the rust is exceptionally bad, a welder's spring hammer might be useful to gently tap off the worst of the scale before switching to a wire brush. The type of wire brush that you will use depends on the tool that you're working on, and is something that you should consider carefully. Brass brushes are softer than steel brushes, and are less likely to damage the surface of the metal or leave scratches behind. However, because they are soft, they also wear out quickly.

Removing heavy rust by hand is very time-consuming. It's much quicker to use an angle grinder with a cup brush or wire wheel to remove deeply ingrained oxidation. Just as with a hand brush, the type of brush that you use will depend on the hardness of the tool you're working on. Hard metals can be worked with a twisted steel brush, while softer metals will do better with a straight wire or cup brush. When working with an angle grinder, do not underestimate the danger, noise, and mess that the tool will create. You do not need to apply a lot of pressure with the grinder – you are trying to remove the rust, rather than the underlying metal.

I recommend working outside while wearing goggles, gloves, a good-quality filter mask, a leather apron, thick trousers, sturdy shoes, and a well-fitting hat when you are using a wire brush in an angle grinder. Cup brushes and wire wheels cast off metal bristles with alarming regularity, and with enough force to penetrate skin. The angle grinder also kicks lots of metal dust into the air while you are brushing at the rust, and this is not something that you want floating around indoors.

Brass brushes are available for angle grinders, but the speed of an angle grinder means that they will wear out very quickly. If you are considering using a brass brush in your angle grinder, then you might be better switching to an electric drill with a →

YOU'LL NEED

- **Light oil** (WD40 or similar)
- **Scouring pads or wire wool**
- **Wire brushes in various sizes**
- **Angle grinder or electric drill** (optional)
- **Rotary wire brushes** (optional)
- **Vinegar or Evapo-Rust**
- **A plastic container with a lid**
- **Ultrasonic cleaner** (optional)

LOOKING NICE

Removing all of the rust from an old tool can make it look less aesthetically pleasing. If you're restoring a tool for use, this isn't really a problem because it'll soon get dirty and develop a patina. You can help things along (and add an additional layer of protection) by wiping a thin layer of dark Danish Oil over the tool. The dark oil wash will not only dry into a protective layer, but it will also 'weather' cracks and imperfections to make the details of the tool more visible.

Above
Wire wheels can make short work of rust, but should be used with caution

> " Soaking a hammer with a wooden handle in vinegar **is likely to result in a strange desire for fish and chips every time you knock nails in** "

wire brush instead. The lower rpm of the drill will be more forgiving to the tool, and the brush will be less likely to cast off metal wires at lower speeds. Brass brushes can be useful to use when a tool has a wooden handle. You can tape up the handle with masking tape for extra protection, and use the brass brush in the areas nearest to the wood to reduce the chance of gouging chunks out of the handle.

For less bulky items, you can use a bench grinder with a wire brush fitted. A bench grinder can be less messy, and has most of the wire wheel enclosed inside a metal guard, and some can have a dust extractor fitted. If you can get your hands on a bench grinder, this is a very good solution for cleaning screwdrivers, tinsmith irons, and other small items.

QUICK TIP

Use gaffer tape, wrapped with the sticky side out around a magnet, to collect stray wires from a wire brush. The tape will hold on to the wires when you remove the magnet.

SUPER-**SIZED**

Not every tool is small enough to dunk into a vat of chemicals. If you have something large or immovable, like a vice or a metal frame, you can soak paper towels in vinegar or Evapo-Rust, and apply them to the surface that you need to treat. Cover the paper towels in plastic, and hold them in place with cable ties, magnets, or any other method you can think of.

Above
Some vinegar, or Evapo-Rust, and a plastic tub is all you need to remove rust chemically. It's also handy to have some protective gloves for handling the items while they're wet

CHEMICAL BROTHERS

If you aren't in a rush, chemical rust removal can be a low-fuss way to restore tools. Although there are many different professional formulations for removing rust, the most common and widely available chemical solution is probably ordinary vinegar. To remove rust from an iron tool, soak it in a plastic tub of vinegar for 24 hours or more. It doesn't matter whether you use malt or distilled vinegar, although distilled vinegar has the advantage of being transparent. The exact time that you soak the tool will depend on how rusty it is, how warm the vinegar is, and how many times the vinegar has been used already. Remove the tool from the vinegar, and wash it in water with a stiff scrubbing brush. The rust will wash away into the water, leaving clean metal behind.

Unfortunately, there are some limitations to using vinegar, and the most noticeable of these is the strong smell, which will permeate into any porous surfaces. Soaking a hammer with a wooden handle in vinegar is likely to result in a strange desire for fish and chips every time you knock nails in. Another big limitation is that tools will start to rust again almost immediately (literally within seconds of being dried) unless you protect them with grease or oil. For a single tool, this is not a significant issue, but if you are restoring several tools at once, you might find that you need to work quickly to avoid wasted effort.

Some of the more expensive rust removing solutions are odourless, but you should still take care with the product you use. Many rust removers

Above
Evapo-Rust and vinegar give similar results on metal, as you can see from these before and after photographs. Half were soaked in vinegar, while other others in Evapo-Rust

Left
The rust can be completely removed, but will leave a mottled imprint on the metal

Below
Angle grinders are loud, messy, and can be dangerous if used improperly, but they are quick

use harsh acids and toxic chemicals that need to be carefully disposed of, and might be difficult to work with from a safety or environmental perspective. An exception to this is Evapo-Rust, which is much less harmful, and works as a chelating agent rather than an acid. You can use Evapo-Rust just as though you were using vinegar, by dipping the tools into a plastic container of Evapo-Rust and leaving them submerged for 24 hours.

If you can't afford to wait 24 hours, there are a few ways to speed up the chemical cleaning process. Increasing the temperature of your solution will increase the rate of reaction, and agitation of the part will also yield quicker results. If you have access to a heated ultrasonic cleaner, you can add vinegar or Evapo-Rust to it and clean small parts very quickly.

In summary, use mechanical methods if speed is a factor, and chemical methods if you want to use the minimum effort. Use the minimum amount of abrasive that you can, try to use environmentally friendly chemicals, and dispose of any waste appropriately. You might be able to get faster results if you use electrolysis, and you definitely will get faster results if you use lasers. However, since these methods are either potentially explosive, poisonous, or incredibly expensive – it's better to use the techniques described here unless you know exactly what you're doing. □

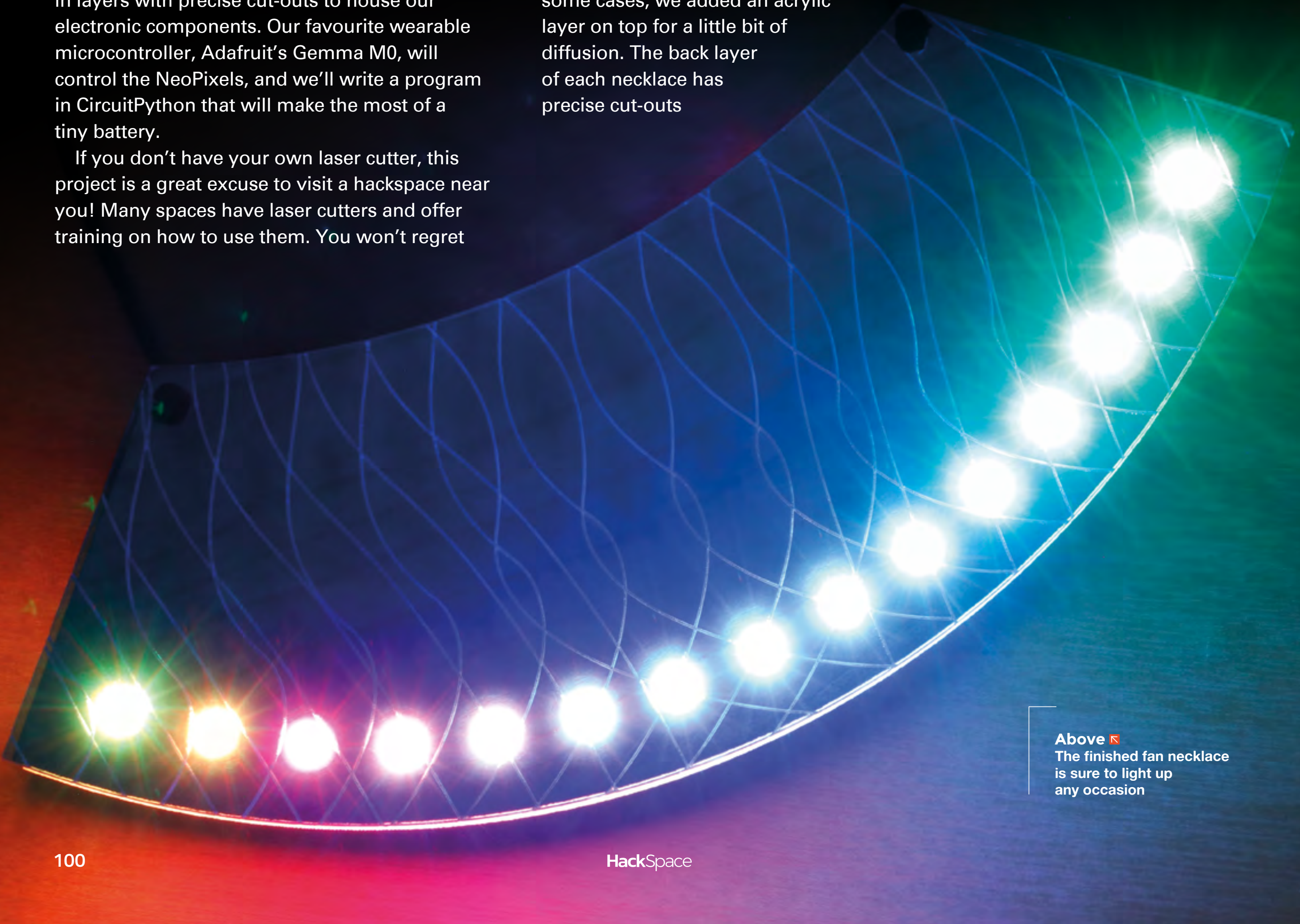
Laser-cut NeoPixel necklace

Create a stunning piece of jewellery that really shines

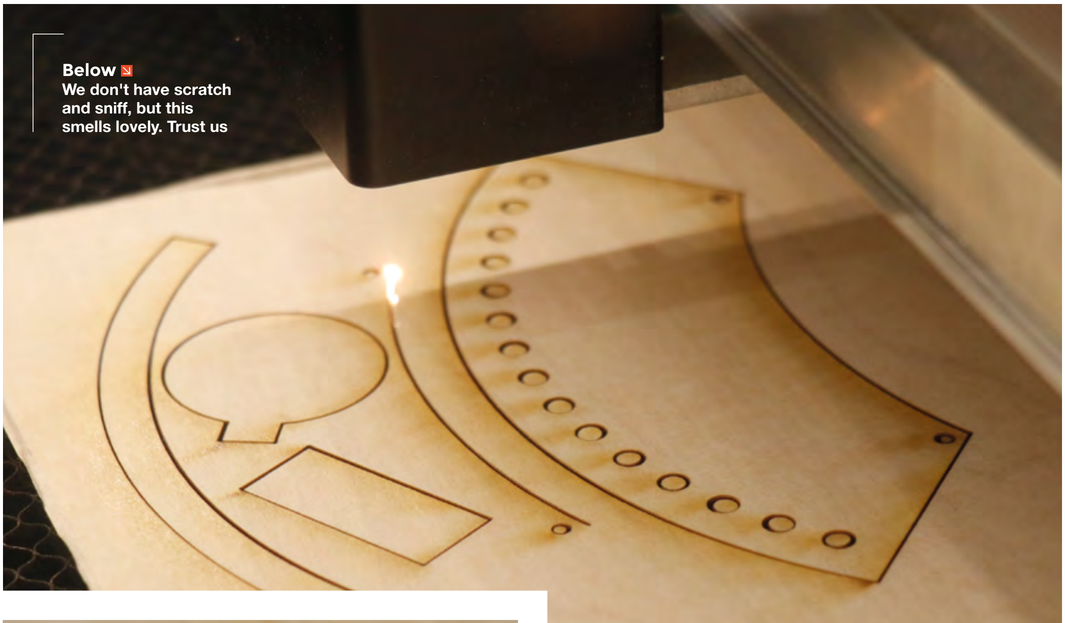
Jewellery is meant to be shiny, so it's the perfect place for NeoPixels! The trick is: how to get everything you need for a NeoPixel circuit to fit inside a small piece like a pendant. Enter the laser cutter! Laser cutters are great for precision cuts in materials like thin wood and acrylic. In this project, we'll use a laser cutter to create a sleek necklace that shines bright with NeoPixels from Adafruit. We'll build the necklace in layers with precise cut-outs to house our electronic components. Our favourite wearable microcontroller, Adafruit's Gemma M0, will control the NeoPixels, and we'll write a program in CircuitPython that will make the most of a tiny battery.

If you don't have your own laser cutter, this project is a great excuse to visit a hackspace near you! Many spaces have laser cutters and offer training on how to use them. You won't regret adding it to your skill set, and you'll be supporting your local hackspace too. If you simply can't access a laser cutter on your own, try sending the files out to a service like Ponoko. Or, take this concept of layering and apply it to materials you can cut by hand, like balsa wood, foam, or even cardboard.

Each of our necklace designs consists of several layers. All have a layer that hides the electronics but shows the NeoPixels through tiny holes that align perfectly with each LED. In some cases, we added an acrylic layer on top for a little bit of diffusion. The back layer of each necklace has precise cut-outs



Above The finished fan necklace is sure to light up any occasion

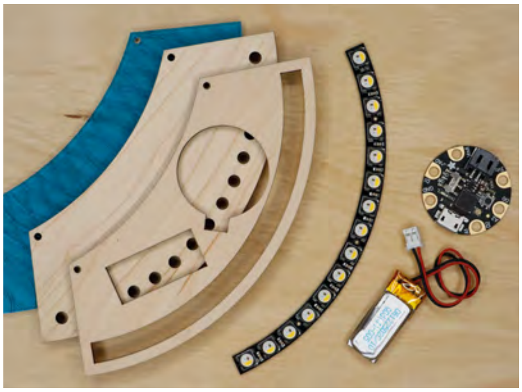


Below
We don't have scratch and sniff, but this smells lovely. Trust us

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**



Above
Ready for assembly

to hold the NeoPixels, microcontroller, and battery in place. The NeoPixels face the front of the necklace, and the Gemma M0 faces the back, making it easy to access the on/off switch while wearing. This tutorial will step you through the build process for the fan necklace, which illustrates this layering concept. Then, you can use what you've learned to make the other necklaces, or design your own!

The arc of NeoPixels in our fan necklace is actually one quarter of a giant ring of NeoPixels – you can solder four of these arcs together to make a complete circle. Instead of using it for its intended purpose, we're using the quarter-circle as the focal point of a bib necklace. We started with the shape of the NeoPixel arc, and widened it to create the fan shape. Then we topped it off with a piece of blue acrylic for diffusion, and added delicate, swirling line art so that the necklace will look just as great when the NeoPixels are powered off.

