MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | March 2019 | Issue #16

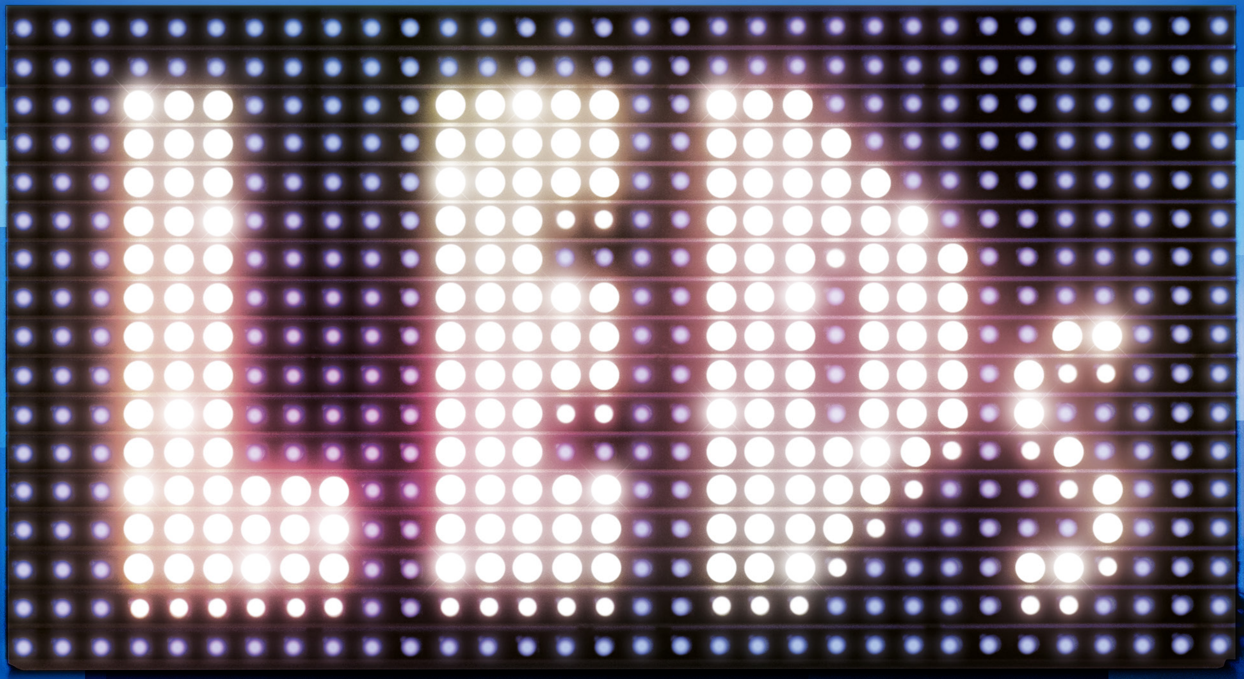
ACCESSIBLE CONTROLLERS

Helping disabled gamers join the fun

Extreme Electronics

Tesla coils, sparks, and high voltage

MAKE ART WITH



DRONE MAPPING

Taking to the skies to find the streets

Mar.2019
Issue #16 £6

Raspberry Pi PRESS

PLOTTER DRAWING

Beautiful pictures created with motorised pens

ARDUINO **SPONGES** AMPLIFIERS **RESIN**

Antennas
Development Systems
Cellular, M2M
Expansion Boards
Zigbee/Thread
Cables/Connectors
Sensors
Wi-Fi
Digi-Key® ELECTRONICS
LP-WAN
Bluetooth
Transceivers

Welcome to HackSpace magazine

There seems to be something in human nature that likes to categorise things. We put pieces of knowledge, or processes into boxes and name these boxes things like science, technology, engineering, and maths. However, these aren't distinct subjects, and there's plenty of overlap between them. The universe is far more chaotic than our neat, ordered brains would like to believe.

In this issue, we're looking at art, an often misunderstood subject. The very essence of it is that objects should mean something to you – they should evoke emotions. For centuries, we've let paint-splatterers and pencil-wavers dominate the field, and they created art that meant something to them. It's now our turn – technologists, scientists, engineers, and tinkerers – to step up onto the stage and create art that means something to us. LEDs create a great medium for this – they're flexible, easy to use, and look awesome – but this is just one approach. The important thing isn't what you're using to create art, it's that you're doing it.

> **The universe is far more chaotic than our** neat, ordered brains would like to believe

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

Available on the App Store

GET IT ON Google Play

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

Photography
Fiacre Muller, Rich Hogben

CONTRIBUTORS

Lucy Rogers, Andrew Huang, Cameron Norris, Andrew Lewis, Mayank Sharma, Dave Astels, Jo Hinchliffe, Graham Morrision, Matt Bradshaw, Andres Muniz-Piniella, Sophy Wong, Les Pounder, Marc de Vinck, Richard Smedley, Gareth Halfacree

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

DISTRIBUTION

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

SUBSCRIPTIONS

Mann Enterprises Ltd,
Unit E, Brocks Business
Centre, CB9 8QP
hsmag.cc/subscribe

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.

Contents

120

Tutorial

Plotter drawing

96

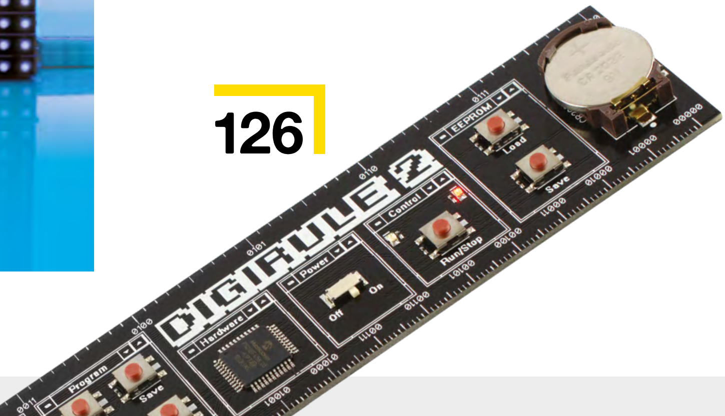
To reveal art and conceal the artist is art's aim – Oscar Wilde

Cover Feature

Create something beautiful with silicon, electricity, and your endless imagination

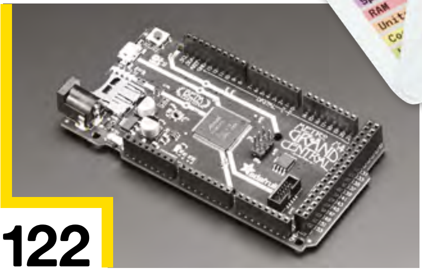
34

126



128

100



122

Direct from Shenzhen

Resin casting kit

112 Encase tiny things in resin for less than a fiver

20

Interview

Jasmine Brackett

60 Why people like to buy from people, not big companies

73 FORGE

111 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.