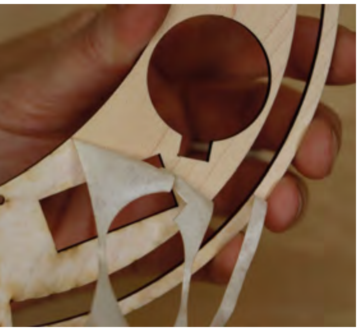
The battery in this project is a tiny lithium-ion one. These are great little powerhouses that are perfect for driving NeoPixels while keeping the project small. However, these batteries can be dangerous if not handled properly and must be treated with care. Be careful not to puncture, compress, or otherwise abuse this type of battery. Ensure that it is held securely but not squished inside your project. The chemistry in these batteries can vary among brands, so it's important to only charge them with the chargers recommended by the battery's manufacturer. Do not cut the wires of the battery, and do not pull on the wires to unplug the JST from the Gemma or charger. Inspect the battery often for damage or bulging. If any change is observed in the battery's shape, dispose of it properly according to your local hazardous waste regulations. Because these batteries require such careful handling, this project is for intermediate to advanced makers who are comfortable working with power supplies.

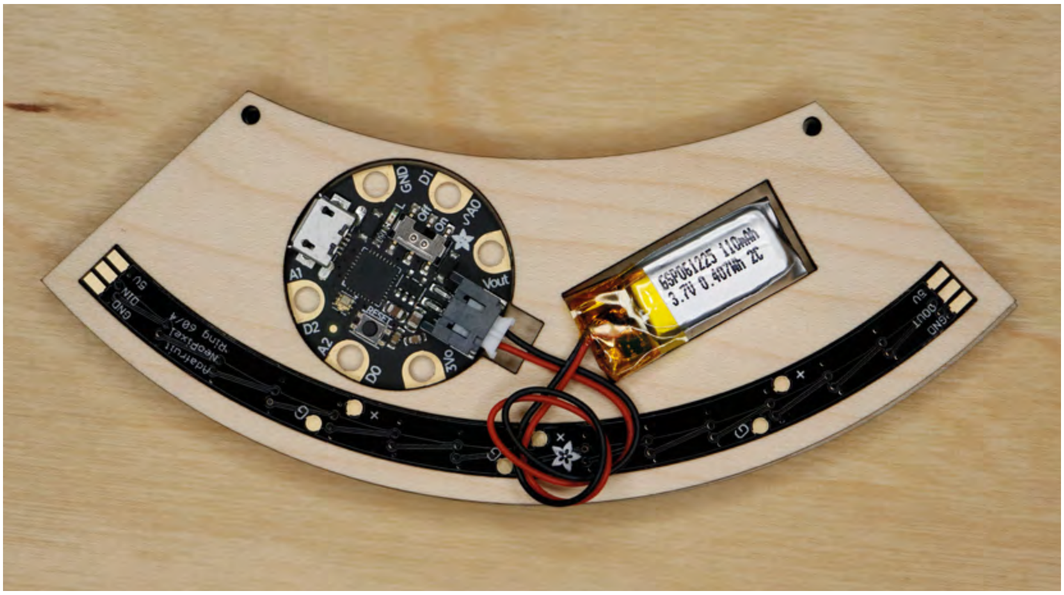
CUT THE PIECES

Use the provided files (**hsmag.cc/book2-assets**) to cut your pieces out on the laser cutter. For our design, the two back pieces are cut out of wood, and the front piece is cut out of acrylic as a diffusion layer. The front piece has two files: one for cutting the shape, and one for scoring line art on the surface. Be careful not to accidentally cut the scoring layer. You may need to experiment with the settings on your laser cutter to get the depth you prefer. We highly recommend doing some test cuts before cutting the final piece. →

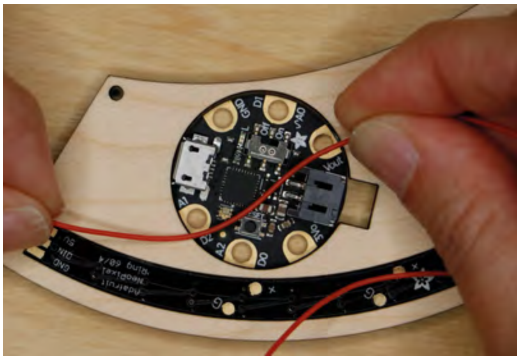
YOU'LL NEED

- **Gemma M0**
- **100 mAh lithium-ion battery (don't forget the charger!)**
- **NeoPixel 1/4 60 Ring – 5050 RGB LED** (product #1768)
- **Thin wood for laser cutting (we used Glowforge medium maple hardwood)**
- **Acrylic for laser cutting (we used translucent light blue acrylic)**
- **Double-sided tape**
- **Jewellery cord**
- **Clasp, cord ends, split jump rings**
- **Jewellery crimping pliers**
- **30 AWG silicone-coated stranded wire**
- **Soldering tools and supplies**





Above & Right
All the laser-cut pieces should fit snugly together



Materials that are sold for use in laser cutters come with protective film on both sides to prevent scorching from the laser. Leave this film on the material while cutting. When your pieces are cut, you can remove the protective film on both sides. This can be a bit tedious when dealing with the tiny scored details, but just take your time and enjoy the reveal!

> **With the NeoPixel arc face down,** gently press it into place at the bottom of the piece. It will be a snug fit and should stay in place

ASSEMBLE THE NECKLACE

Start with the back layer that will hold the electronic components. Lay the piece down on your surface, with the hole for the Gemma M0 on the left, and the battery hole on the right. With the NeoPixel arc face down, gently press it into place at the bottom of the piece. It will be a snug fit and should stay in place without additional support. If your NeoPixel arc doesn't fit, use a small file to widen the hole in the wooden piece where you need to. If the arc is loose, add a few small dabs of hot glue to keep it in place.

Lay the component piece on top of the other wooden piece, and check that each NeoPixel lines up with its hole in the front piece. With these two layers stacked together, put the Gemma M0 into its hole. If the Gemma M0 has any burrs around its edge, you may need to carefully file them down with a nail file first. The JST battery port on the Gemma M0 should align with the square notch pointing toward the battery hole. Plug your battery into the Gemma M0 and check that it fits into its hole as well. Make sure the battery fits comfortably and is not compressed. Use a file to enlarge any holes that don't fit their components. When you're satisfied that everything fits, remove the battery and set it aside before proceeding.

BUILD THE CIRCUIT

It's time to wire everything together! We'll be soldering the connections as follows: GND on the NeoPixels to GND on the Gemma, data-in on the NeoPixels to D1 on the Gemma, and 5 V on the NeoPixels to Vout on the Gemma.

We want to reduce bulk on the back of the necklace as much as possible, so we'll keep our wires short and neat. Cut three pieces of wire that reach from the pads on the data-in side of the NeoPixel arc to their corresponding pads on the Gemma M0. Make sure the wires are a little longer than you need them for now; we will cut them down exactly later on.

Strip and tin one end of each wire, and tin the pads on the data-in side of the NeoPixel arc. Carefully solder one wire to each pad, being sure not to bridge the contacts on the NeoPixels – they're very close together, and it's easy to touch two of them by mistake. Solder the wires so that they point back towards the Gemma.

Take the end of each wire and hold it over its pin on the Gemma M0. Cut each wire exactly to length, leaving about 1 mm for soldering. Strip and tin about 1 mm of each wire, then tin GND, D1, and Vout on the Gemma M0. Solder each wire to its pad. Tweezers can be helpful here for precision. Don't worry too much if you burn the wood a little with your soldering iron – it won't show on the front.

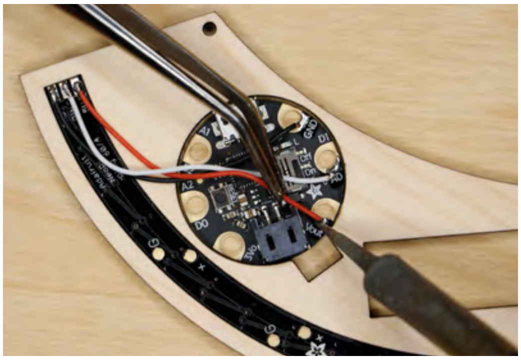
PROGRAM THE GEMMA M0

With the circuit assembled, let's program the Gemma M0 with CircuitPython and make our LEDs light up! Since we're using a tiny battery, we'll create an animation that only illuminates the NeoPixels for a short amount of time. This will increase the battery life of your necklace greatly while still allowing for a high impact look. Our code will intermittently send a colour, chosen at random, across the NeoPixel strip from left to right. Then the NeoPixels will darken from left to right and pause before sending the next colour.

Plug a micro USB cable into the Gemma M0 and connect it to your computer. The Gemma will show up as a drive called CIRCUITPY.

As always, make sure your Gemma M0 has the latest version of CircuitPython and updated libraries. Follow the guide at **hsmag.cc/ZKvDqa** to make sure your Gemma M0 and its libraries are up to date before starting to code. The code below uses the **neopixel.mpy** library, so you'll need to copy the latest version of the library into the **lib** folder on the CIRCUITPY drive.

You can use any text editor of your choice to program the Gemma M0, but we like to use a Python code editor called Mu. To learn more about Mu and how to use it, visit Adafruit's guide on Mu here: **hsmag.cc/NtvMQr**.



Above Cut the wires to the correct length to keep it neat

In Mu, or your text editor, open the **main.py** file on the Gemma M0. The Gemma comes preloaded with example code, but we'll write our own colourful animation here instead. Select all the text in the file and delete it to start from scratch.

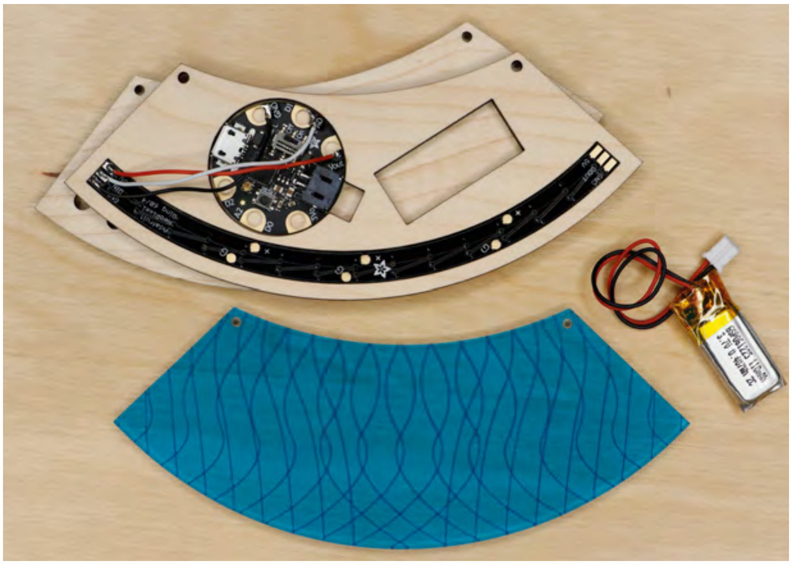
The first step is to import the libraries we'll be using and set up the NeoPixel arc as an object. We'll also create some variables that are common to NeoPixel sketches: `pixel_pin` and `num_pixels`. Adding these two variables at the top will make it easy to modify this code for other projects with different numbers of NeoPixels.

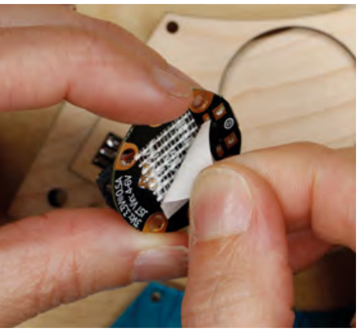
```
import time
import board
import neopixel
import random

pixel_pin = board.D1
num_pixels = 15

pixels = neopixel.NeoPixel(pixel_pin, num_pixels,
brightness=0.1, auto_write=False)
```

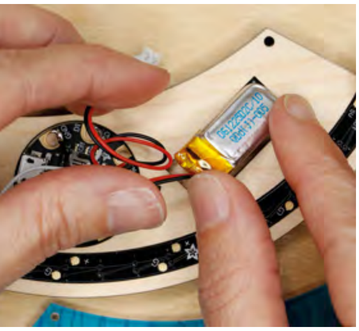
Next, we'll use a helper function called `wheel`, which will let us use a single number to assign a colour to the NeoPixels, instead of having to enter separate RGB values as (r, g, b). When we give it a number between 0 and 255, the `wheel` function will return the corresponding colour in (r, g, b) format. Giving the `wheel` function 256 will return (0, 0, 0), which is black. →

Below If you use a different board, you'll need to adjust the design

Bottom Everything works! Time to put it together







Above
Sticky tape, a maker's best friend

```
def wheel(pos):
    # Input a value 0 to 255 to get a color value.
    # The colours are a transition r - g - b -
back to r.
    if pos < 0 or pos > 255:
        return (0, 0, 0)
    if pos < 85:
        return (255 - pos * 3, pos * 3, 0)
    if pos < 170:
        pos -= 85
        return (0, 255 - pos * 3, pos * 3)
    pos -= 170

    return (pos * 3, 0, 255 - pos * 3)
```

Now we're ready for the main loop. First, we need to choose a colour for the NeoPixels. To do this, we create a variable called **`color`** and assign it a random number value between 1 and 256 using **`random.randint`**. If the number generated is between 1 and 255, the NeoPixels will show the corresponding colour from the colour wheel. If 256 is generated, the pixel will turn off (black). If you're using a code editor like Mu, printing generated values like this in the serial monitor can help when debugging.

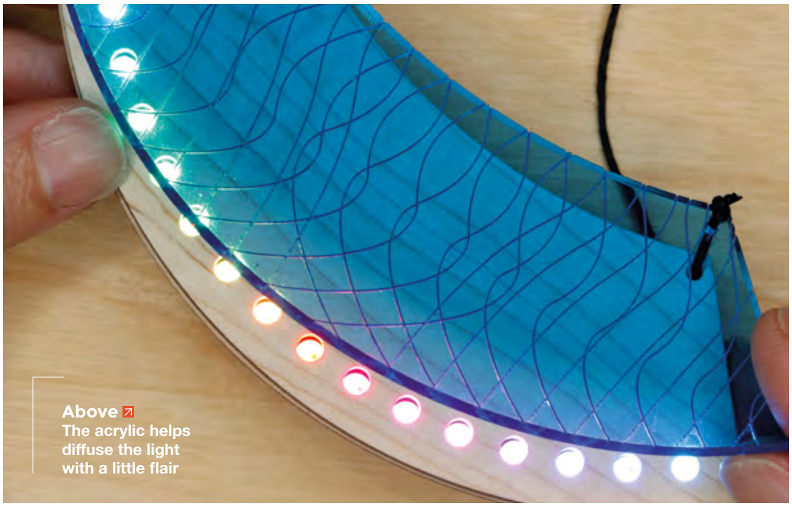
```
while True:
    color = random.randint(1, 256) #choose a
random color, includes black

    print("color", color)
```

Next, we have two **`for`** loops. The first **`for`** loop uses the **`color`** variable to fill the strip of pixels one at a time, starting at the first pixel (0), until the whole strip is filled. When that task is complete, the second **`for`** loop will take over, and start filling the strip with black, turning off the pixels. This will create a sort of chase effect from left to right on the strip (or clockwise in a circle if using a NeoPixel ring), with a new, randomly chosen colour each time the strip fills. Changing the **`time.sleep`** values will change the speed of the animation, so play with this until you like the effect.

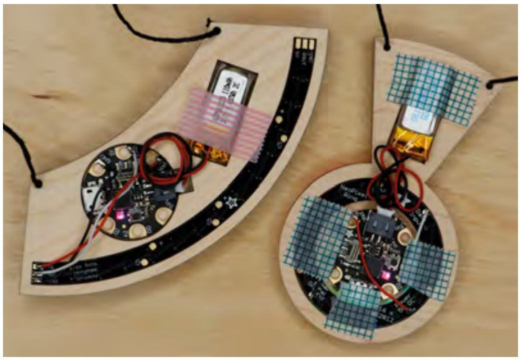
Above
A final knot to bring everything together

```
    for j in range(num_pixels):
        pixels[j] = wheel(color) #set (the random
pixel + j) to the random color
        pixels.show()
        time.sleep(.01)
        j = j+1 #add 1 to j every time

    for i in range(num_pixels):
        pixels[i] = wheel(-1) #same as above but
set pixels to black
        pixels.show()
        time.sleep(.01)
        i = i+1

    time.sleep(2)
```

Saving the edited **main.py** file on the CIRCUITPY drive will automatically load the code onto the Gemma and you should see the colour chase animation appear on your necklace. If your NeoPixels are lighting up as expected, hooray! You can unplug the micro USB cable and move on to installing the Gemma M0. If your NeoPixels don't light up, double-check your connections and make sure you haven't bridged the tiny pads on the NeoPixel arc. For help with troubleshooting and more information about programming the Gemma M0 with CircuitPython, visit **hsmag.cc/ZKvDqa**.



Above
The acrylic helps diffuse the light with a little flair



Above
If in doubt, tape!

INSTALL THE ELECTRONICS

When your NeoPixels are lighting up as expected, you can tack the Gemma M0 into place. You'll want to make sure that you can access the micro USB port in the future, in case you want to update your necklace with new animations. You'll also need to be able to remove the battery's JST plug from the connector on the Gemma M0 for charging. For this reason, we recommend taping the components in place instead of permanently gluing them in. Double-sided carpet tape is both thin and strong, and works well for this type of application.

With both wood layers stacked together, use double-sided tape to affix the Gemma M0 in place in its hole. It will be stuck to the back of the front wooden piece. Again, make sure the JST port is facing the square notch. Use another piece of double-sided tape to stick the battery in place as well.

The battery may have long wires, but instead of shortening these, just wind them gently into a low profile coil on the back of your necklace. You can use a piece of single-sided tape to hold the coil in place. Do not put any stress on the point where the wires connect to the battery. It's a good idea to stabilise this point with some gaffer tape or a dab of E-6000 glue.

> "With both wood layers stacked together, use double-sided tape to affix the Gemma M0 in place in its hole"

FINISH ASSEMBLY

Place the acrylic layer in the front of the stack, on top of the first wood piece. With regular translucent acrylic, it's nice to see the scored lines on the front of the necklace, but if you are using fluorescent acrylic you may want the score lines to be next to the LEDs for a cool line-art effect. The choice is yours!

Cut two equal length pieces of jewellery cord. Each piece should be half of the final length of the necklace you want to make. For the necklaces shown here, the cords are about 10 cm long. Thread each piece through one of the holes at the top corners of the stack, from back to front. On the front of the necklace, tie a tight knot at the end of the cord.

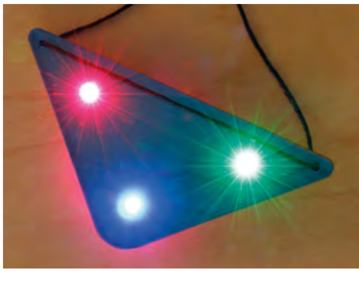
At the other end of each cord, use crimping pliers to add cord ends for jewellery findings. Then add split jump rings and a clasp. Split jump rings are coiled so that they are less likely to open during wear. Pull the coil open slightly and wind the ring onto your cord end. Use the same process to add a clasp to one side, and you're done! The Gemma M0 has an on-board on/off switch, so flip that switch on and watch your necklace light up.

GOING FURTHER

Once you've made one necklace, it's easy to make more in different shapes. Use our designs (from **hsmag.cc/book2-assets**) or make your own shapes. Instead of diffusing the NeoPixels through a layer of acrylic, try scoring the top wood layer with line art. You can use the same code for a few sewable NeoPixels, or even a NeoPixel ring. Just change the `num_pixels` variable to the number of NeoPixel LEDs in your new design.

Whether you make this project, or something completely different, we want to see it! Show us your builds at **hackspace@raspberrypi.org**! □

Below
Unleash your creative juices and make your own light-up jewellery



Adding WiFi with an ESP32

Add networking to your favourite microcontroller

Ben Everard

@ben_everard

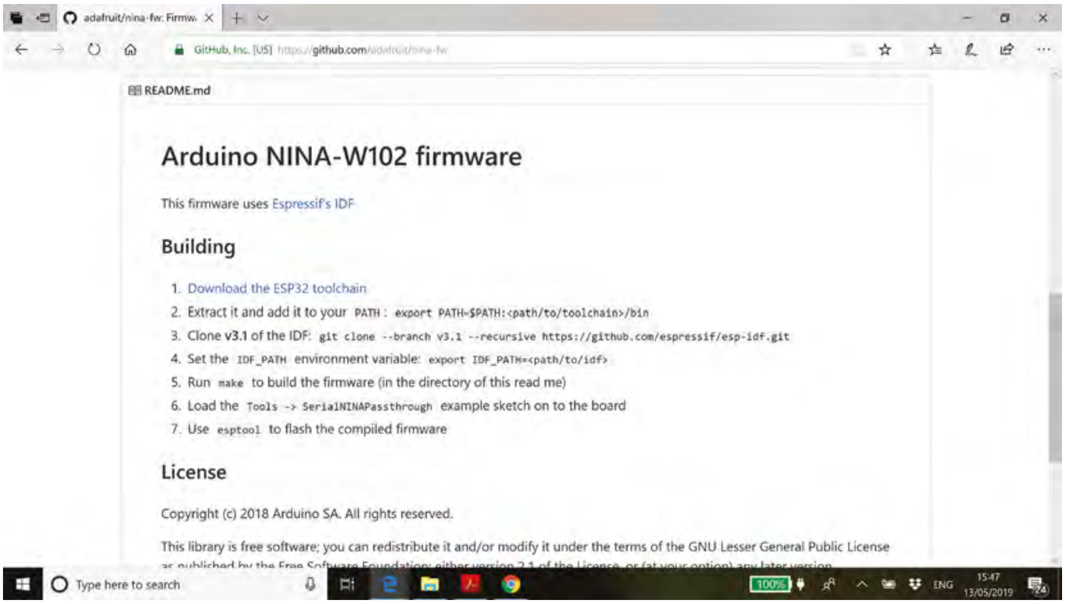
Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

There's a new breed of microcontroller board around that combines a main processor with a WiFi coprocessor. Boards like the Arduino Uno WiFi Rev2 and the PyPortal let you dedicate the main processor to controlling the hardware, while letting you offload the sending and receiving of data to a second processor – usually the ESP32. Let's take a look at how we can do this to add WiFi support to a microcontroller. This isn't for the faint-hearted. Although the hardware is fairly straightforward, setting up the tools to compile the code is a bit of a mammoth task. Don't worry, though – we're here to guide you.

We'll need an ESP32 module. Usually these come with all the features we need, but make sure you've got the ability to upload code (usually through on-board USB, though you can do it via UART and the main microcontroller if you wish), and the required pins exposed – these are:

- IO 5 (CS)
- IO 14 (MOSI)
- IO 18 (SCK)
- IO 23 (MISO)
- IO 33 (Busy)
- Reset
- 5 V
- GND

For a 3.3 V microcontroller, we won't need any more hardware, but if you want to connect this to a 5 V board (such as an Arduino Uno), you'll need level shifting to convert between the voltages. Note that even though the logic is 3 V, we need to power the ESP with 5 V, as this is needed for the WiFi.



Arduino NINA-W102 firmware

This firmware uses Espressif's IDF

Building

1. Download the ESP32 toolchain
2. Extract it and add it to your `PATH` : `export PATH=$PATH:<path/to/toolchain>/bin`
3. Clone **v3.1** of the IDF: `git clone --branch v3.1 --recursive https://github.com/espressif/esp-idf.git`
4. Set the `IDF_PATH` environment variable: `export IDF_PATH=<path/to/idf>`
5. Run `make` to build the firmware (in the directory of this read me)
6. Load the `Tools -> SerialNINAPassthrough` example sketch on to the board
7. Use `esptool` to flash the compiled firmware

License

Copyright (c) 2018 Arduino SA. All rights reserved.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

Right As the WiFiNINA firmware is open source, you can peek inside as well as upload it to your projects

AIRLIFT

If you'd rather someone else do the hard work for you, you can get an ESP board preprogrammed with this firmware from Adafruit. Its AirLift board comes all set up, ready to be used in a project. As it doesn't have any extraneous hardware, it's smaller than most ESP32 modules as well. Take a look at **hsmag.cc/pCQjGW** for details.

As always, the hardware needs to be paired with software. We want the microcontroller to be able to send commands to the ESP to control what network it connects to and what data to send and fetch. There are a few ways of shuffling data between two devices, but the fastest available to us is Serial Peripheral Interface (SPI). This actually needs a couple more connections than regular 3-wire SPI – we'll look at those a bit later.

Thanks to the great work of open-source-led organisations like Arduino and Adafruit, we don't need to write our own firmware for this. Arduino created the WiFiNINA firmware for its boards, and Adafruit adapted it for its own boards. It's the Adafruit branch of this that we'll use to power our hardware.

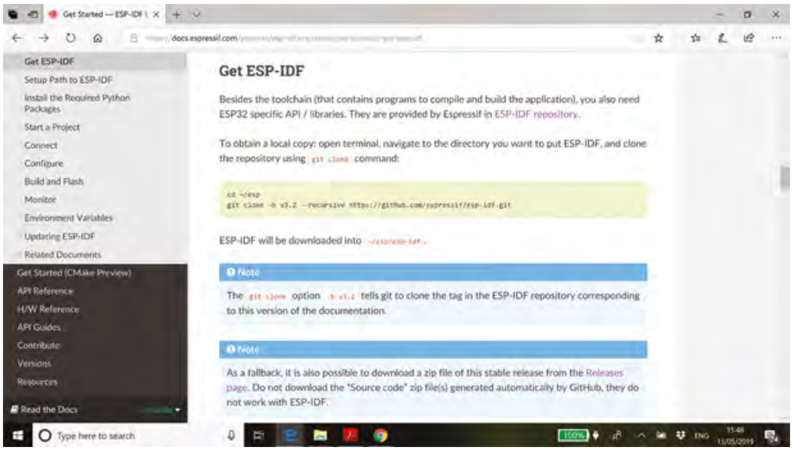
The WiFiNINA firmware is created using the ESP toolchain and the ESP Integrated Development Framework (IDF).

> " Arduino created the WiFiNINA firmware for its boards, **and Adafruit adapted it for its own boards** "

The toolchain is available for Windows, macOS, and Linux, and you can download the latest version from here: **hsmag.cc/fNiliC**. It's not the most straightforward build process on Windows, as it involves switching some versions of a terminal emulator; however, it does work.

If you look at the GitHub page for the Adafruit WiFiNINA firmware (**hsmag.cc/iuFyQy**), the README will tell you to use a particular version of the IDF, but this is out-of-date. You need the latest version, and you can get this by running the following (in the terminal provided by the toolchain):

```
git clone --recursive https://github.com/
espressif/esp-idf.git
```



Get ESP-IDF

Besides the toolchain (that contains programs to compile and build the application), you also need ESP32 specific API / libraries. They are provided by Espressif in ESP-IDF repository.

To obtain a local copy: open terminal, navigate to the directory you want to put ESP-IDF, and clone the repository using `git clone` command:

```
cd ~/esp
git clone -b v3.2 --recursive https://github.com/espressif/esp-idf.git
```

ESP-IDF will be downloaded into `~/esp/esp-idf`.

Note

The `git clone` option `-b v3.2` tells git to clone the tag in the ESP-IDF repository corresponding to this version of the documentation.

Note

As a fallback, it is also possible to download a zip file of this stable release from the Releases page. Do not download the "Source code" zip file(s) generated automatically by GitHub, they do not work with ESP-IDF.

This will download the IDF to the current directory. You'll need to add the newly created directory to your path and your IDF_PATH. In the examples below, we cloned the GitHub repo to **/c/msys32/esp-idf**.

```
export PATH=$PATH:/c/msys32/esp-idf
export IDF_PATH=/c/msys32/esp-idf
```

You'll also need to download the code for the device firmware. You can get this with:

```
git clone https://github.com/adafruit/nina-fw.git
```

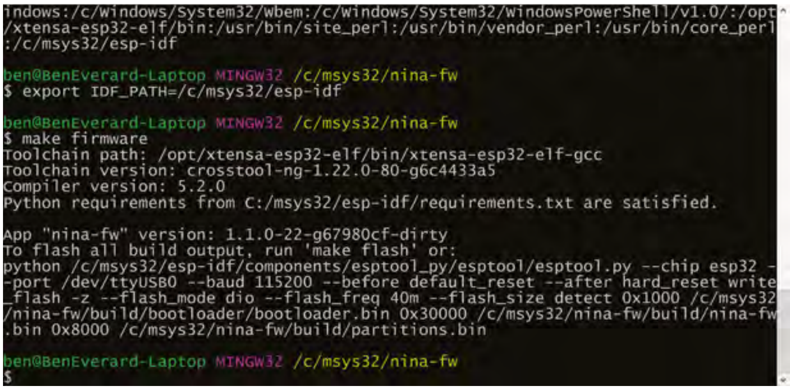
Again, this will download the code to the current directory. If the IDF and toolchain is set up correctly, you can cd into the new directory and build the firmware. This is done with:

```
cd nina-fw
make firmware
```

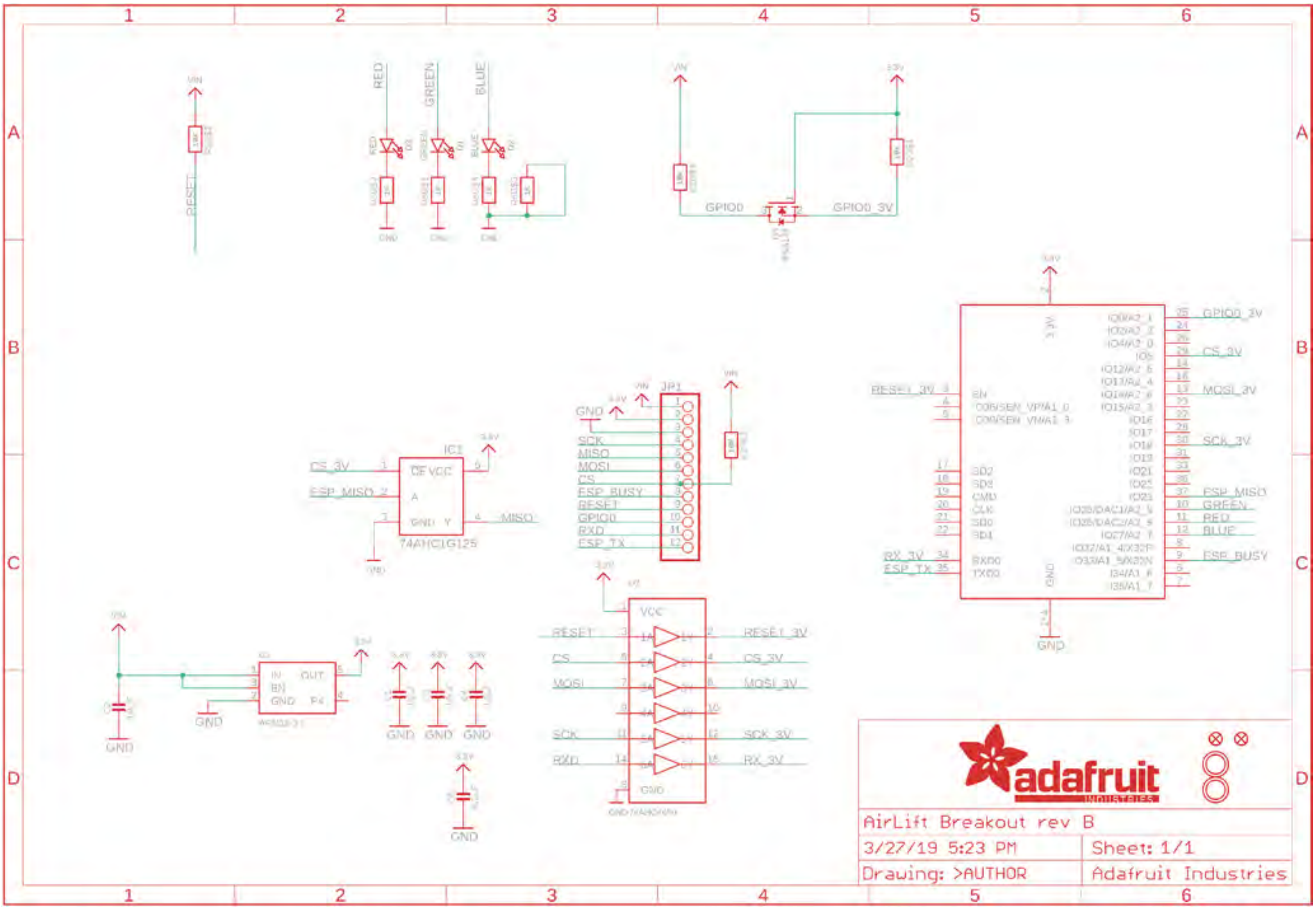
You'll now need to flash the firmware onto the board. The final stage of the compilation should have given you a rough command to run in order to flash the firmware. However, if you're running this on Windows, remember that this is based on the MSYS →