Composition #11

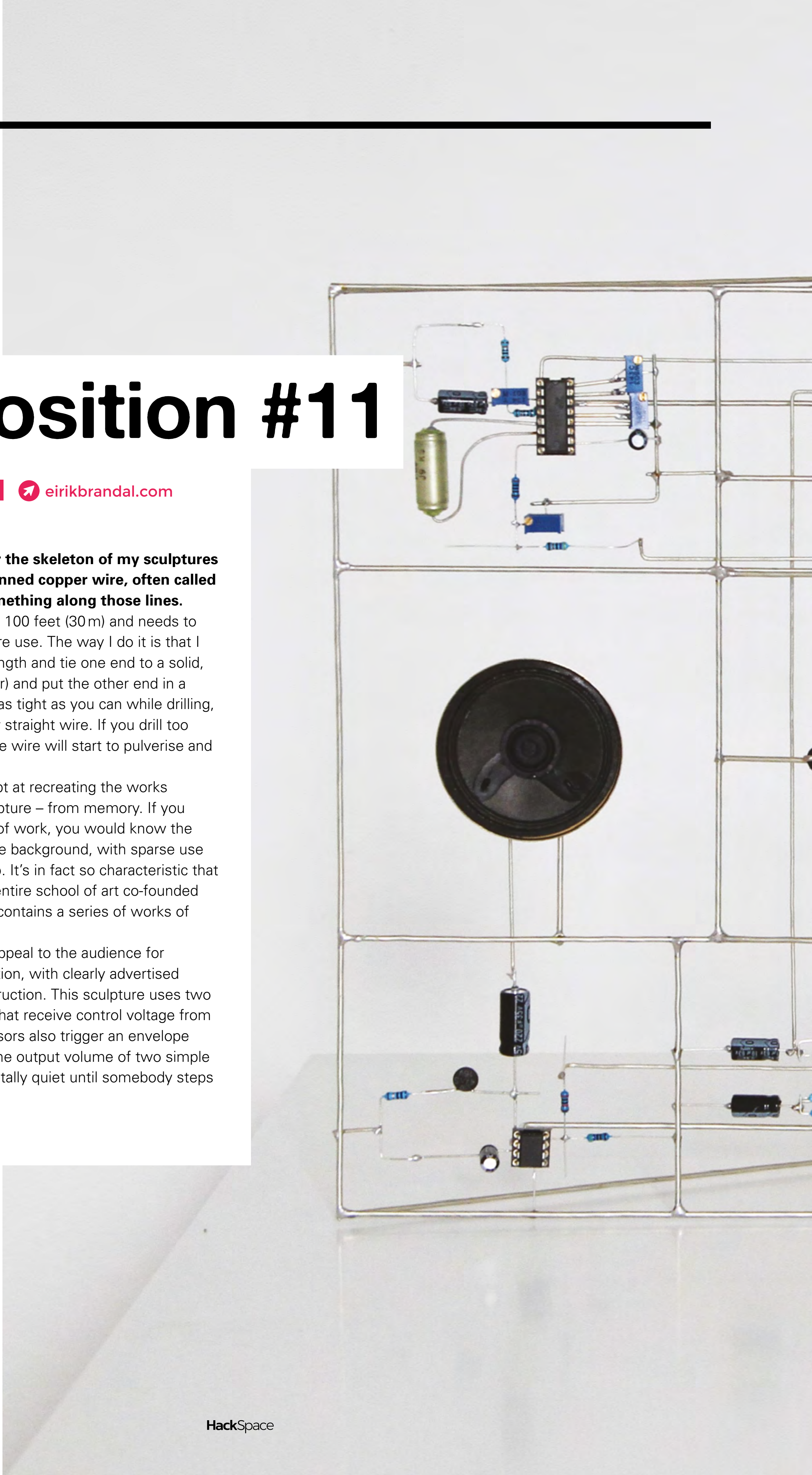
By Eirik Brandal

eirikbrandal.com

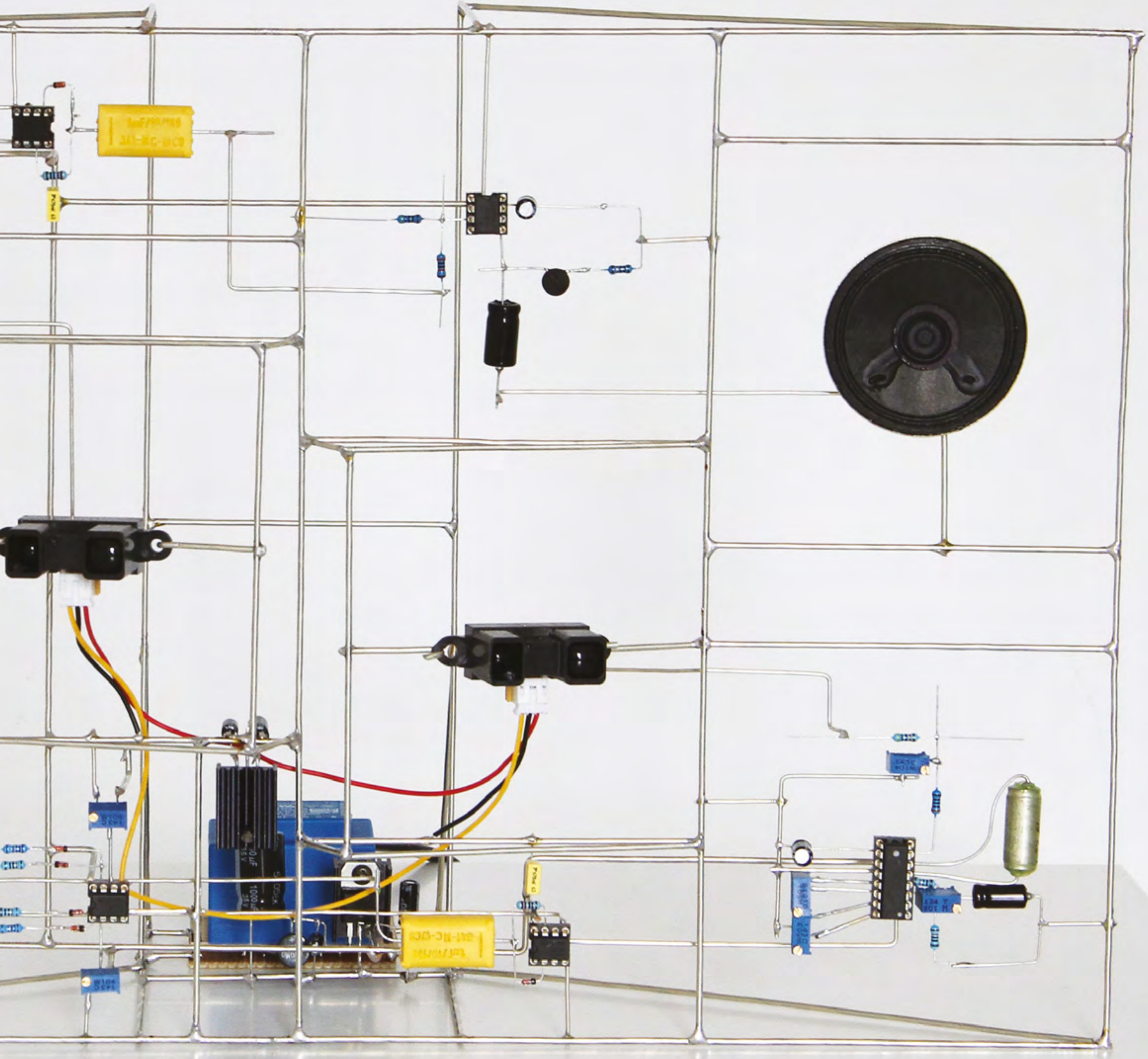
The material I use for the skeleton of my sculptures is 14 and 18 AWG tinned copper wire, often called bus bar wire, or something along those lines. This comes in rolls of 100 feet (30 m) and needs to be straightened before use. The way I do it is that I cut off the desired length and tie one end to a solid, heavy object (in my case, a radiator) and put the other end in a drill. If you keep your copper wire as tight as you can while drilling, you should end up with a perfectly straight wire. If you drill too much, however, the tin layer on the wire will start to pulverise and become weakened.

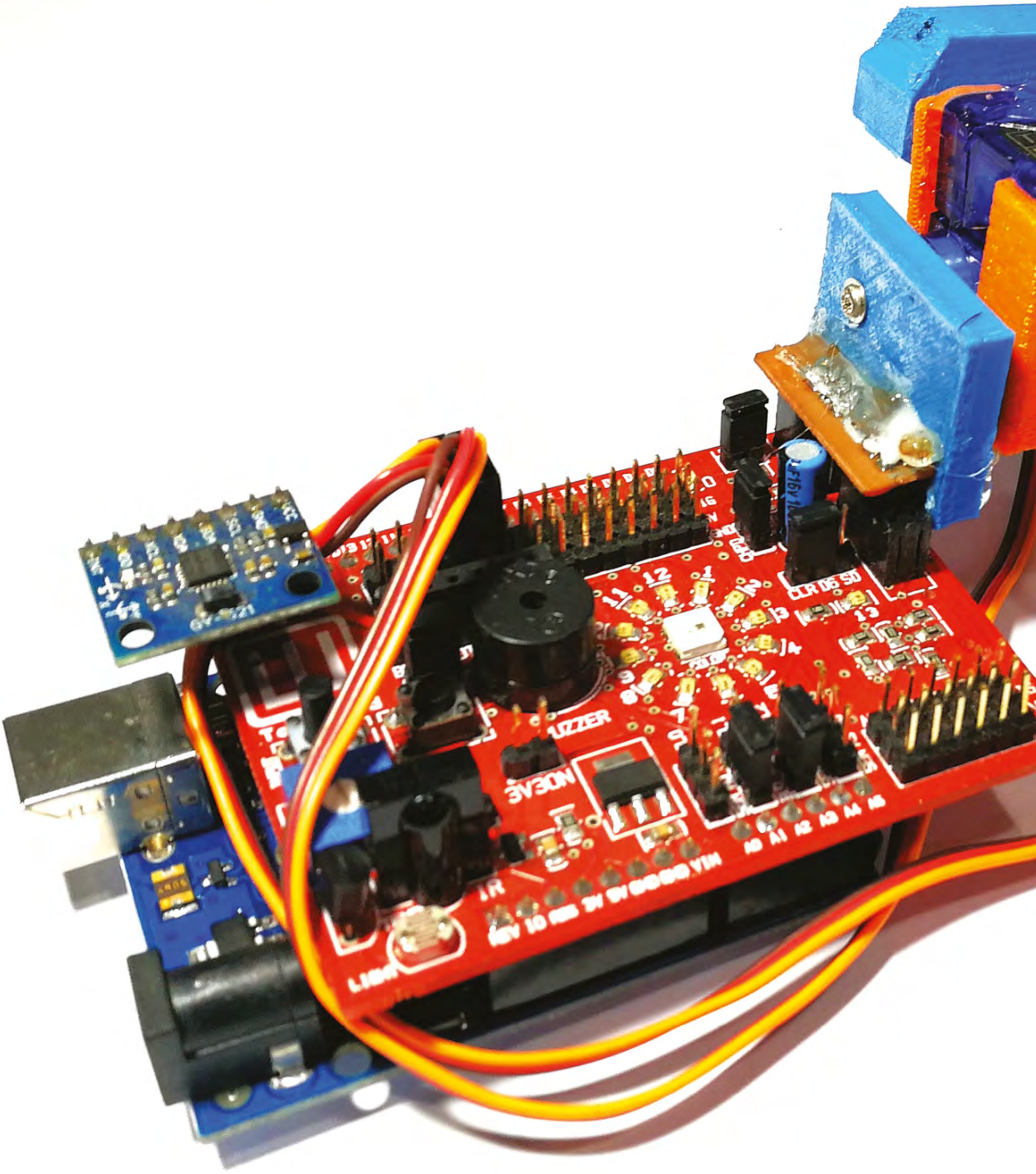
Composition #11 was an attempt at recreating the works of Piet Mondrian into a single sculpture – from memory. If you are familiar with Mondrian's body of work, you would know the characteristic black lines on a white background, with sparse use of primary colours sprinkled on top. It's in fact so characteristic that you could speculate whether the entire school of art co-founded by Mondrian, named De Stijl, just contains a series of works of highbrow plagiarism.