Above We're only scratching the surface of what's possible with the ESP development toolchain. Take a look in the ESP example folders for more ideas on what you can do

Below The compiling should output a template for uploading the code to your ESP32



```
indows:/c/Windows/System32/Wbem:/c/Windows/System32/WindowsPowerShell/v1.0/:/opt
/xtensa-esp32-elf/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/core_perl
:/c/msys32/esp-idf

ben@BenEverard-Laptop MINGW32 /c/msys32/nina-fw
$ export IDF_PATH=/c/msys32/esp-idf

ben@BenEverard-Laptop MINGW32 /c/msys32/nina-fw
$ make firmware
Toolchain path: /opt/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a5
Compiler version: 5.2.0
Python requirements from C:/msys32/esp-idf/requirements.txt are satisfied.

App "nina-fw" version: 1.1.0-22-g67980cf-dirty
To flash all build output, run 'make flash' or:
python /c/msys32/esp-idf/components/esptool_py/esptool/esptool.py --chip esp32 -
-port /dev/ttyUSB0 --baud 115200 --before default_reset --after hard_reset write
_flash -z --flash_mode dio --flash_freq 40m --flash_size detect 0x1000 /c/msys32
/nina-fw/build/bootloader/bootloader.bin 0x30000 /c/msys32/nina-fw/build/nina-fw
.bin 0x8000 /c/msys32/nina-fw/build/partitions.bin

ben@BenEverard-Laptop MINGW32 /c/msys32/nina-fw
$
```



Above Good hackers borrow, great hackers steal

environment, which doesn't have access to the USB connection. So we'll need to set up Windows so it can do the flashing.

You'll need Python installed. If you don't have this already, the easiest way to get this is via Anaconda. You can grab Python3 from **anaconda.com.**

Now, open an Anaconda shell, and install the ESP flashing tool with:

```
pip install esptool
```

It's now time to flash the firmware. You can use the command from the compilation, but you'll need to change the paths to Windows-style (usually, this means replacing /c/ with c:/ in the file paths) and changing the port from /dev/ttyUSB0 to the actual port. Typically, this is COMXX, where XX is a number. You can check which COM port your board is on by plugging it in and opening the Arduino IDE. It will show the available ports in Tools > Ports.

That's it. You now have your coprocessor programmed. It's time to link it up to your main processor. We'll use the Adafruit Grand Central as an example, as this bucket-load of GPIO is an obvious candidate for adding WiFi to.

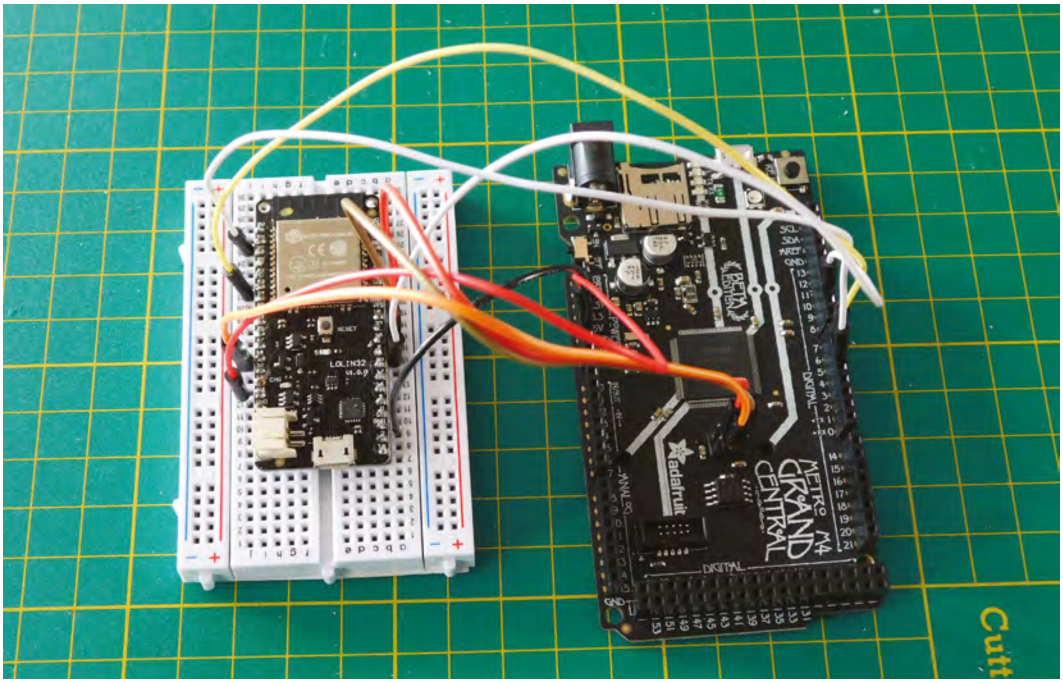
The ESP32 will need quite a bit of power – possibly as much as 350mA when transmitting – so you'll need a power supply that can deliver this much at 5V. The internal regulator on the Grand Central can do this, but we found that we had to use an external power supply – the USB on our laptop couldn't send sufficient current. So, with both our boards unplugged and powered off, let's connect them. We don't need any components, just wires.

OPEN HARDWARE AND SOFTWARE

We were only able to write this tutorial because both the hardware and software that it's based on are open source. We obviously needed the software to be open so we could compile it for our boards, but the hardware being open made it much easier to make sure we had the right connections. We were able to find out which pins connect to which using the schematic of the Adafruit AirLift.

As you can see, the pins are labelled on the ESP (the IOXX number refers to the GPIO number), and should match with the numbers on your ESP module.

We owe a huge thanks to both organisations – and all others that open source their work.



Left The coding might be a bit complex, but the wiring is fairly straightforward

- 5 V on your ESP needs to go to 5 V on your microcontroller (having checked the data sheet to make sure this pin can supply enough current. If not, you'll need a separate 5 V supply.)
- GND on your ESP to GND on your microcontroller. If you've used a separate supply, you'll need to connect it to both GND on the microcontroller and GND on the supply.
- IO 5 (CS) to digital 10
- IO 33 (Busy) to digital 7
- Reset to digital 5

You will also need to connect the SPI pins to the relevant pins on the microcontroller. On the Adafruit Grand Central board, this is on a six-pin SPI header – that's the six-pin male header, located just below the main CPU. When held so the text on the board is the correct way up, ESP IO 14 (MISO) connects to the top-left pin, ESP IO 18 (SCK) to the pin below this, and ESP IO 23 (MOSI) to the middle-right pin.

With all that connected, we just need to program the board. There's CircuitPython code that can communicate with WiFiNINA, but the Arduino code is a little more verbose for checking errors. You'll need the Adafruit version of the WiFiNINA library as we have used that version of WiFiNINA. You can download this from **hsmag.cc/0SNRpz,** and unzip it to your Arduino Libraries folder.

In the Arduino IDE, in Files > Examples, you should now have some sketches for testing out the firmware.

The first one to check everything's working is WiFiNINA > ScanNetworks.

Flash this to your main board and open a serial monitor. If everything's gone well, after a few seconds, you should see a list of local access points that you could connect to. If something goes wrong, there are two likely points:

- If you don't get a MAC address, then there's been a problem communicating with the ESP. The most likely problem is the wiring, but if that looks good, make sure that the firmware uploaded properly.
- If you don't get a list of networks, then something went wrong with the scan. This is most likely caused by insufficient power during transmission, so try again with a more powerful power supply.

If that's gone well, you can connect to a network using the WiFiWebClient example sketch (you'll have to set the WiFi network details in the Secrets tab). This sketch should give you the details you need to connect to other websites and pull or push the data you need. As all the WiFi work is done on a separate processor, you've still got the full power of your main processor available to control hardware, or do whatever you want your project to do. □

Working with polystyrene plastic

Prototype, mock-up, and model with surprising speed using this common plastic

Gareth Branwyn

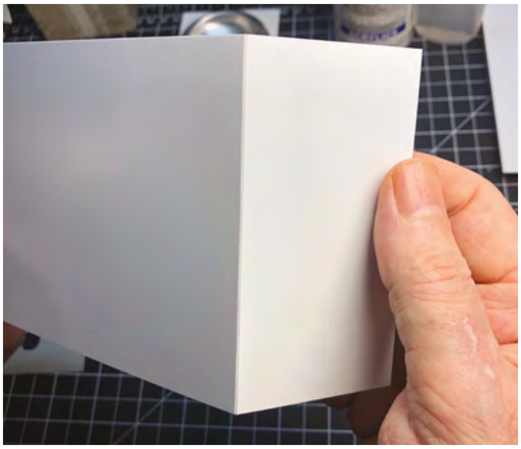
@garethb2

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

Anyone who has built a scale model kit has worked with polystyrene plastic (often shortened to 'styrene' and abbreviated as 'PS'). This ubiquitous thermoplastic is moulded into everything from disposable product packaging through CD/DVD 'jewel boxes' to model kits and toys. But what many makers may not be aware of is the speed and versatility of this material for mocking up three-dimensional designs – for prototyping, creating custom project boxes, and for scratch-building models. It can also be used in vacuum forming. With a few simple tools, a pot of cement, and a bit of practice, you can render a design in 3D with impressive speed. If properly engineered, the resulting object can be surprisingly durable and strong.

ACQUIRING STYRENE

Polystyrene stock can be purchased in a dizzying array of sizes, thicknesses, shapes, and textures. You can buy sheet, tube, rod, strip, and architectural detail stock directly from well-known manufacturers like Plastruct (**plastruct.com**) and Evergreen Scale Models (**evergreenscalemodels.com**), at online retailers, or from your local hobby or game store. If you're new to buying styrene stock, you might want to see what you are getting physically, to choose the types of thicknesses and shapes you are interested in. To build up a supply, Evergreen sells assortment bundles of sheet stock, tubes, and rods. Common sheet sizes are 6″×12″ (15 × 30 cm) and thicknesses of .010″ (.25 mm), .020″ (.50 mm), and .040″ (1.0 mm). Rods, tubes, I-beam, T-beam, L-beam, and other shapes come in various sizes at 14″ (35 cm) length. If you start using styrene a lot as a prototyping/building material, besides acquiring a supply of new stock, you can also start collecting polystyrene-based packaging, sprue material from model kits, and model kit parts to incorporate into your creations.



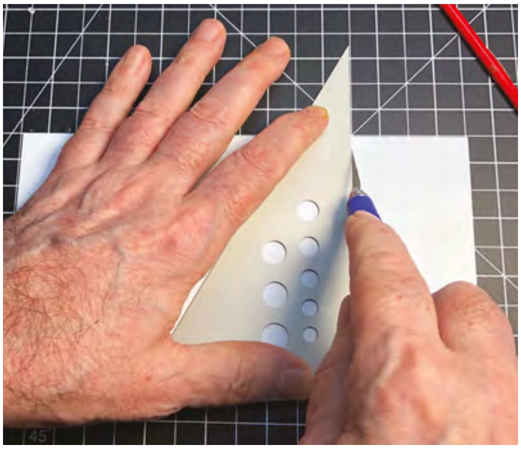
Right You may need to 'deburr' breaks with a knife or sandpaper

SCORING AND BREAKING

One of the great benefits of styrene is that it is a score and break plastic. You don't have to cut through it all the way with a razor knife – you simply have to score it a few times, and then you can gently bend and break it cleanly along the score line. For thicker stock, it is recommended that you use a plastic cutter (or linoleum knife) for deeper scoring. Square nose pliers are also helpful for creating a clean break along the score line of thicker stock. Once you have

BEND ME, **SHAPE ME**

Because polystyrene is a thermoplastic, you can easily heat-bend it to achieve many desired shapes. In hobby modelling, a common technique is called 'sprue-bending'. This is where you take a piece of leftover model sprue (the plastic frame that holds small extrusion moulded parts), hold it over a candle flame, and bend it to get the desired shape you want. With a little practice, you'll be amazed at what sorts of parts you can fabricate with scrap or stock styrene rod.



Above Styrene makes quick work of generating parts with the ‘score and break’ method. Simply score a few times with a sharp hobby knife, and then gently bend for a clean break

a design plan for what you are constructing, and have precisely marked out your pieces, you can quickly and accurately score and break all of the plastic components you need.

DEBURRING

After scoring and breaking your pieces, you’ll want to check to make sure that you have clean cuts, without burrs. If you do find excess plastic along your breaks, you can use perpendicular strokes of a hobby knife blade (or a deburring tool) to clean everything up before gluing. A light sanding of the edges can also clean up any excess.

> What many makers may not be aware of is the speed and **versatility of this material for mocking up 3D designs**

BASIC CONSTRUCTION

Polystyrene is an impressively strong material for its weight, and it can be very flexible in smaller thicknesses. To maximise its structural integrity, you want to think about reinforcing your design. Styrene is very forgiving to work with. As professional model-maker (and MythBuster) Adam Savage likes to say, “styrene hides a ton of crimes.” It is a material that’s perfect for precision construction, but it is also great for improvisational designing and scratch-build model-making. For a perfect example of this, check out Savage’s YouTube build (**youtu.be/ZfvtGrhYk0I**) where he scratch-builds an interpretation of a Mobius sci-fi illustration in a day. This video is a styrene building masterclass, with tons of great ideas.

GLUING

There are a number of glues and solvents you can use with polystyrene. The most common is plastic model cement. The drawback here is a longer set time. Using cyanoacrylate (CA) glue can greatly shorten the wait and allow you to build/prototype much faster. Professional model-makers use a solvent cement, such as Weld-On. This can quickly be brushed on or applied with a needle applicator. Professional solvent cement is rated by drying times. Weld-On #3 has a one-minute working time and a two-minute fixture time – a common choice for quick constructions. The drawback to solvent cement is that it is extremely flammable and requires good ventilation. For structural integrity, it beats all glues because it literally fuses the plastic pieces together. □

FINISHING

One of the other great benefits to working with polystyrene is that it is easily sanded. If you ‘work proud and sand down’, you can quickly create smooth and perfect edges, hide seams, etc. A good complement of sandpaper, sanding block, and sanding sticks, at varying grits, will allow you to quickly remove material and work your way up to a desired smooth finish.