Composition #11 is tailored to appeal to the audience for interaction and individual composition, with clearly advertised sensors at the centre of the construction. This sculpture uses two XR2206 function generator chips that receive control voltage from two proximity sensors. These sensors also trigger an envelope generator, which in turn controls the output volume of two simple VCAs, so the sculpture is in fact totally quiet until somebody steps in front of it. □



Right Eirik originally studied music composition, and his work has shifted toward electronic sound sculptures, looking to uncover the intrinsic beauty of circuits





Assistive spoon

By CuriosityGym hsmag.cc/jHHMIS

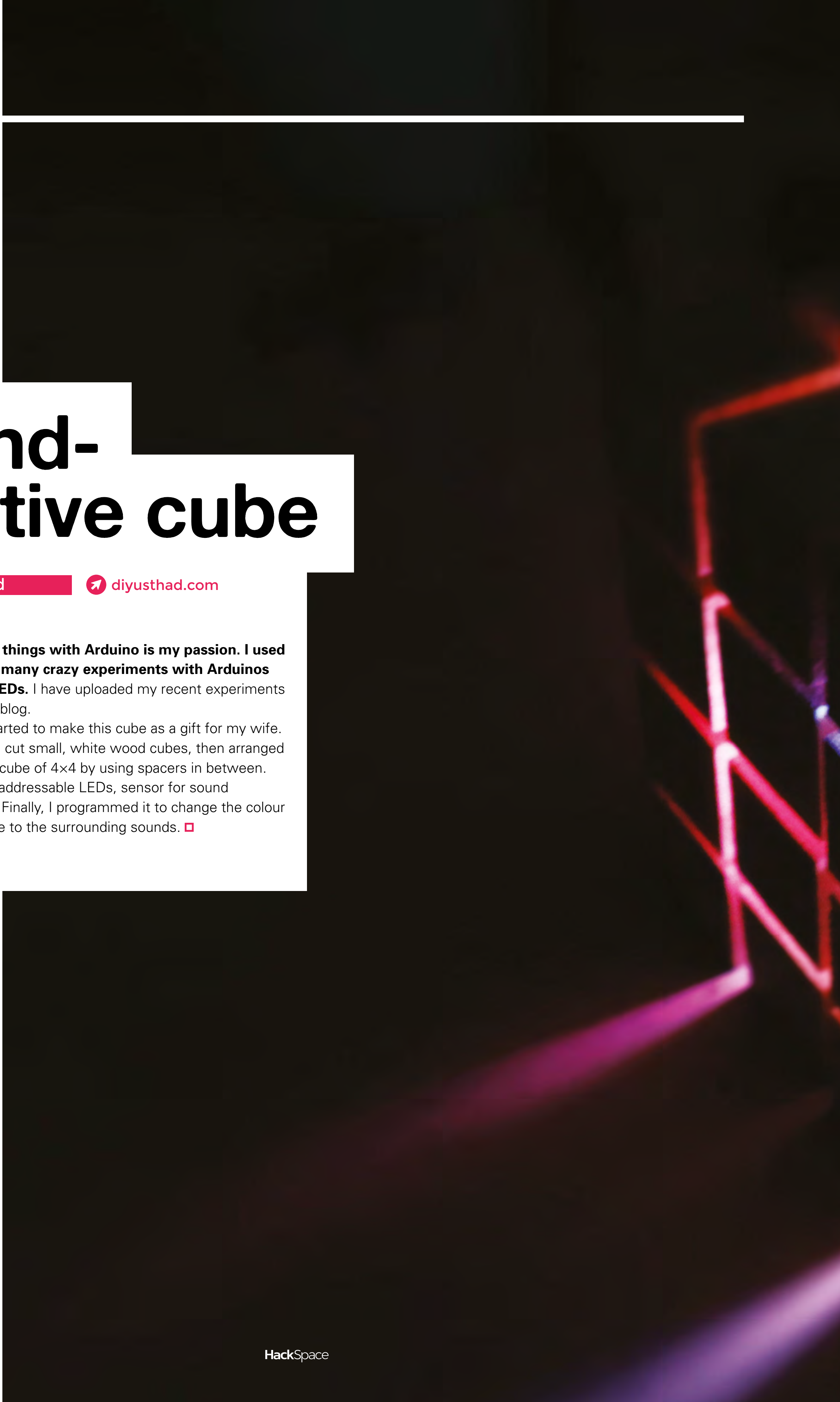
Parkinson's disease is a progressive nervous system disorder that affects movement. Symptoms start gradually, sometimes starting with a barely noticeable tremor in just one hand. Tremors are common, but the disorder also commonly causes stiffness or slowing of movement.

We were greatly inspired by Liftware Steady (see **liftware.com/steady**), which is a product sold for this very purpose.

We realised early on that this was expected to be a challenging project and would take the team into the arenas of motion control, physics, and 3D maths that we had a working knowledge of, and this project helped us put our theory into practice.

The design was expected to be held in hand, and be able to help cancel out any tremors the person's hand felt, thus providing the ability to level and steady out a spoon held at the end of the prototype. □

Left The project was made by CuriosityGym team members Siddhesh Murudkar, Rupin Chheda, and Jehangir Khajotia



Sound-reactive cube

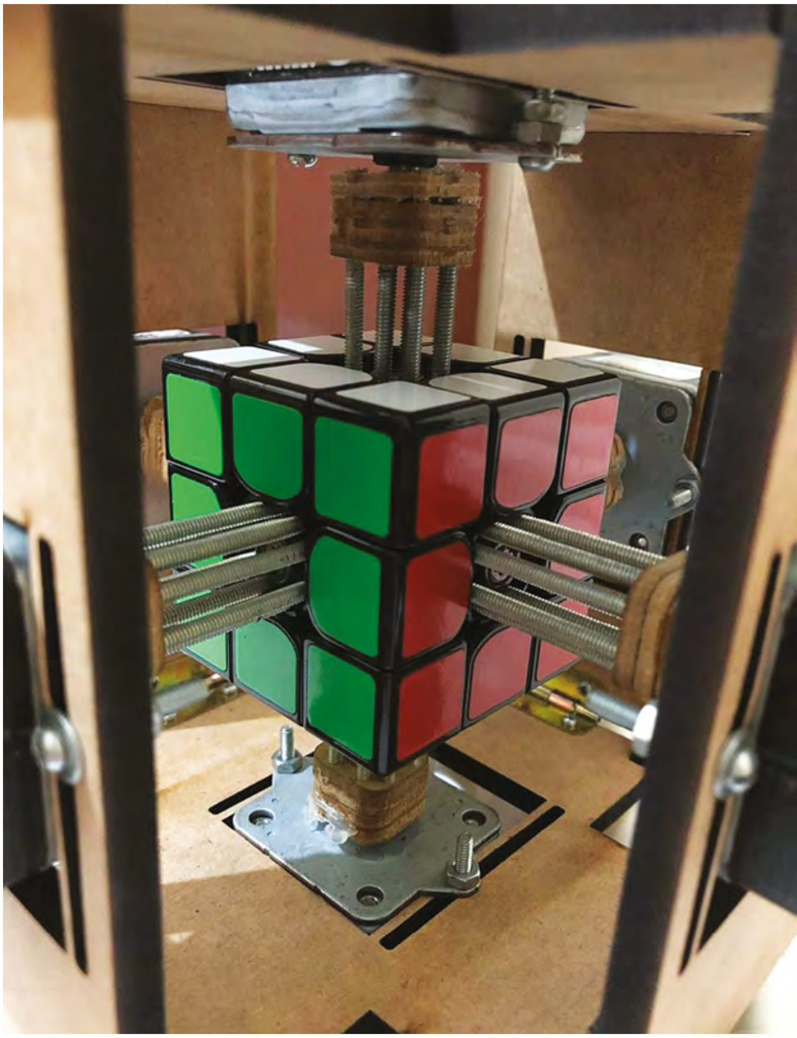
By Mohamed Najad

diyusthad.com

Making things with Arduino is my passion. I used to do many crazy experiments with Arduinos and LEDs. I have uploaded my recent experiments in my blog.