To get the best paint results, it is a good idea to clean your polystyrene constructions with alcohol (or just a light bath in mild detergent water) before painting. Polystyrene can be primed with spray (or brush-on) acrylic, enamel, or lacquer primers, and finishes are beautiful with brush-on or air-brushed acrylics. The composition and finish of styrene is such that you can achieve a wide range of faux-finish and weathering effects with this material.

Left Structures can be assembled quickly using brush- or needle-applied solvent cement. A simple solvent dispenser can be made from a soda can bottom and styrene card

YOU’LL NEED

BASIC TOOLS

- **Cutting mat** (or similar sacrificial surface)
- **Hobby knife** (with new, sharp blades)
- **Metal ruler and triangle**
- **Glue** (Cyanoacrylate (CA) glue or plastic cement)
- **Sanding paper** (or stick or block)

ADDITIONAL TOOLS

- **Plastic glue or solvent cement**
- **Plastic/linoleum cutter**
- **Mitre shears** (for cutting precise angles)
- **Styrene chopper** (for making a lot of repetitive cuts)
- **Square nose pliers**

WIN! 1 OF 10 AMAZING JAM HATS
The MagPi
The official Raspberry Pi magazine
Issue 82 | June 2019 | magpi.cc
BUILD A PI KEYRING
Keep your computer with you at all times
YURI 3 MARS ROVER
Recreating a space robot
LEARN TO CODE
SCRATCH & PYTHON
Issue 82 June 2019
Learn To Code wi

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

REVIEWS

DIRECT FROM
SHENZHEN

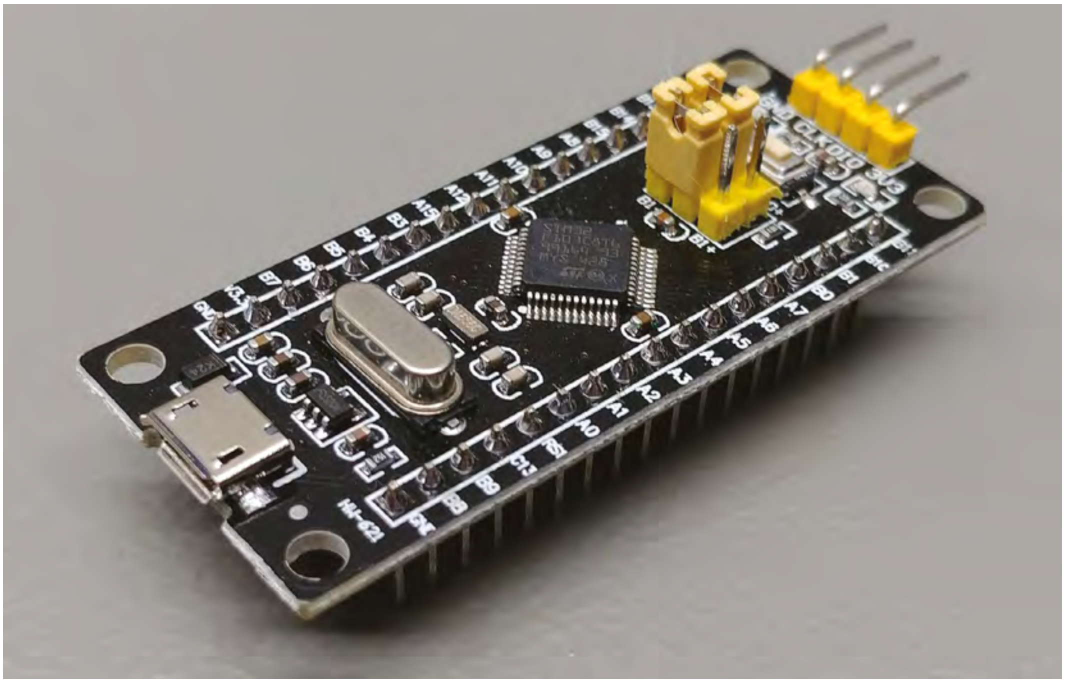
Black and Blue Pills

Two cheap boards built on the same microcontroller

By Ben Everard @ben_everard

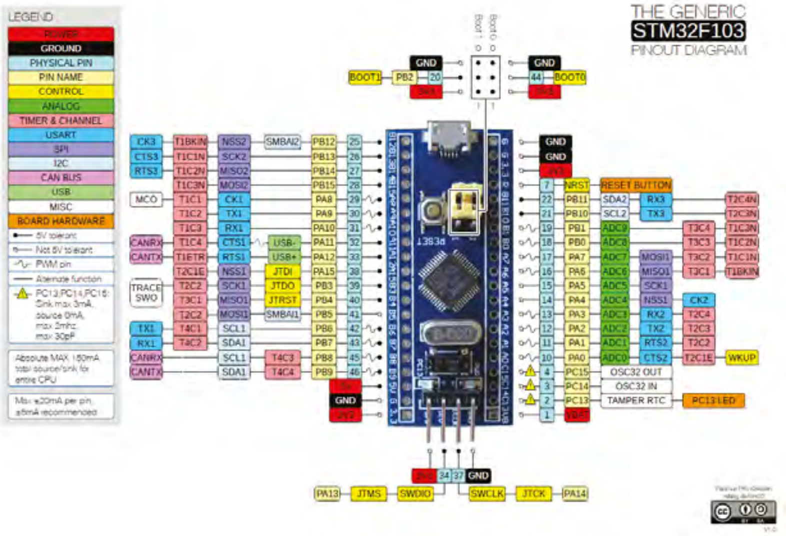
The 'Blue Pill' – a generic design of microcontroller, based on the STM32F103 – has been around for a while. The Black Pill is a newer, similar design based on the same MCU. These names are given to the boards by the community, so you won't find them for sale under these titles. Instead, they're usually called things like 'STM32F103C8T6 ARM STM32 Minimum System Development Board Module', and you'll have to pick them based on the image (as there are other board designs sold under similar names). There used to be a red version as well, but this doesn't seem to be available any more. The two we got were 'STM32F103C8T6 ARM STM32 Minimum System Development Board Module For Arduino Kj' for £1.79, including delivery, from GadgetsCloud on eBay for the Blue Pill, and 'STM32 Minimum System Development Board STM32F103C8T6 ARM Module for Arduino M' for £1.99, including delivery, from Ukings on eBay for the Black Pill. Similar boards are available for similar prices on most direct-from-China websites.

The CPU is based on an ARM Cortex-M3 running at 72MHz with 64kB of flash and 20kB of RAM. There



Right ⬗
The Black Pill is more robust, and more likely to have working USB

Credit
Thomas Gravekamp
GNU documentation licence



Right Both boards have an impressive selection of GPIOs and peripherals

are 37 IOs (35 on the Black Pill), including ten which can handle analogue input. All this comes for typically under £2.

While the MCU on both is the same, there are a few differences on the board. Most notably, the Blue Pill often has the wrong resistor on the USB port, which can cause problems for USB connections on some computers. It is possible to replace it, but it might be easier to get the Black Pill and avoid the problem. It's hard to know exactly what hardware you're getting as the board isn't version-controlled – there's not an official name, let alone official versions, so you just pay your money and see what turns up. Such is the nature of bargain-basement boards from no-name suppliers. For under £2, it can be worth the risk though.

> "These boards aren't as plug-and-play as boards made by hobbyist companies, and they don't have **WiFi like the similarly priced ESP8266 boards**"

DESIGN WOES

Another common problem is the soldering on the USB port. Although we didn't have any problems, some users have reported that it was weak and prone to coming off the board – it's usually fixable with a drop of solder. Again, this problem is solved on the Black Pill. For these two reasons alone, the Black Pill is worth the extra 20p, unless you specifically need the Blue Pill (such as if your other hardware is designed for it, or if you need the extra two GPIO pins on this board).

Although the board does have a USB port, it doesn't come with a USB bootloader by default, so you'll need to burn a bootloader to it. This can either be done using a JTAG adaptor or by using a USB to serial adapter. The USB to serial adaptor is cheap, and the setup is just a case of connecting the wires together, so this shouldn't be enough to put off people used to working with microcontrollers, but for beginners, it might be best to start with a more straightforward board.

Once the bootloader is burned, you can program as you would any other Arduino-compatible board. Install the correct board definition, then connect the USB port and upload the programs as needed. Not all libraries will work out-of-the-box, but many have been ported to the STM32 (and you can see the list of those here: **hsmag.cc/LvLKDu**).

As well as acting as a programming port, the USB port can be used to allow the board to act as a USB device. For example, the Venabili Keyboard (**venabili.sillybytes.net**) uses a Blue Pill to convert physical key presses into a USB communication your computer can understand (look out for more detailed coverage of this in a future issue). Similar processors were even used as the USB controller on the announced, but not yet released, Arduino Cinque which featured an open-source RISC-V microcontroller at its heart (**hsmag.cc/xhybzr**).

While there's a bit of tweaking necessary to get this board up and running, and no support from the manufacturers, there's a community of hobbyists who have got a lot running, and help each other out. You'll find lots of people working with these boards on the forums at **stm32duino.com**, and most of the info you need to get started at, **wiki.stm32duino.com**.

These boards aren't as plug-and-play as boards made by hobbyist companies, and they don't have WiFi like the similarly priced ESP8266 boards. However, they do have a fast processor and plenty of IOs.

There's something inherently nice about working with a board that doesn't quite work correctly alongside a group of other enthusiasts. You'll probably hit a few bumps along the way as you try to get a Blue or Black Pill to work, but those have probably been hit and documented by other users and, as you perform the workarounds, you'll find you learn a bit more about the workings of microcontroller boards. Of course, this is only interesting (rather than frustrating) if you've got the time and skill to go through the workarounds. For £2, we think they're well worth the money, purely for something to have a bit of a tinker with. You might find they fit your use-case perfectly, but bear in mind that there's a chance you might not be able to get them to work as expected. □

DIRECT FROM **SHENZHEN**

ONLY THE BEST

Where am I? A roundup of GPS modules

Geolocation made easy thanks to GPS breakout boards

By Marc de Vinck @devinck

Knowing where you are is fairly simple thanks to your smartphone, or in-car navigation system, based on the Global Positioning System (GPS) data. But there was a time when getting around wasn't such a trivial task. The US Department of Defense first started development of its GPS in the late 1970s, but it wasn't available for consumer use until the 1980s, and the entire system wasn't fully realised until the mid-1990s. There are also competing, or sometimes complementary, systems from China, Japan, Russia, India, and the EU.

So, finding out where you are has become fairly simple. And fortunately, because of all the amazing work from a series of makers and manufacturers, adding GPS to your electronics project has become very easy. In many cases, it's as simple as plug-and-play. In this Best of Breed, we'll be taking a look at several solutions for adding positional awareness to your Raspberry Pi or Arduino project.

Before making a decision on which board is the best for your project, make sure you research some of the important variables like time to determine your position, available antenna types, amount of satellite data used, and accuracy. And the most important thing to know is whether or not you can purchase the specific GPS in your country or for your specific use. Be sure to check out all the project pages for more detailed information and shipping information.

SparkFun GPS Logger Shield vs Adafruit Ultimate GPS Logger Shield

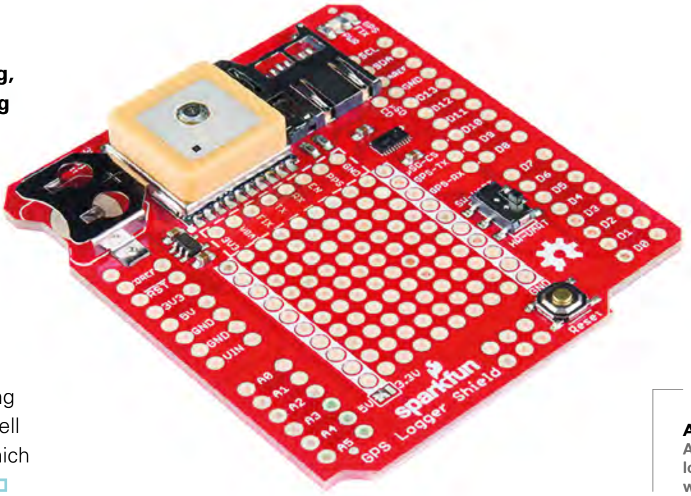
For when you need to remember where you've been

SPARKFUN GPS ◆ $49.95 | sparkfun.com

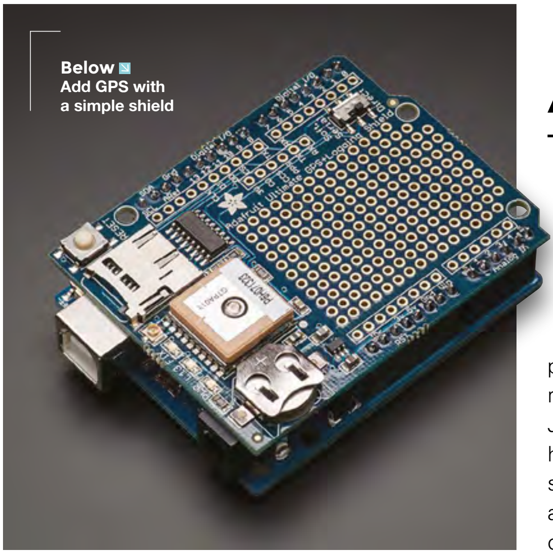
ADAFRUIT ULTIMATE GPS ◆ $44.95 | adafruit.com

The SparkFun GPS Logger Shield allows your Arduino to calculate position tracking, speed monitoring, altitude observing, all while logging the data to a microSD card. The switch on the shield allows you to select the GPS module's UART interface between either hardware or software ports, adding extra versatility to your project.

The shield also features on-board voltage regulation and distribution, making it really easy to use. The shield doesn't come with any headers, which are required to use it as a shield, so make sure to add that extra cost into your decision-making process. You'll also want to pick up a 12 mm coin cell battery for the GPS module and real-time clock, which allows for faster satellite acquisition on power-up. □



Above
Add GPS and data-logging capabilities with this shield



Below
Add GPS with a simple shield

Adafruit has been making GPS boards and shields for a long time. One of my first projects included an Adafruit GPS shield for Arduino that I soldered together. In the latest version of its GPS shield, Adafruit has made a major change. The new kit comes mostly assembled, including the GPS module, which in the past you had to add via a small connector cable. This makes the latest revision a lot easier to use, and much faster to implement into your project.

Each board is tested prior to shipping, and the price includes the male header pins which are required for using the shield with your Arduino. Just keep in mind that you will need to solder those header pins in yourself. Don't worry, it's about as simple as soldering gets. You'll also want to pick up a battery and microSD card to get all the functionality out of this handy little shield. □

VERDICT

SparkFun GPS

A good basic GPS board with data-logging capabilities.

8/10

Adafruit Ultimate GPS

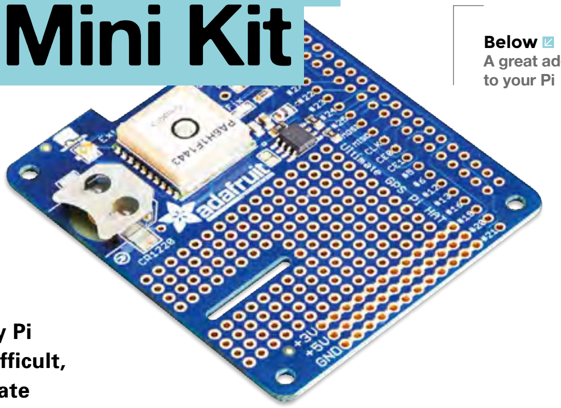
Well documented and easy to use.

9/10

Adafruit Ultimate GPS HAT for Raspberry Pi – Mini Kit

Finding your Pi from the sky

ADAFRUIT ◈ **$44.95** | adafruit.com



Below A great addition to your Pi

VERDICT

Adding to your Raspberry Pi is simple with this GPS HAT.

9/10

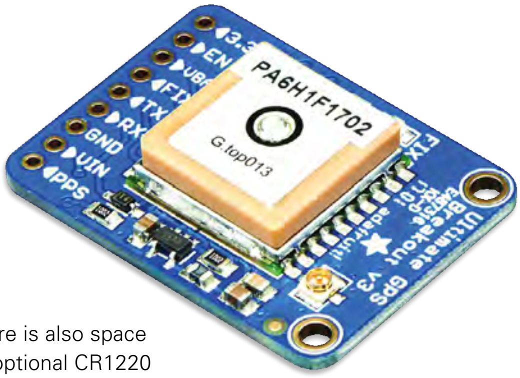
Adding GPS to your Raspberry Pi project doesn't have to be difficult, thanks to the Adafruit Ultimate GPS HAT for Raspberry Pi Kit. The HAT is based on Adafruit's line of Ultimate GPS products, so there is a long history of reliability and ease of use. There is plenty of prototyping space for some additional components, a battery for the real-time clock (RTC), and an internal antenna.

One thing to keep in mind: this HAT uses the Raspberry Pi's hardware UART to send and receive data to the GPS module. There are workarounds if you need to use the RX and TX pins. Head on over to the Adafruit site for more information and a comprehensive tutorial on how to use this HAT. The board comes mostly assembled. You will have to solder on the included 2×20 GPIO header before you can connect it to your Raspberry Pi. Don't worry, it's really easy. □

Adafruit Ultimate GPS Breakout

Maximum flexibility and control

ADAFRUIT ◈ **$39.95** | adafruit.com



Right Small and powerful GPS

Adafruit sells a few different GPS modules, but it was never totally satisfied with all their features. This led the firm to develop the Ultimate GPS Breakout board. It started by designing the board around the MTK3339 chipset. This low-power module can track up to 22 satellites on 66 channels, features a high-sensitivity receiver and a built-in antenna, all in one small package.

In addition to using a great GPS module, Adafruit also added an ultra-low dropout 3.3 V voltage regulator, and an enable pin so you can turn off the module using almost any microcontroller pin or switch. There is also space for adding an optional CR1220 coin cell to keep the RTC running, which allows for warm starts and a small LED for simple status updates. And to our surprise, you can even store a small amount of positional data to the microcontroller inside the MTK3339. That's really handy! And in typical Adafruit fashion, there's a great online tutorial to get you up and running fast. □

VERDICT

Just what you need for almost any GPS project.

10/10

SparkFun GPS Breakout – XA1110

Get started qwiic-ly

SPARKFUN ◈ $49.95 | sparkfun.com

Right
Small and simple to connect to your Qwiic-compatible boards

The SparkFun XA1110 GPS Breakout board is a small-form-factor I²C-based module that was designed to easily integrate into your Qwiic Connect ecosystem. The SparkFun Qwiic Connect System makes prototyping faster and less prone to wiring errors. All Qwiic-enabled boards use a common 1 mm pitch, 4-pin JST connector, which is polarised so you really can't hook it up wrong. Adding this GPS module to your project is as easy as plugging in a cable. In addition to the super-simple hook-up, you can also add an external antenna through the U.FL connector for added signal reliability. There is also an on-board battery for the real-time clock which allows for fast first satellite acquisition times. □

VERDICT

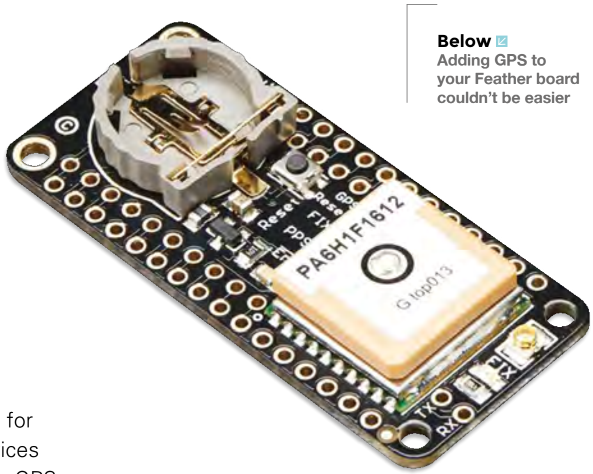
A good choice if you have Qwiic-compatible boards.

9/10

Adafruit Ultimate GPS FeatherWing

Stackable and trackable

ADAFRUIT ◈ $39.95 | adafruit.com



Below
Adding GPS to your Feather board couldn't be easier

VERDICT

If you have a Feather board, this is your best choice for simple GPS.

9/10

If you're already using the Feather board for your project, then adding locational services couldn't be easier thanks to the Ultimate GPS FeatherWing. This add-on board, or wing, is based on Adafruit's Ultimate GPS module, which enables precise and low-power locational information. In addition to your precise location, you can get very accurate time once you acquire a few satellites.

Adafruit added a few extra goodies to make it better than just a simple breakout board. It's capable of low power use and, thanks to a CR1220 coin cell holder, you can store accurate time which allows for warm starts. The board is capable of tracking 22 satellites and has an accuracy of about three metres, which is basically the same as most GPS modules. Check out the product page for more information about compatibility with the specific Feather board you're using, along with some great tutorials to get you up and running quickly. □

> "You can also get very accurate time **once you acquire a few satellites**"

SparkFun GPS-RTK2 Board

You have to pay for precision

SPARKFUN ◈ **$219.95** | **sparkfun.com**

At first glance, you might think this board is a bit overpriced for a GPS shield, and you would be right if it had the same specifications as the other boards in this roundup, but it's different. Very different. In this particular instance, you are paying for positional accuracy. Most GPS modules are accurate within three metres or so, but this one is much more accurate. Thanks to the ZED-F9P module, and real-time kinematics (RTK), the board can sense its position with an accuracy down to 1 cm. That is exponentially better than the competition. However, it's also exponentially more expensive. If you need precision, this is a tough board to beat □

Right ◈
A great addition to your Pi

VERDICT

When you need accuracy, this is the best board for you

9/10

Indoor Navigation Positioning System Starter Set

Satellites can't get everywhere

ROBOTSHOP INC. ◈ **$399** | **robotshop.com**

Sometimes it would be great if you had GPS-like features implemented on an indoor device like a robot. Unfortunately, GPS typically relies on line of sight to gather accurate signals from satellites orbiting the Earth above. And although you can gather some of these signals inside a building, it's usually unreliable at best. Also, most GPS-based systems have an accuracy of about three metres, which is fine for a car or general information about where you are in the world, but it's typically not as useful when working indoors. The Indoor Navigation Positioning System Starter Set from the RobotShop allows for indoor locational services to be implemented with an accuracy of about 2 cm. That's more than enough accuracy to track people or robotics inside a building. The range of each of the beacons is about 50 metres, and each beacon reports to the main router 16 times per second. The starter kit comes with one mobile beacon, four stationary beacons, and one router. When you need GPS-like features in an indoor environment, this kit is something that should be on your radar. □

Right ◈
A system for locational services where GPS typically won't work

VERDICT

Acuurate and works anywhere, but comes at a price

8/10

Can I Hack It?

A set of remote lights?

Can we hack a set of remote lights to control anything?

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 **bigl.es**

YOU'LL NEED

Hyundai 3 × LED Remote Control Lights

COST

£3

WHERE

Alternatives available via Amazon

There are times when we do not need a full microcontroller or processor for a project. Sometimes we just need a simple switch that will bring our project to life. Well, for £3 we have found such a switch. In fact, there are three of them! Let's take a look at a pack of three LED lights which are controlled via an infrared remote. Can we use them in the project? Well, there is only one way to find out: we need to tear it apart!

GENERAL CONSTRUCTION

The LED lights are constructed in sections from easily worked plastic. These sections are all held together by three cross-head screws. When the screws are removed, the unit will naturally fall apart as it has no physical retention tabs. Inside the light, we see a silver plate that fits over the LED board. While it may look like metal, it is not and merely silver-coloured plastic. Removing the plate requires the use of side cutters to remove two plastic 'stakes' that hold both together. Once you've snipped these, we have access to the electronics. Remember the orientation of the stakes and where the screw holes and standoffs are, as they all need to connect to hold the light as a solid unit.

The remote control unit is made from the same plastic, and for such a small remote it has five cross-head screws holding it together. Once removed, the top layer containing the buttons will come away to reveal a plastic sprue, similar to model kits, that could be easily reproduced using CAD software, enabling your own custom buttons and layout.

ELECTRONICS

The electronics for the LED lights are obviously built to cost, and that is evident with the soldering and general neatness of the board. The infrared receiver is the most obvious example in that the legs of the IR receiver have not been snipped off and sit rather too close together. So get your snips and trim the legs! On the board, we have a no-name eight-pin chip, but we can see that the output of the IR receiver connects to pin five of the chip, and that means we can read the signal if we so wished. Pin three connects to the momentary switch on the reverse, and by pulling pin three to GND, we can toggle the LEDs on and off. Power for the LEDs is provided by the AA batteries via a 10 kΩ resistor.

The remote control uses a KXD6122-S1 which, according to its datasheet, is an IR remote-control transmitter circuit. And this is directly connected to the momentary switches via specific pins. The chip will constantly 'scan' the inputs to see if any have been pressed; if that is so, then the pin will receive ~76 mV and this change of state triggers the chip to send a code via the IR emitter. There are four buttons on the remote, but the chip can detect up to 16 inputs.

Right Inside these unassuming plastic packages, there is enough technology to enable anyone to remote-control their projects

MULTIPLE COPIES

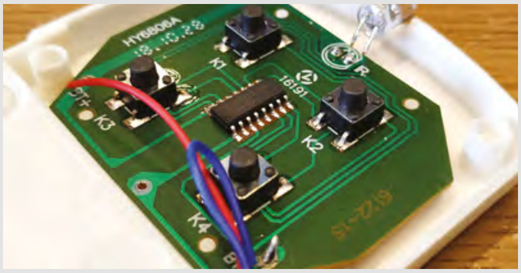
Chances are that, unless you live in the UK, you are unlikely to find this exact product. But despite being sold under a Hyundai brand, it will be made by an original equipment manufacturer (OEM) which will produce similar products for different companies and markets. But no matter the brand on the box, the electronics will be similar enough to hack. Our top tip is to use your multimeter to first trace where the VCC (+) and GND (-) connections are around the board.

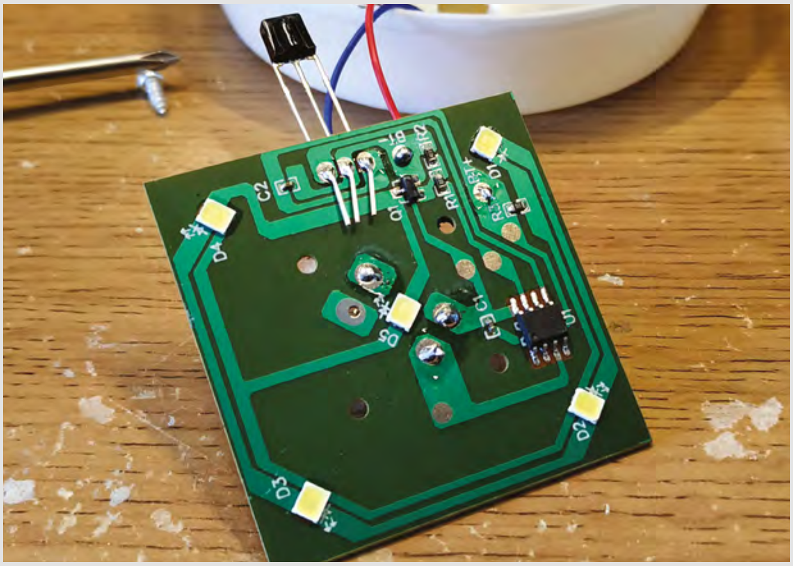
This also requires us to find VCC and GND on the chip; for that, you'll need to use the continuity test function of your multimeter. To do so, put one probe of your multimeter on a GND connection, and the other is gently dragged across the pins until you hear a beep from your meter. Make a note of the pin and its use. Do the same for VCC. Then, trace the connections from the IR receiver to the chip. To test the buttons, set your multimeter to DC voltage, and connect a probe from your multimeter to a button leg, and the other to a leg on the chip. Press the button. Does the voltage change? If so, that means the chip to which it is connected is waiting for a HIGH / LOW signal to trigger an action. If there is no change, move to the next leg until you see a change. Your multimeter is an essential tool that can help diagnose faults and trace paths around any circuit board.

HACKABILITY

For a £3 piece of kit, we can hack this in a number of ways. Let's start with the remote control. We can easily remove the PCB and use it with a board such as an Arduino. The button presses can be replaced with sensors that when triggered will pull the correct pin on the KXD6122-S1 high and trigger the IR emitter to send a signal to the LEDs. 3D-printing new buttons and a case for the board is also a fun activity and means that the board can be easily integrated into cosplay – a remote-controlled arc reactor for an Iron Man costume perhaps?

Below
A simple, functional board with a chip that can do so much more! Easily removed and integrated into your next cosplay project

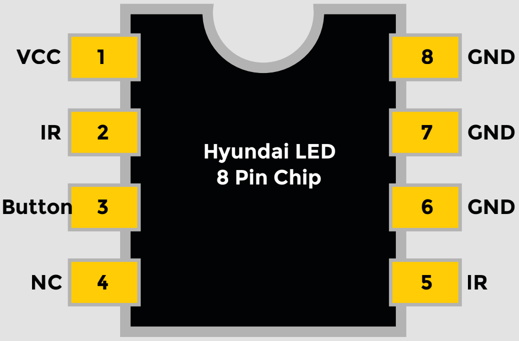




Above
A simple circuit board from which we can learn so much with our trusty multimeter

The LEDs are the easiest part of this product to hack. But the first question to ask is: do we still want the LEDs? If not, then the best option is to desolder them from the board, and this will require a hot-air rework station if you mean to keep the LEDs; if not, brute force and a soldering iron will do the job. The outermost track of the PCB is our GND connection, and the next inner track is VCC which receives power when the chip is triggered by the IR receiver. So we can either integrate the LEDs into a project, or remove them and use their tracks to trigger other devices. We scraped away a little of the tracks and soldered an L298N motor controller along with a simple geared motor which we triggered to life when we turned on the 'LEDs'. So this means we have a simple and cheap remote trigger for projects, including using it with Arduino and other boards.

For £3, we have an IR transmitter capable of sending up to 16 codes, and three receivers that can be hacked into projects with great ease. The PCB for both devices is easy to work with and will only take a few minutes to decipher and hack into something much grander than lighting your cupboards. □



Below
The eight-pin chip for the LEDs has no markings, but after using our multimeter, we quickly found what each pin does

Particle Mesh WiFi bundle

An IoT network in a box

PARTICLE ◆ $89 | store.particle.io

By Ben Everard @ben_everard

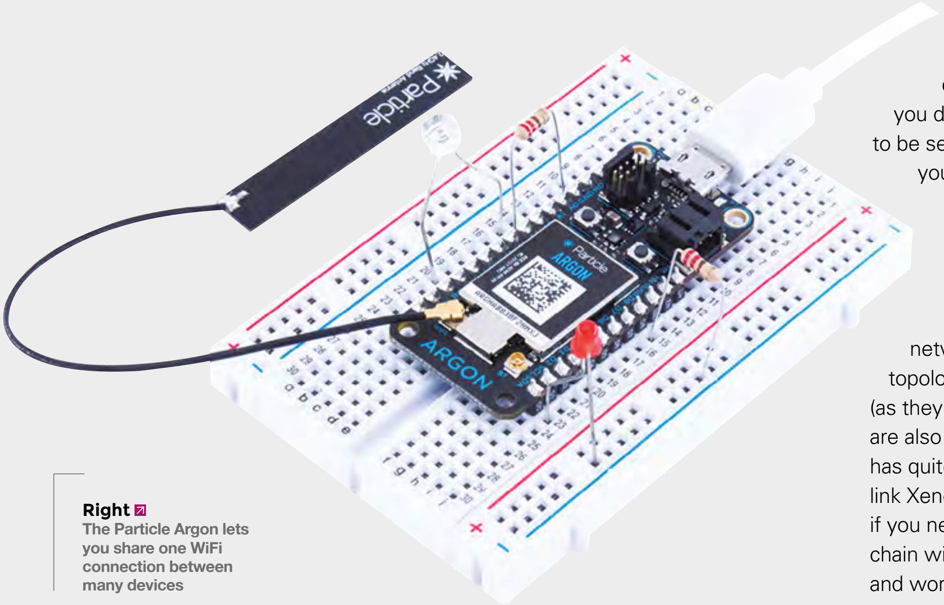
This bundle is made up of kits, one based on the Argon board, and two based on the Xenon board. As well as the dev board, these kits include a breadboard, USB cable, and a few components for getting started (an LED, a phototransistor, and some resistors).