I started to make this cube as a gift for my wife. First, I cut small, white wood cubes, then arranged them in way to make a cube of 4×4 by using spacers in between. Then, inside I fixed the addressable LEDs, sensor for sound detection, and Arduino. Finally, I programmed it to change the colour of each LED in response to the surrounding sounds. □

Right You can find the complete build tutorial at: hackster.io/najad



Rubik's Cube solver

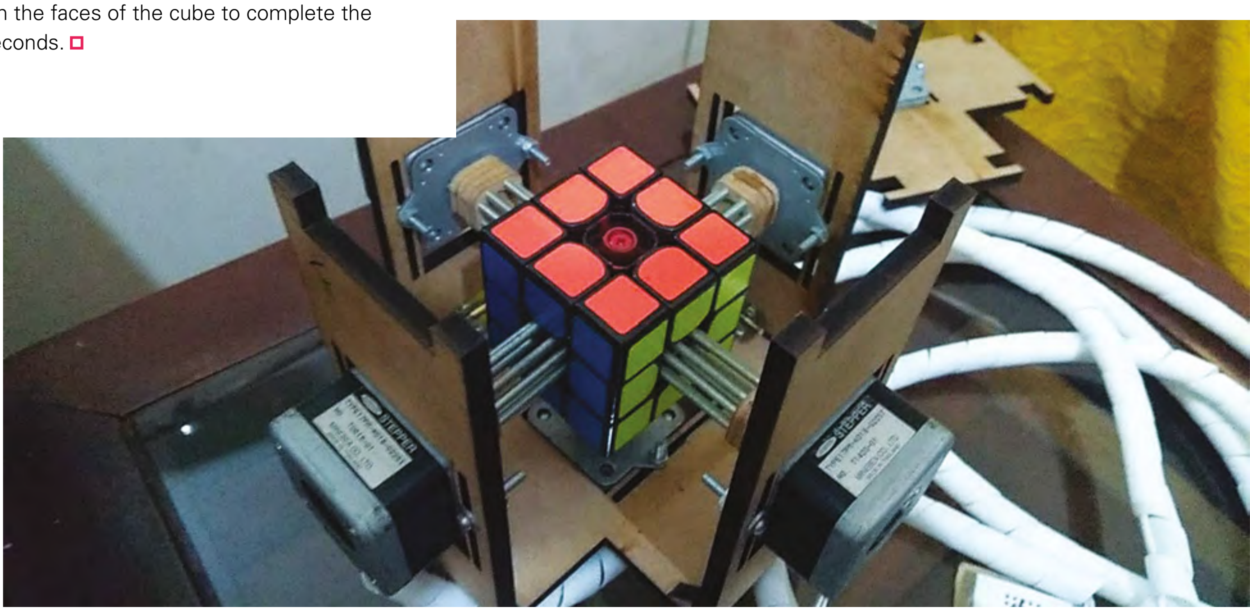
By **André Angelucci**

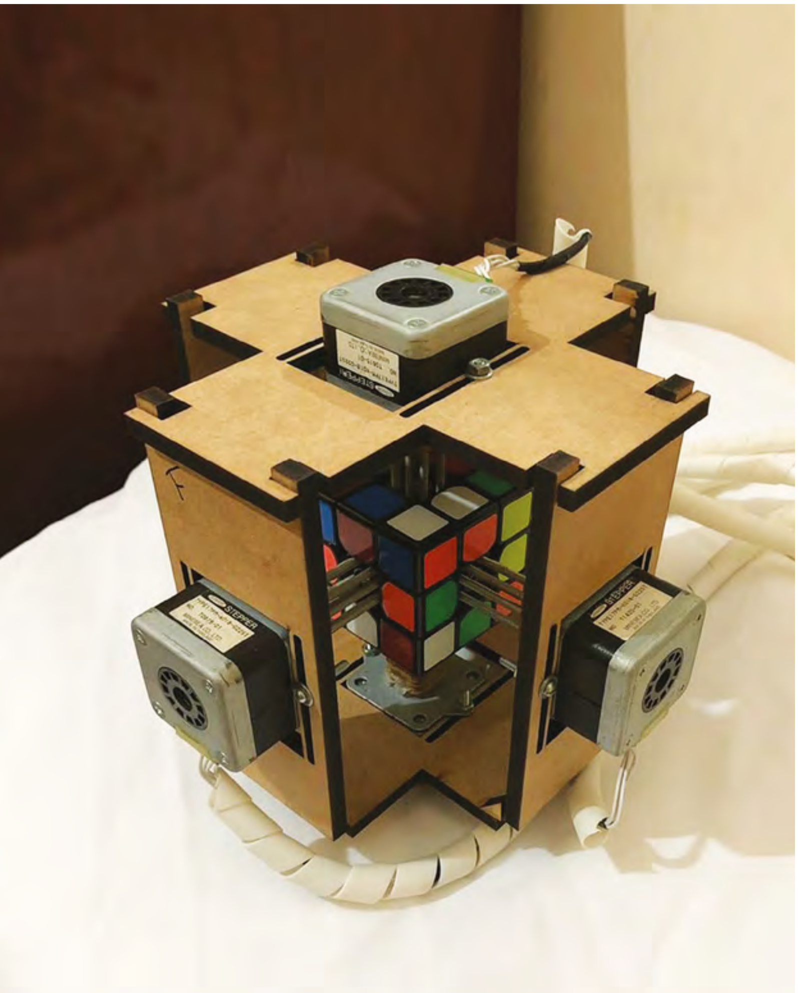
hsmag.cc/mJpcxn

Just like computing and robotics, the Magic Cube is a puzzle that involves reasoning, geometry, calculations, and dedication. That motivated me to develop a robot, as the work of completing my degree in Computer Engineering, capable of solving it.

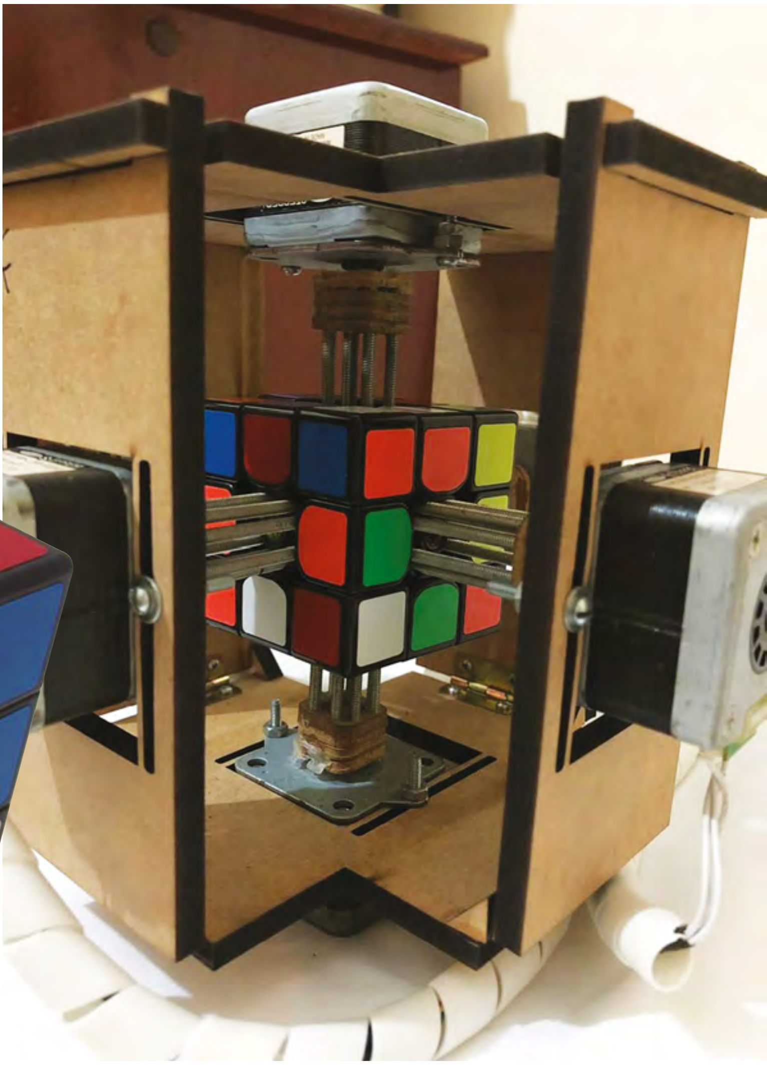
Basically, its operation involves software responsible for calculations, and a robot that physically solves the puzzle.