The Xenon (available individually for $15) is Feather-compatible, and based on the Nordic nRF52840 microcontroller, which brings in Bluetooth connectivity. The Argon (available individually for $25) has the same hardware as the Xenon, plus an additional ESP32 which brings in WiFi connectivity. Particle also makes a Boron board which can connect to the mobile phone network (the LTE version is $59, while the 2/3G kit is $69).

From a hardware point of view, these boards bring exactly what you'd expect from Feather boards – they have 19 GPIOs, of which six can also be analogue inputs. There's a USB, and a LiPo charger for power. Thanks to the Feather compatibility, there's a wide range of add-on boards available from the usual internet stockists of hobbyist electronics parts.

While the hardware works as expected, it's the software that really makes these boards stand out. Firstly, there's the firmware that lives on the boards, known as the Particle OS. Perhaps the most unusual part of this is that it doesn't take data or programs over the USB connection. Instead, it comes via the wireless link – either Bluetooth or WiFi. Initial setup is done via a mobile app where you can scan a QR code on the device, and this connects via Bluetooth to let you do additional setup. The WiFi-enabled Argon has to be set up first, and once this is connected via WiFi, you can move on to the Xenon.

It has to be done in this order because the Xenon connects to the internet via a Bluetooth link with the Argon. This mesh network of devices lets you connect many Xenons to a single Argon and build up your network. There are a few advantages to this topology – the Xenons are cheaper than the Argon (as they don't have the ESP32 WiFi chip), and they are also lower power. While any single Bluetooth link has quite a short range (ten metres or so), you can link Xenon to Xenon and build up a chain of devices if you need to cover a large area. Of course, this chain will rely on all the devices being powered on and working.



Right The Particle Argon lets you share one WiFi connection between many devices

Above The Feather form factor is small and well supported

This firmware and networking connects everything to the Particle Device Cloud – the latter includes a web-based IDE for programming (similar to the Arduino IDE, but there are a few more built-in functions), a management portal, and a messaging system (known as Events in Particle-speak).

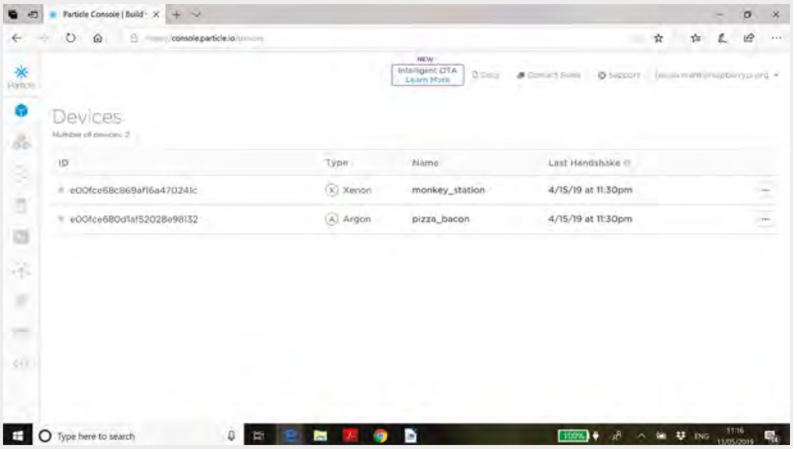
There are a few ways to connect your network of Particle hardware with the outside world via this cloud. You can register functions which automatically create an HTTP endpoint that takes GET or POST requests. You can also access the Events either via the API or by using them to drive webhooks that can link with services such as IFTTT (if this, then that).

> "You can get hardware, connectivity, and middleware running very quickly"

This all comes out-of-the-box with no additional setup beyond connecting the devices to the Particle Cloud, and whatever programming your devices need. We'd imagine that in most cases it will be connecting to some sensor hardware and pushing that data to the Events, or reading the Events and controlling some actuator hardware, but obviously, this depends on your particular project.

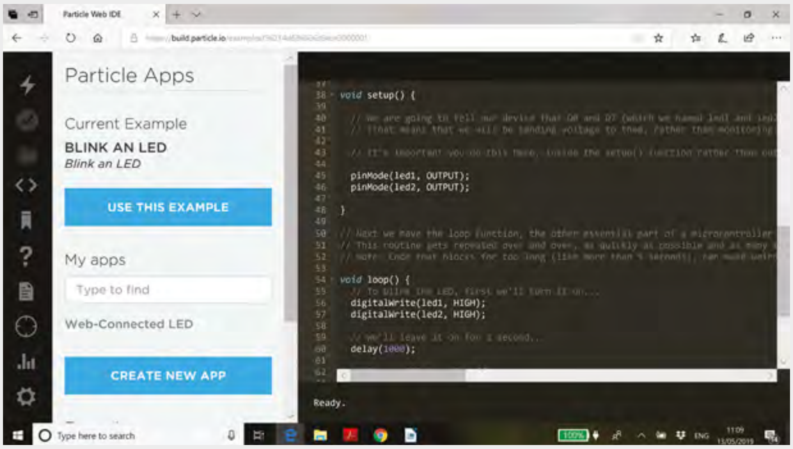
AS A SERVICE

On the free 'Prototyping' plan, you can create up to 100 nodes in your mesh networks, with up to ten internet-connected points (either Argon via WiFi, or Boron via the mobile phone network). This is big enough for most hobbyists to use without having to sign up for the monthly 'per-device' payment plans.



Devices

ID	Type	Name	Last Handshake
e00fce68c869af16a47024lc	Xenon	monkey_station	4/15/19 at 11:30pm
e00fce680d1af52028e98132	Argon	pizza_bacon	4/15/19 at 11:30pm



Particle Apps

Current Example

BLINK AN LED

Blink an LED

USE THIS EXAMPLE

My apps

Type to find

Web-Connected LED

CREATE NEW APP

```
void setup() {

  // We are going to tell our device that D0 and D7 (which we named led1 and led2
  // (that means that we will be sending voltage to them, rather than monitoring

  // It's important you do this here, inside the setup() function rather than out

  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);

}

// Next we have the loop function, the other essential part of a microcontroller
// This routine gets repeated over and over, as quickly as possible and as many
// Note: Code that blocks for too long (like more than 5 seconds), can make weird

void loop() {
  // To blink the LED, first we'll turn it on...
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);

  // We'll leave it on for 1 second...
  delay(1000);
```

```
Ready.
```

If you need a lot of web-connected sensors, the Particle Mesh provides a way to get set up quickly and share a single internet connection between many devices in close proximity. The Feather form factor means that there are already plenty of hardware add-ons available for them, and the enhanced Arduino programming environment means there are drivers too. Combined with the Particle Cloud, it means that you can get hardware, connectivity, and middleware running very quickly. It's not quite all-in-one, as you'll need to handle the data in some way; perhaps kicking off processes via webhooks or reading event data into some system. The Particle IoT Rules Engine fills in a bit more of the puzzle with a Node-RED-based system for processing data, but it's currently in closed beta.

There's currently a plethora of small, connected boards that could work in a similar way; however, they usually leave it up to you to manage the connection to each other and a back-end. The Particle Mesh removes the pain, at a price that's quite impressive, especially if you stay under the ten networks and 100 devices free limit. At the moment, the lack of an out-of-the-box back-end programming environment holds this back from being a complete solution for your IoT network, but it gets you a lot closer than most alternatives. □

Top You can view your connected boards, and check their current status via the online device manager

Above The web-based IDE uploads code to your boards without needing them to be plugged in

VERDICT

Professional-level IoT control at a price tag for hobbyists.

9/10

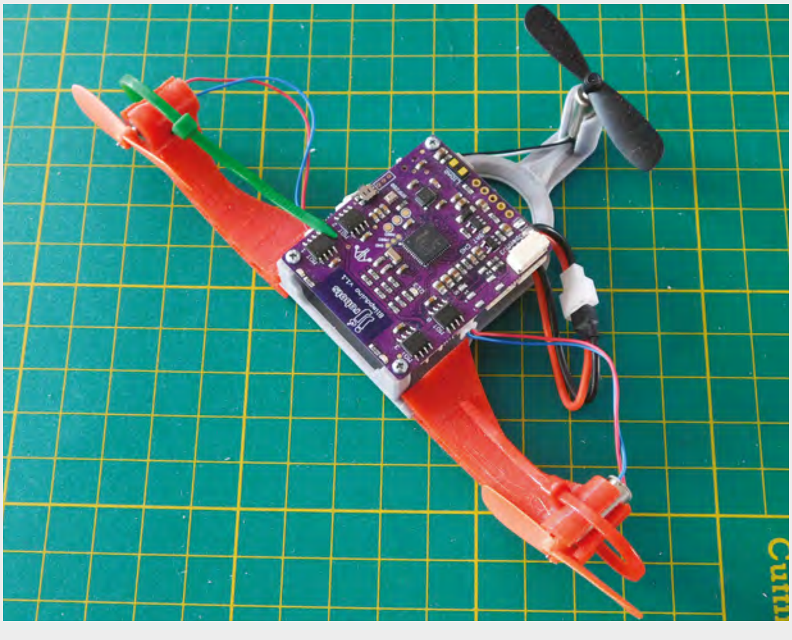
Blimpduino 2

Taking to the skies – the sedate option

JJROBOTS ◆ From £75 | jjrobots.com

By Ben Everard @ben_everard

There are lots of ways to get your projects airborne – we've looked at two options in past cover features: drones and rockets. However, there's a less common, more calm, and controlled option: blimps. These use a lighter-than-air gas (typically helium) in a balloon to lift the weight, then use several small motors driving fans to manoeuvre the aircraft around. Since there's no weight to lift (the blimp should be neutrally buoyant so neither fall nor ascend when floating free in the air), these motors don't have to be particularly powerful, and so a blimp needs much less power than most aircraft. The downside of this setup is that it's much slower and much more susceptible to wind.

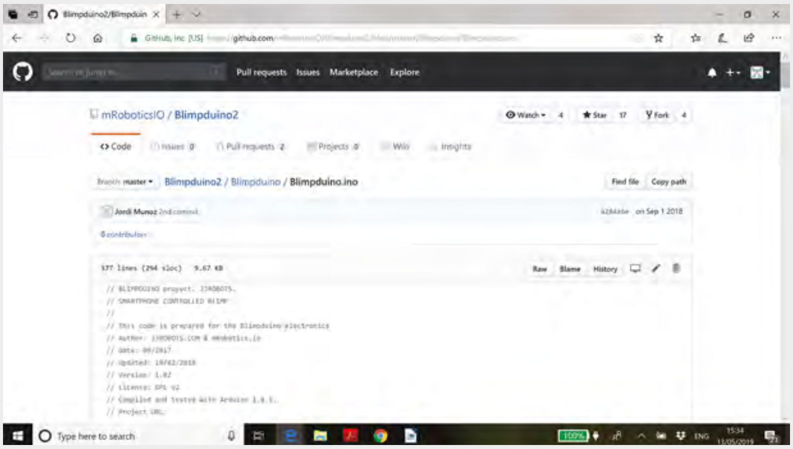


Below ◆ The assembled control structure, ready to be attached to a balloon

The Blimpduino 2 is, as the name suggests, an Arduino-compatible blimp. All the electronic functions are packed onto one board, which has an ARM Cortex-M0, accelerometer, LiPo charging port, and lidar. Additionally, the kit comes with three motors with propellers. You can also order a 3D-printed gondola ($9.12) if you don't have a 3D printer, a LiPo battery ($7.81), and a helium balloon (from $5). The only thing JJRobots doesn't supply is the helium, which you should be able to source from local party shops.

JJRobots recommends that its blimp be flown inside. Although the hardware should continue to work upwards for quite some distance (until the air pressure changes enough to stop the helium lifting enough weight), the lidar distance sensor is only accurate to four metres.

Although the blimp is Arduino-compatible, this isn't actually relevant for the basic operation. Once the hardware's connected up and powered on, you can control it via a phone app. This gives you basic controls for flying. We found that the app didn't work properly on the first phone we tested (there's a thread about this on the forums – **hsmag.cc/uZJmbn** – but as we go to print, this hasn't made it into the published app). However, we were able to get it working on another phone. This gives you a basic, WiFi-controlled remote. You can steer the blimp, and take it up and down. These controls aren't simply connected to the motor output: there is enough intelligence in the Blimpduino 2 to compare your instructions to the received data from the accelerometer and lidar so it will behave intelligently, not just spin out of control.



```
// BLIMPDUINO project. JJROBOTS.
// SMARTPHONE CONTROLLED BLIMP
//
// This code is prepared for the Blimpduino electronics
// Author: JJROBOTS.COM & mRobotics.io
// Date: 09/2017
// Updated: 18/02/2018
// Version: 1.02
// License: GPL v2
// Compiled and tested with Arduino 1.8.5.
// Project URL:
```

'Attempt' is an important word here as it's limited by the hardware as to how effective it will be. For example, if your balloon is poorly balanced, it might not be able to ascend and descend properly.

It's a little hard to give good estimates of how long it will fly for, as the amount of energy it has to expend depends so much on how you fly it. In ideal conditions, it should be able to hold its altitude without expending any energy, so could hover indefinitely. However, you're probably going to be moving it around. We found we got about 20–30 minutes from a 500 mAh battery between recharges, but you may find you get different results.

> "In ideal conditions, it should be able to hold its altitude **without expending any energy**"

LOOKING INSIDE

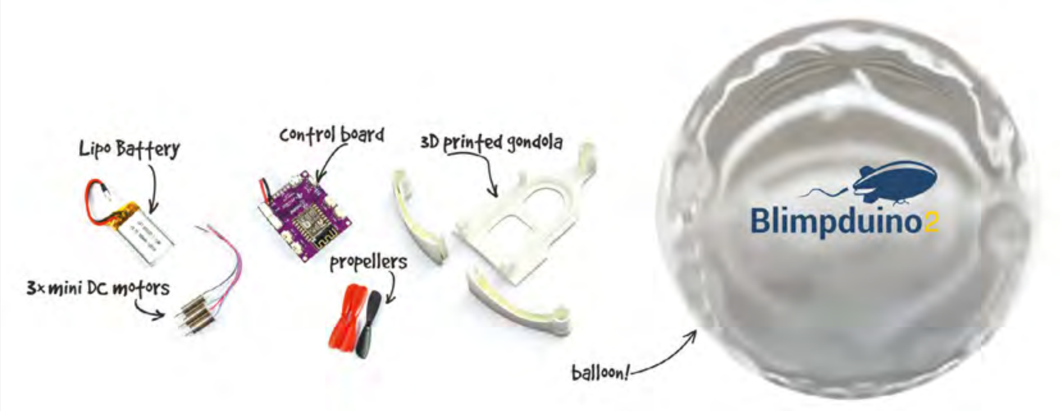
If you want to go further, the firmware behind the device is written in Arduino C++, and the programming port is open, so you can add whatever extra features you like, whether that's some form of autopilot or additional sensors. You can add more weight without it being a burden on the motors, provided you've got enough lift capacity in your balloons. You can add larger (more) balloons, but this will eventually slow down your blimp and make it harder to control.

While there aren't any specific guides to modifying the hardware or software, the source code (available here – **hsmag.cc/SHSBAu**) is commented and reasonably straightforward to follow. Perhaps the most unusual part of it is that the default app communicates via UDP, rather than the more common TCP. This isn't particularly hard to use, but it's less likely to be familiar to users. There is no need to stick with this communication method – you've got full control of the hardware, and can communicate with it however you like. The only slight issue is that you don't have access to the programming port on the ESP8266 used for WiFi, so you'll need to be familiar with the commands for this if you want to modify the wireless setup.

We really enjoyed how adaptable the Blimpduino 2 is. In its most basic use, it's a remote-controlled blimp (some assembly required). You can switch out the balloon and 3D-print replacement parts if you need, and you don't need any particular technical skills to get to this level. However, if you like to tinker with things, it's easy to dive further in, and everything's accessible to you. You can add other things that don't require programming, such as an off-the-shelf FPV camera/transmitter module (these just need power and will transmit automatically without programming). If you do want to go further, there is the hardware to support it. □

Above Go forth and code your own blimp – take and modify the code from hsmag.cc/cvIMBZ

Below The full pack – just add helium!



VERDICT

A fun flying machine, suitable for all abilities.

8/10

Boxes.py

Parametric laser-cut designs

FLORIAN FESTI ◈ Free | **hsmag.cc/WNfdMR**

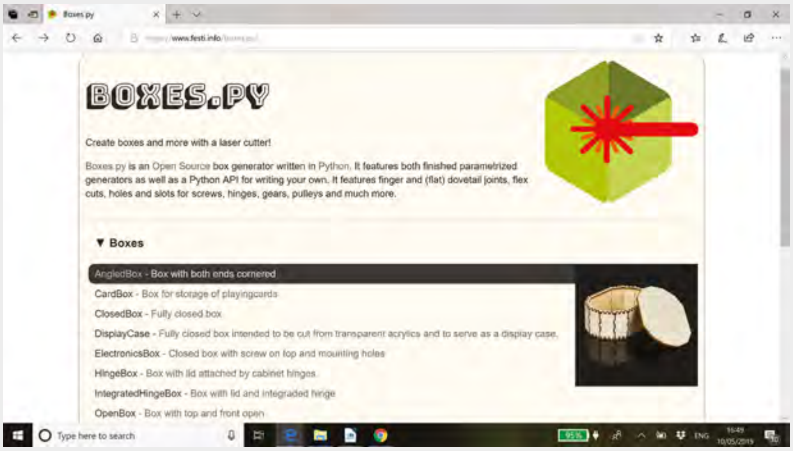
By Ben Everard @ben_everard

Laser cutters are fantastically capable machines. They can cut and etch sheets of material quickly and accurately, allowing you to build up structures far faster than with other desktop fabrication techniques. However, the design process isn't quite as straightforward. You don't just need a solid representation of the device you want to make, you need a structure of it laid out as flat sheets that can be joined, and some form of joint for holding them together. Depending on exactly what you want to make, this varies from the simple to the fantastically complex. Fortunately, there are lots of tools we can use to sidestep this entire design process.

> **Overall, this is a great repository of** parametric laser-cut designs

Parametric designs allow us to customise an object to our needs. For example, a parametric box may let you enter the width, height, and depth, and then output the files needed to cut a box. Boxes.py hosts a wide range of parametric designs for different structures, that you can tweak to your needs. As the name suggests, there are lots of boxes, but also trays, pulleys, gears, robots, and a miscellany of other parts. It's not completely exhaustive in any area (though the range of different box designs is quite extensive), but does provide a good first place to check when you're looking for parametric laser-cutting patterns.

It's available as a webpage, standalone program, or Inkscape plugin (see **hsmag.cc/ESYUiF** for details). We found the webpage easiest to use. It produces SVGs that you can download and import into your laser cutter software of choice. Depending on exactly how you drive your laser cutter, you may find that you have to adjust the colour or width of the lines on the SVGs in order for them to cut properly.



BOXES.PY

Create boxes and more with a laser cutter!

Boxes.py is an Open Source box generator written in Python. It features both finished parametrized generators as well as a Python API for writing your own. It features finger and (flat) dovetail joints, flex cuts, holes and slots for screws, hinges, gears, pulleys and much more.

▼ Boxes

AngledBox - Box with both ends cornered
CardBox - Box for storage of playingcards
ClosedBox - Fully closed box
DisplayCase - Fully closed box intended to be cut from transparent acrylics and to serve as a display case.
ElectronicsBox - Closed box with screw on top and mounting holes
HingeBox - Box with lid attached by cabinet hinges
IntegratedHingeBox - Box with lid and integraded hinge
OpenBox - Box with top and front open

Above ◈ Some of the boxes have example images to let you know what you're getting

All the boxes are held together with finger joints which do work well, but it would be nice to have a few more options, particularly T-slots for cases that need to be put together and taken apart (these are available in the Python library, but don't exist in any of the generator designs).

We printed out several designs including a 'ShutterBox' (which most British people will know as a bread bin), an 'UnevenHeightBox', and a 'TwoPiece' box. All cut well and fitted together securely. However, they don't come with instructions. This isn't a problem for the simple designs, but some of the more complex options (such as the 'FlexBox') can end up resembling 3D jigsaw puzzles.

Overall, this is a great repository of parametric laser-cut designs, but we'd appreciate a few more options for joint types. □

VERDICT

Lots of easy-to-use, laser-cuttable designs.

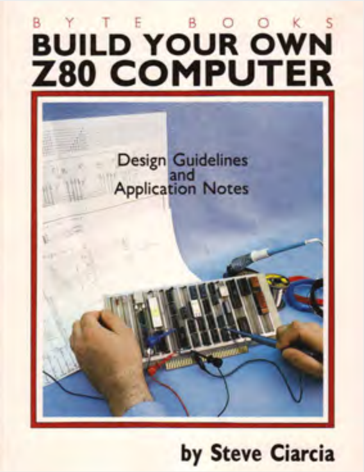
8/10

Build your own Z80 computer

Steve Ciarcia ◆ £54.22 | hsmag.cc/CqLIfH

By **Ben Everard** @ben_everard

In 1981, when this book was first released, the Z80 processor was already five years old, but it'd still be included in commercial products for, well, at least another 38 years (check back next year to see if it's still going strong). Part of its appeal is that it's a well-understood chip that's simple enough to be easy to use, yet complex enough to be useful. At this sweet-spot, it's a great choice for learning the nitty-gritty of how computers work. That's why we looked at building a Z80 computer in issues 7 and 8, but this book goes into far more detail.



In *Build Your Own Z80 Computer*, Steve Ciarcia takes the reader through everything they need to know to build their own computers. When we say everything, we mean it. The book starts with a detailed guide on building your own power supply and voltage regulator – integrated regulators were still new in 1981, and many people preferred to build their own.

Moving on, the computer is built up mostly from logic circuits, allowing the reader to really understand what's going on at the most basic level.

While the final machine isn't as fully featured as most people would expect a computer today – it's probably more analogous to an Arduino or other microcontroller than a desktop. However, it does work and can be used – the book even includes a basic operating system written in assembly language.

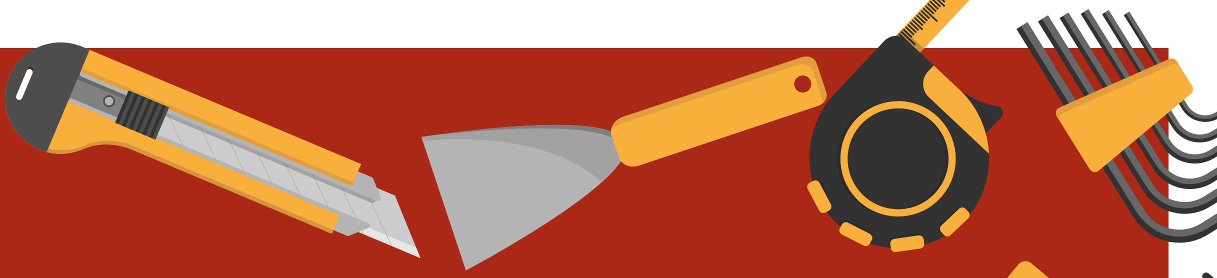
Although building a Z80 machine might be a niche hobby, this book covers the low-level details of how most computers work, even today, and it is an enjoyable read, even if you're not planning on actually building a retro machine. The book is still available from large internet retailers, but if you aren't able to get hold of a physical copy, the digital version is available for free from the following link: **hsmag.cc/u0RLpA.** □

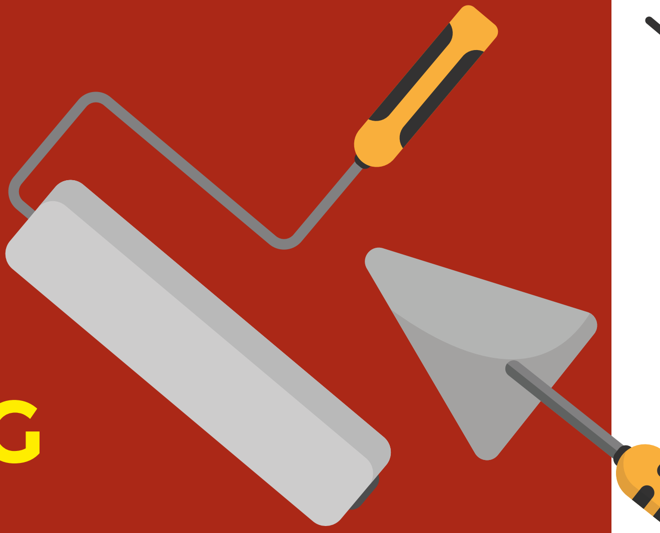
VERDICT

A nostalgic, and informative, look at how computers used to be made.

9/10

next month



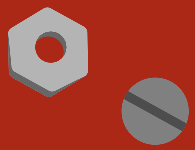


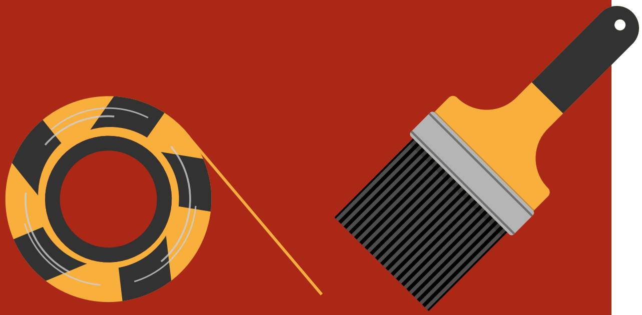
issue #20

ON SALE
20 JUNE

FEATURING

TOP MAKER TIPS





ALSO

- → GARDENING
- → WEAVING
- → ELECTRONICS
- → ARDUINO
- → AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

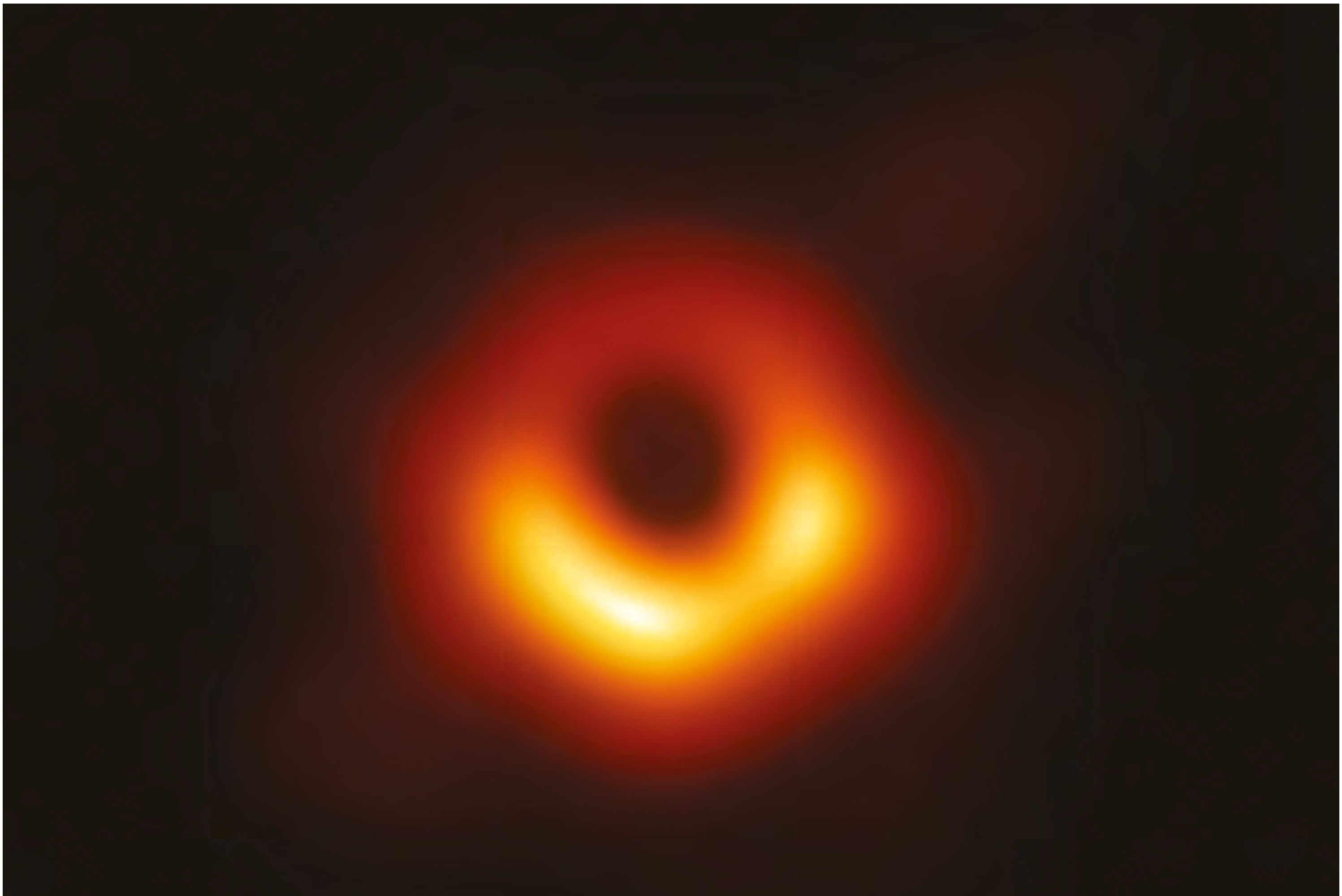


Image EHT

Very long baseline interferometry (VLBI) is a technique that uses multiple telescopes to image an object at the same time, to act like a single giant telescope. The Event Horizon Telescope (EHT) team used telescopes as far apart as Spain and the South Pole to create a virtual telescope almost the same size as the Earth to create this image of the M87 supermassive black hole, which is 6.5 billion times the mass of our Sun and 53 million light years away.

The ring of light around the centre is glowing particles accelerating as they're pulled into the black hole. This ring looks lopsided because the intense gravity of the black hole bends light around it, meaning that the part spinning towards us is brighter than the part spinning away – this further verifies Einstein's Theory of General Relativity.

goBILDA.com
goBILDA.com
goBILDA.com