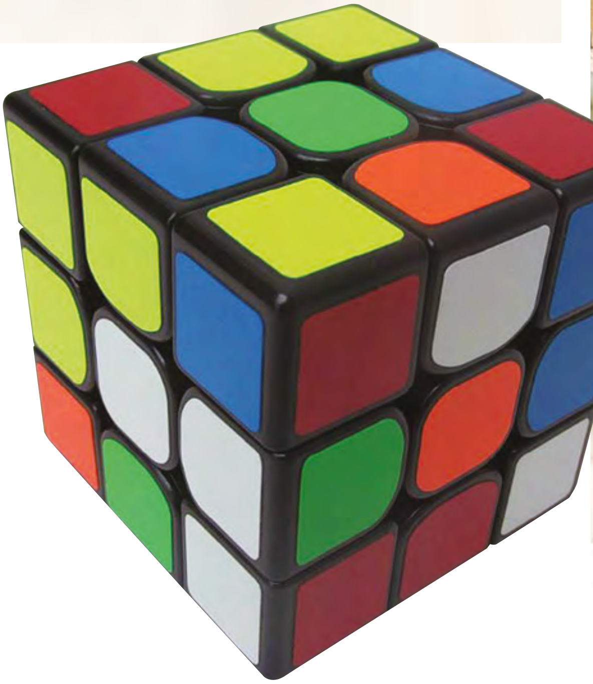
First, the software captures photos of each face of the cube and identifies the colours, through IBM's artificial intelligence API, Watson. With the recognised colours, a calculation is made that results in a sequence of approximately 15 movements that solves the puzzle, using Herbert Kociemba's two-phase algorithm. The solution is sent to the Rubik's Bot, the name of which is a tribute to the creator of the puzzle. An ESP32 microcontroller receives the message, and distributes the motions between six NEMA 17 stepper motors which, through iron rods, physically perform the rotations on the faces of the cube to complete the solution in about five seconds. □





Left
What next for our laser-cut overlords?





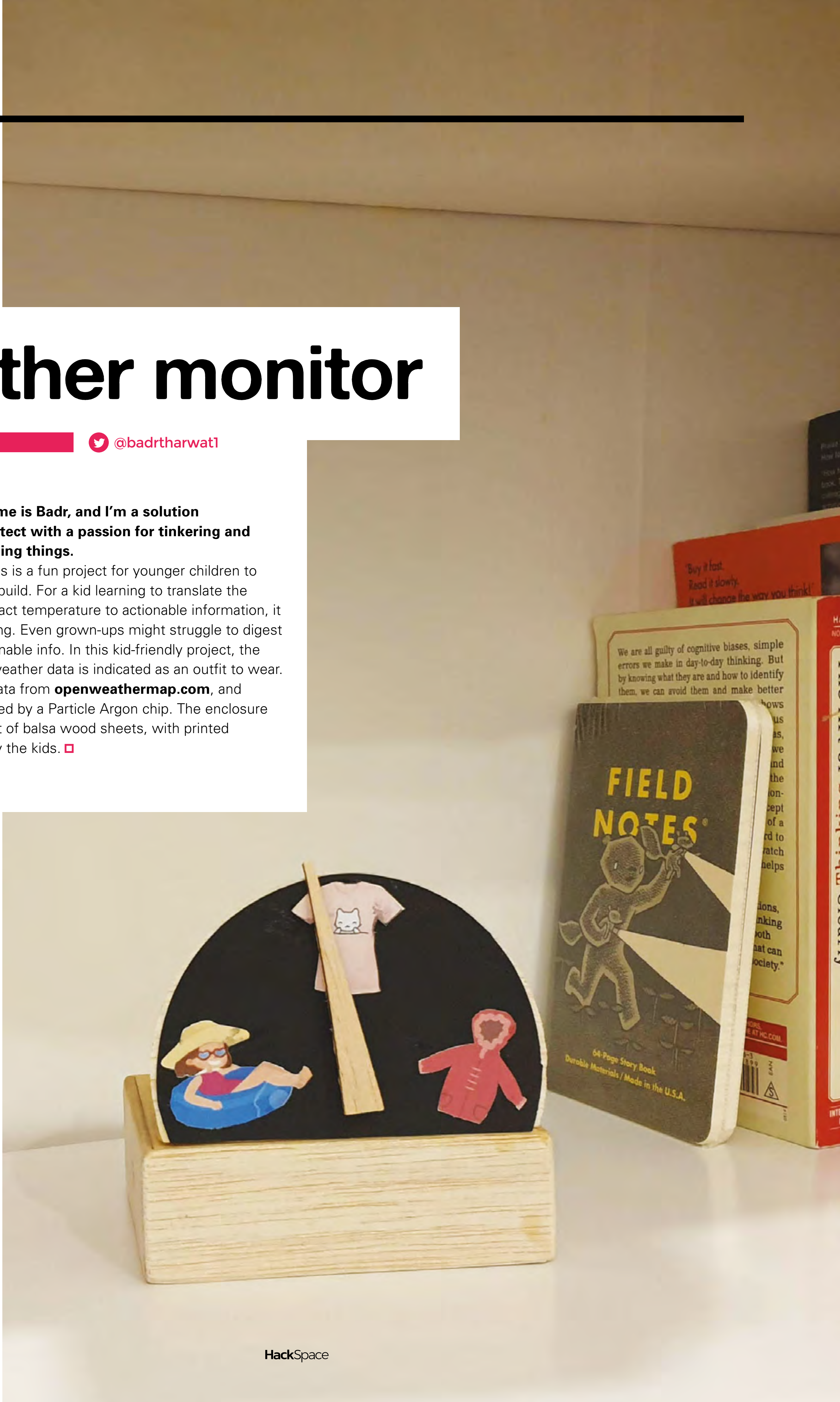
Weather monitor

By **Badr Tharwat** @badrtharwat1

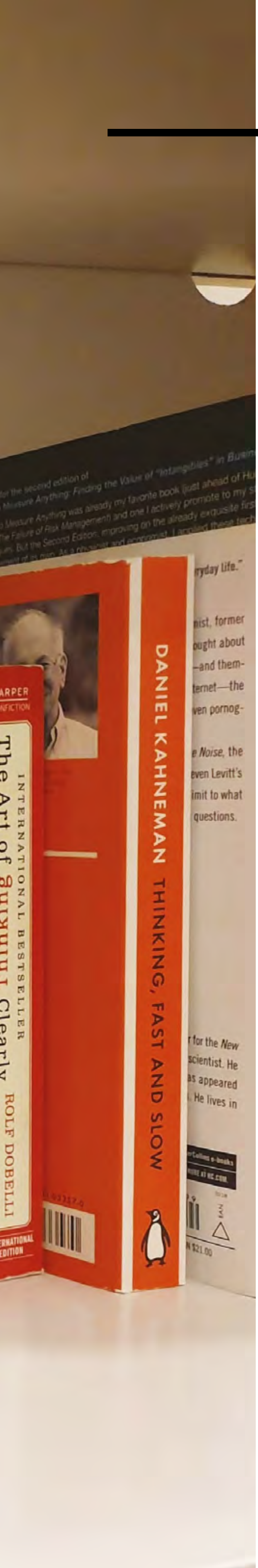
My name is Badr, and I'm a solution architect with a passion for tinkering and building things.

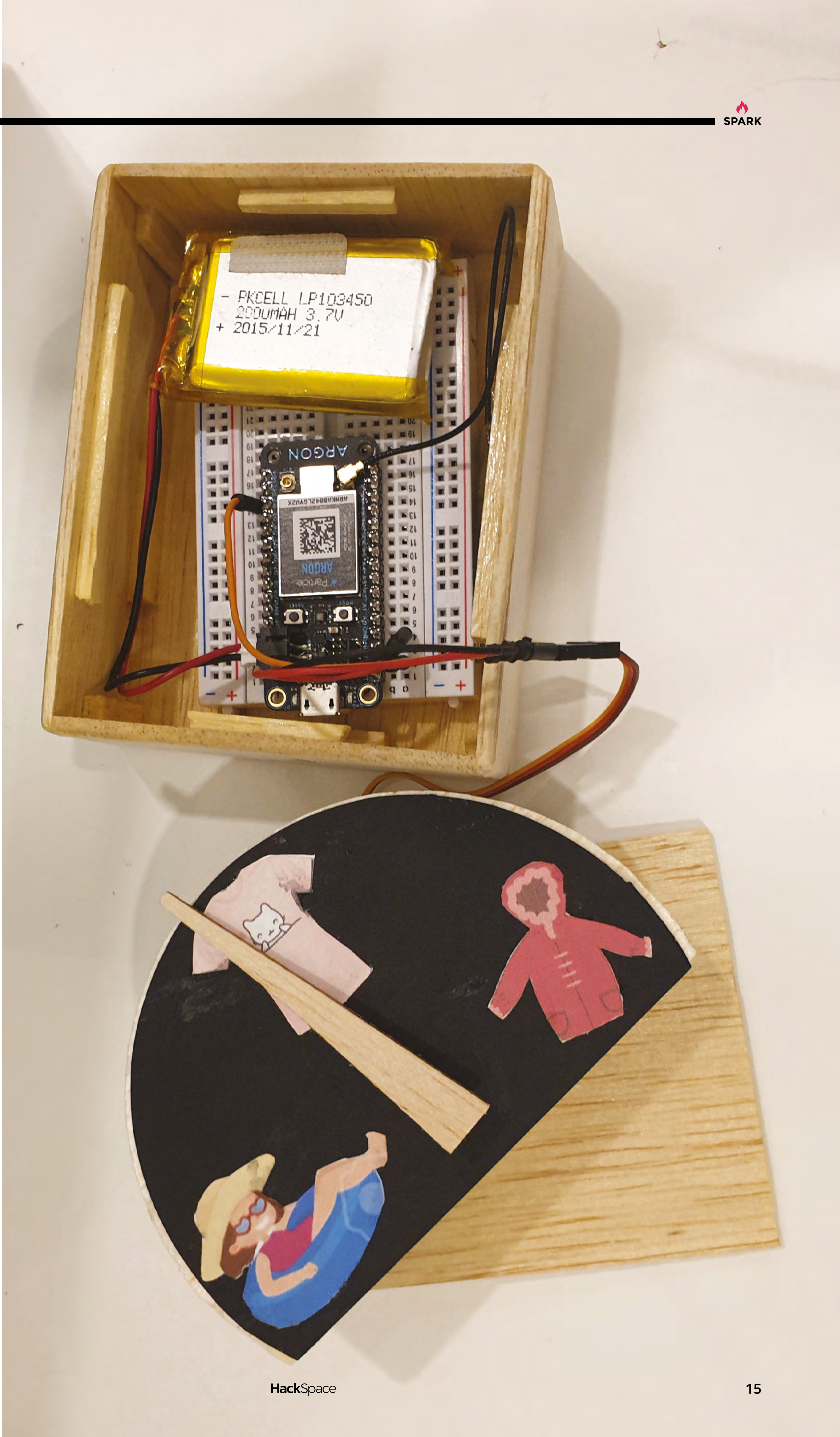
This is a fun project for younger children to help build. For a kid learning to translate the abstract temperature to actionable information, it can be rather challenging. Even grown-ups might struggle to digest weather data into actionable info. In this kid-friendly project, the abstract temperature/weather data is indicated as an outfit to wear.

I read the weather data from **openweathermap.com**, and update a servo controlled by a Particle Argon chip. The enclosure and dial were made out of balsa wood sheets, with printed illustrations selected by the kids. □


FIELD NOTES
64-Page Story Book
Durable Materials / Made in the U.S.A.
"Buy it fast. Read it slowly. It will change the way you think!"
We are all guilty of cognitive biases, simple errors we make in day-to-day thinking. But by knowing what they are and how to identify them, we can avoid them and make better

Right Always leave the house wearing weather-appropriate attire – thanks Badr!


DANIEL KAHNEMAN THINKING, FAST AND SLOW
ROLF DOBELLI
The Art of Thinking Clearly
INTERNATIONAL BESTSELLER


- PKCELL LP103450
2000MAH 3.7V
+ 2015/11/21
ARGON
Particle

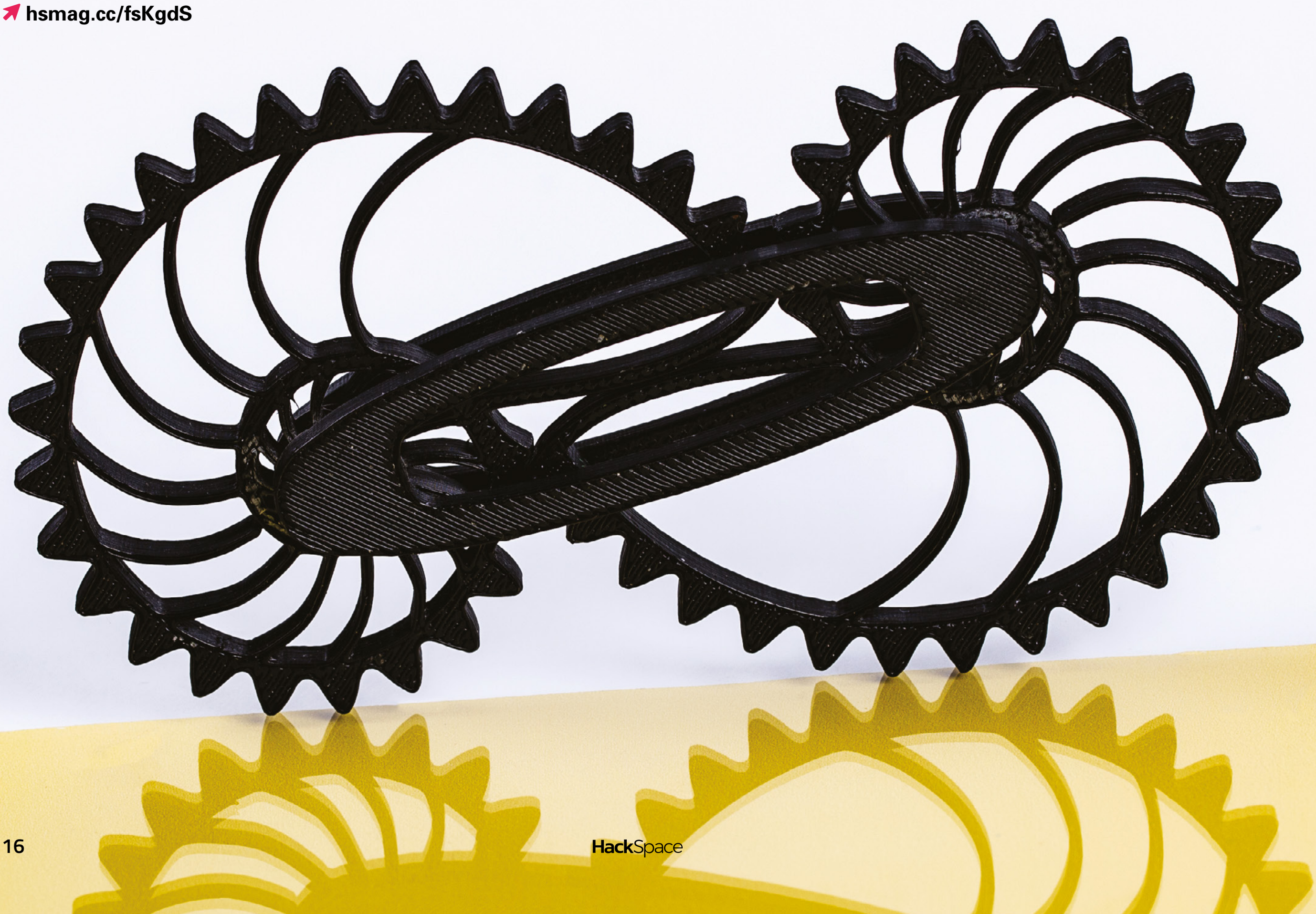
Objet 3d'art

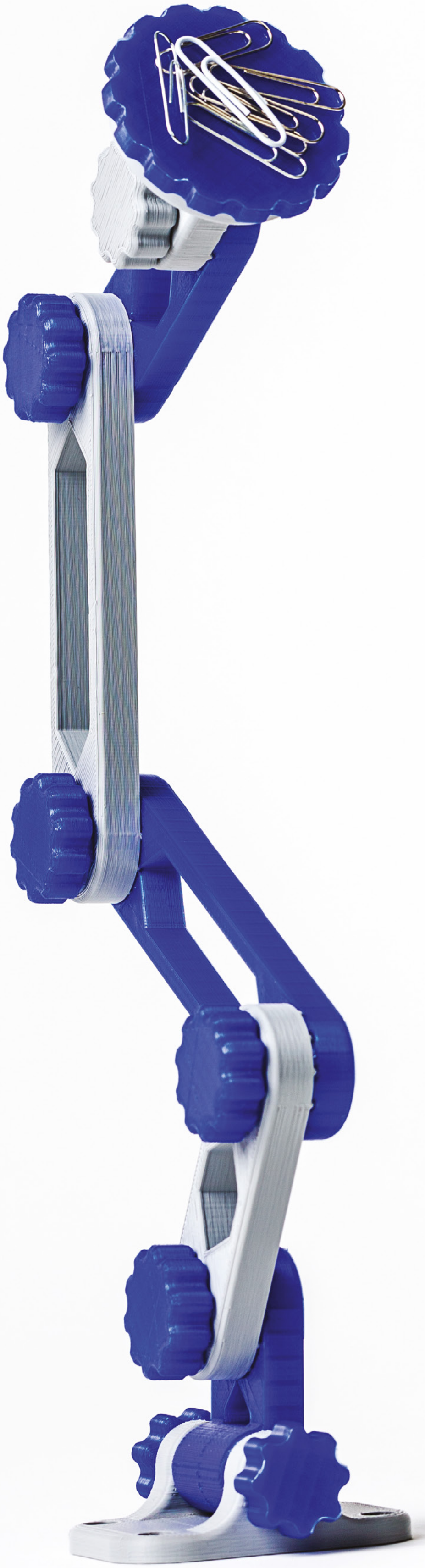
3D-printed artwork to bring more beauty into your life

This looks like it shouldn't work, but it does, and we've spent far too long playing with it. The two gears are non-uniformly shaped, but they're put together so that the widest point of one gear exactly meshes with the smallest point of the other one.

We're not sure how useful this is as an object, but the tolerances on this little toy, from Thingiverse user MishaT, are so small that it makes an excellent test to see if your printer is calibrated properly. □

hsmag.cc/fsKgdS





If you're always running out of hands and find clamps a little too fiddly, you might want to give this articulated magnetic hand a try. One thing to take note of is that it's not completely finished as soon as it comes off the print bed: as far as we know you can't print magnets out of plastic just yet. Instead you'll need four 1/2-inch diameter magnets (you can always modify the design files to make it more metric-friendly), and find some screws to attach it to the wall/workbench of your choice.

Thanks to Kenneth Haynie for putting this out into the world. □

hsmag.cc/VxMnpc

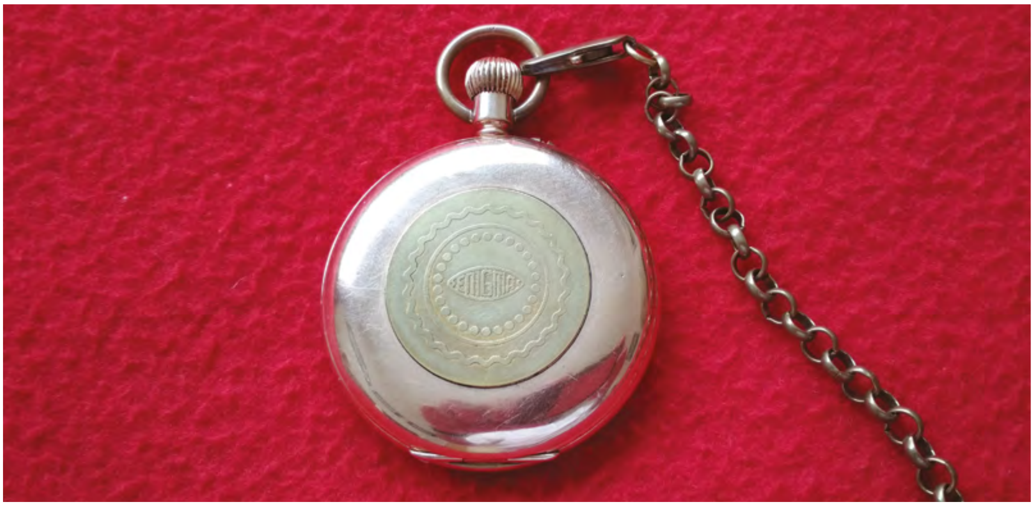
Pocket watch Enigma

The ultimate espionage accessory

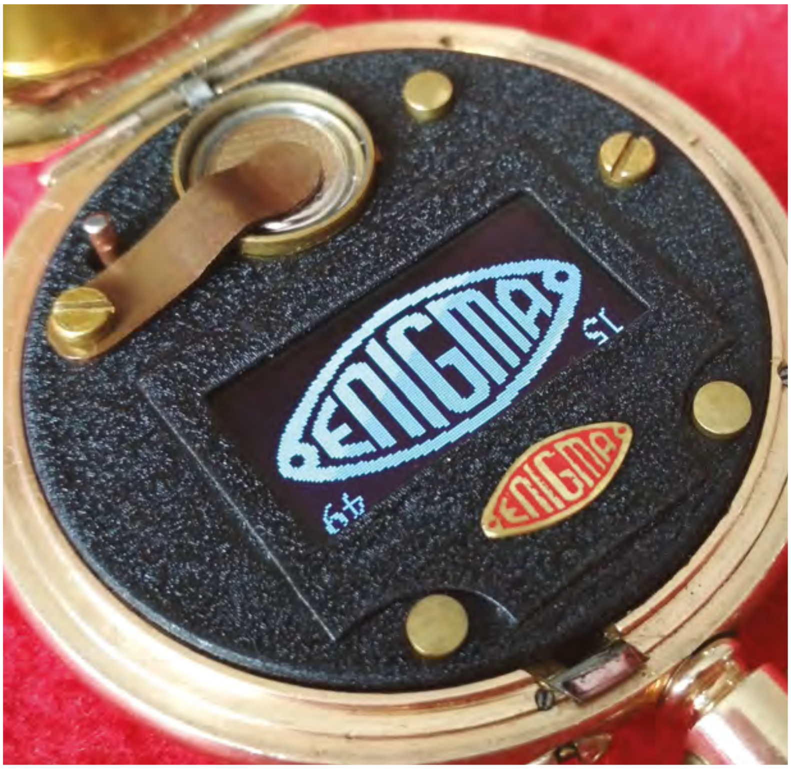
By Simon Jansen

asciimation.co.nz/bb/

I am a long-time maker of gadgets, machines, and devices. My interests are in electronics, computers and programming, as well as mechanics and engineering. A few years ago I ran away and joined the film industry, doing Animatronics at Weta Workshop. After doing enough to get my name in some film credits, I went back to my previous career in IT. I now work on projects of my choosing. I became interested in Enigma cipher machines a few years ago, when I wanted to learn how the Turing Bombe worked. To understand that you need to also understand Enigma. I made several electronic versions of the Enigma, but always wanted to make an Enigma that was more usable everyday. The idea came to put it into a pocket watch I could actually carry. And it really does tell the time too! □



Right ◈
The time is upside down because, when you wear the watch, that is the way it pulls out of your pocket!



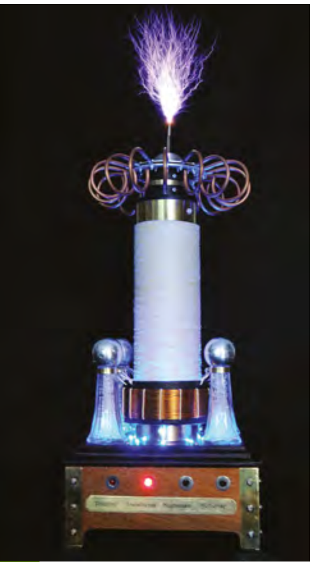
Meet The Maker: Derek Woodroffe

Danger!* High voltage!

(*actually not that dangerous at all)

Right
Derek's beam solenoid engine turns magnetic pulses into rotary motion

Below
This steampunk-looking Tesla coil is built on a cardboard tube



Not since David Bowie played the part of Nikola Tesla have we been so interested in high-voltage electricity than we were when we spoke to Derek Woodroffe. He's the person behind the Extreme Kits electronic toys, and someone who's working hard to make electronics interesting – largely by making it big and noisy.

"Extreme kits pretty much started around three years ago. Funnily enough with a solenoid engine, which was another one of my passions when I was a young kid, I usually made them out of Lego or Meccano, or whatever was to hand. I just had a look round and there was nothing that you could build similarly to that. So I thought I'd have a go. One day I was sitting in a pub, showing a friend of mine an acrylic solenoid engine, and a bloke walked in and said, 'Can I buy one of those?' From then on I knew that there was a market, and I'm still selling those.

"It's a niche market; I'm never going to make a million out of it, but it's something different, something you cannot get anywhere else.

"I do it [demonstrate high-voltage equipment] because they are very interesting, in so much as you can build a Tesla coil really easily, really cheaply. I think there are £5 Chinese kits out there that will make a nice little audio modulated Tesla coil, which is ideal as a hobby [indeed there are kits out there – we featured one in issue #4]. It's completely useless other than that it looks pretty and sounds good. And you can build that right the way up.

Building blocks

"From the making side of things, IKEA and B&Q are my high-voltage suppliers of choice, because that's really where I get all my components from. PVC tubes, various bowls, and all sorts from IKEA.

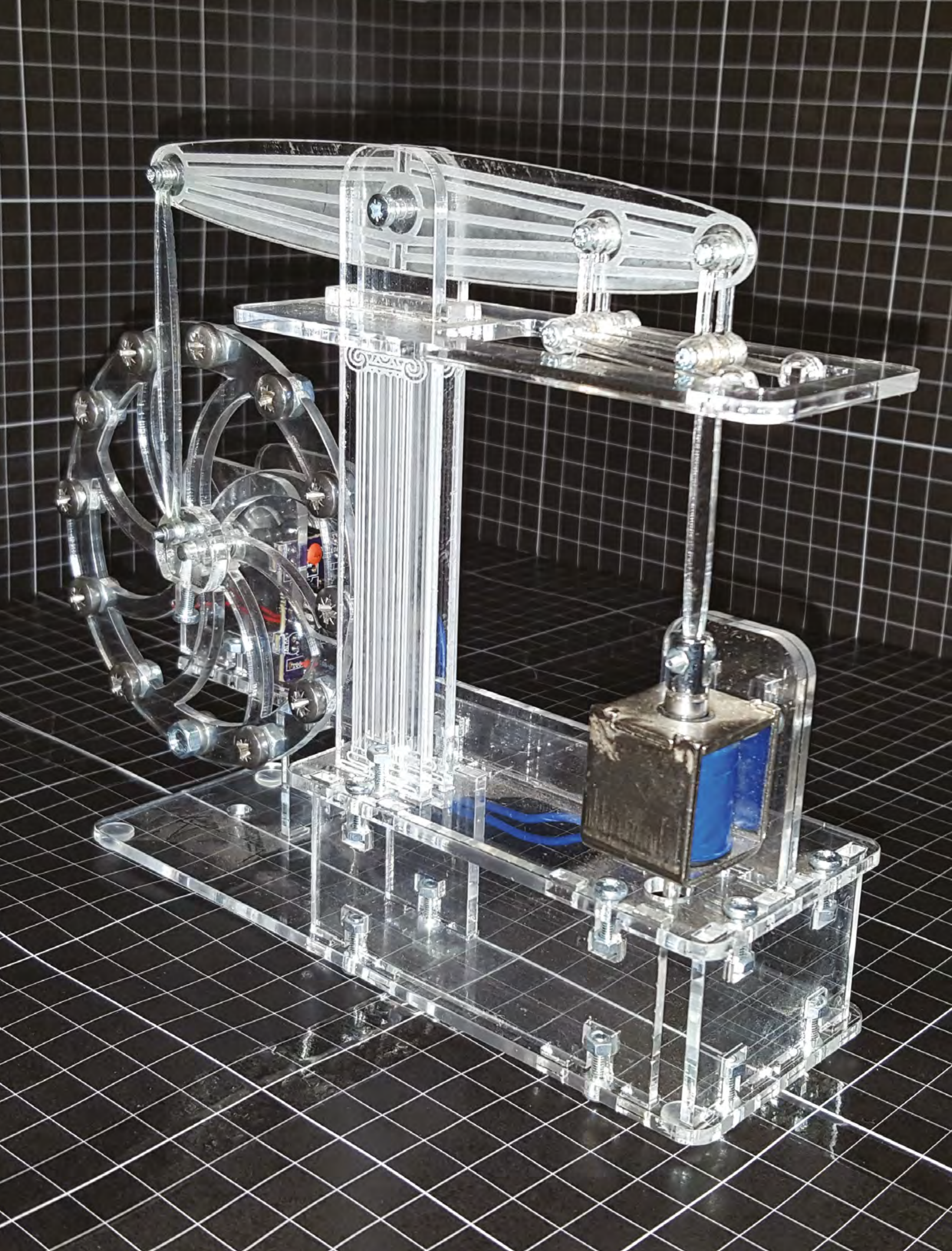
"At first glance, the electronics look really simple; the more you look at it, the more complex it becomes. It's just a fantastic hobby.

> "The electronics at first glance seem really simple; the more you look at it, the more complex it becomes"

"I don't sell Tesla coils myself – I have had one in development that I wanted to sell, and still do, but if I'm going to do it, it's going to work right, it's going to be better than the existing ones on the market, and it must be safe enough for people to build without doing damage to themselves due to lack of knowledge. Until I get all of those things together, I'm not going to sell a Tesla coil of any size.

"I did a Van de Graaff kit; that came out of a conversation with a physicist a couple of years ago who said, 'It's a great shame that nobody plays with a Van de Graaff any more, and it's also a shame that nobody gets to build one'.

"I sat down and thought 'Why not?' Let's make one that's usable in schools, safe, can be built by kids, and something that they can actually learn something from, because watching one is better than not →



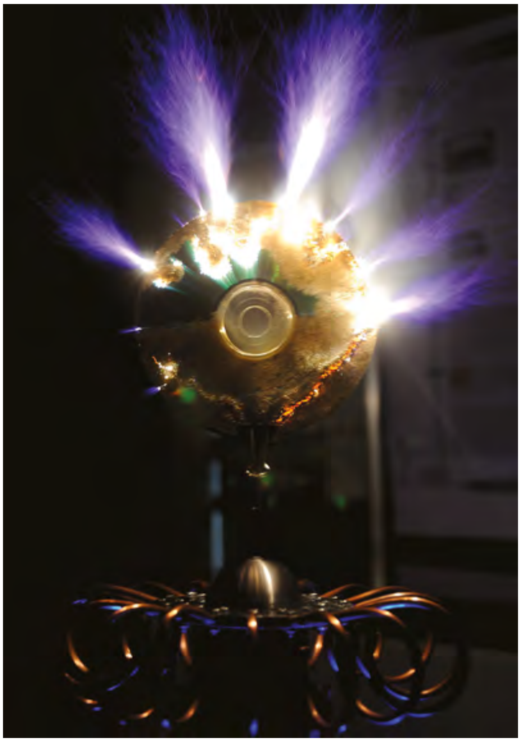
watching one, but building one of your own is way, way better in terms of learning.

"The schools, and even universities, are going this way where people don't actually get hands-on stuff. They come out after three years of watching videos of things and are entirely lost when they get out into the real world and somebody says to them, 'Here's a screwdriver; repair it'.

STAYING SAFE

"The best thing about the talks is doing it in front of kids. They're seeing less and less of the 'dangerous' stuff, both from a health and safety point of view and from the 'we can't afford to buy a Van de Graaff and show them proper sparks'. It was one of the things that really drove me when I was a kid, seeing some real dangerous electronics. Danger is interesting.

"The problem is, I know too much about them to explain why they're not dangerous, because to me it's obvious. I realise that coming from the other side of the coin, when you look at them you see high voltage, you see sparks going up into the air, and you can see all of the negatives. My usual view is that running large Tesla coils, and building large Tesla coils, is probably about as dangerous as riding a motorbike.



Above You can make differing patterns of burning air, depending on the power it's run at

Right This CD has been well and truly wiped

"Like a motorbike, if you take all the relevant precautions and are sensible with it, it's pretty safe. I was thinking the other day, 'I only know of one recorded incident where somebody has died from a Tesla coil.' It was a performer getting on for 70 years ago, who was sitting on top of a Tesla coil on a stage squirting sparks out of his fingers and hit an earthed rod. One of the arcs contacted the rod, he got full jolts through him, and it killed him. That's the only documented death, apart from one minor little thing – if you dig into it, it didn't kill him. He fell off the Tesla coil onto the floor and passed out because he banged his head. That's the only documented death involving a Tesla coil, only no one died.

"To be honest, you're more likely to get burned than you are anything else. It's the temperature of the arc more than anything else. High voltage and high resistance meeting each other, you get heat.

"The arcs from a Tesla coil are actually pretty safe. Because they are such a high voltage, the current is pretty low. As long as you don't get more than 5 milliamps across your heart – I usually take that down to 1 milliamp to be safe – you're not going to do any real damage. Saying that, it's not good for you, but it's not going to kill you either. [Editor's note: this rule-of-thumb isn't complete safety advice for handling high-voltages. Make sure you understand the risks before taking on any high-voltage project]

"About 20 years ago, when I started Tesla coiling, the other Tesla coilers all frowned and said, 'You can't do this, health and safety will never let you do it'.

"A while back I actually thought, 'That can't be true! There must be a way around this'. So I started looking at how to put together a proper risk assessment that would be accepted by people, and see whether I could actually get onto the BBC or whatever. And you can, if you go about it the right way. Occasionally you get the obnoxious idiots who go, 'Oh no, that looks dangerous. I'm not going to ask any more questions, I've just decided that it's too dangerous.'

"In the main they'll come back and say, 'You've given me this risk assessment. It's scary, but how do we get around it?' And you go, 'OK, this is a mitigation for X, Y, and Z, quite happy to have a chat, I'm quite happy for you to see the stuff.' I don't think I've ever been stopped doing anything yet due to health and safety concerns. We've always found a way around it.

"I am kind of fighting against the health and safety culture that bans anything interesting, because most of it's just idleness. 'Ooh, it's got high voltage in. We have a rule that says you may not touch anything more than 100,000 volts.' OK – so you've banned people walking across your carpet then, have you?

'Oh no, that's not the same.' You can easily get 100,000 volts walking across a carpet!

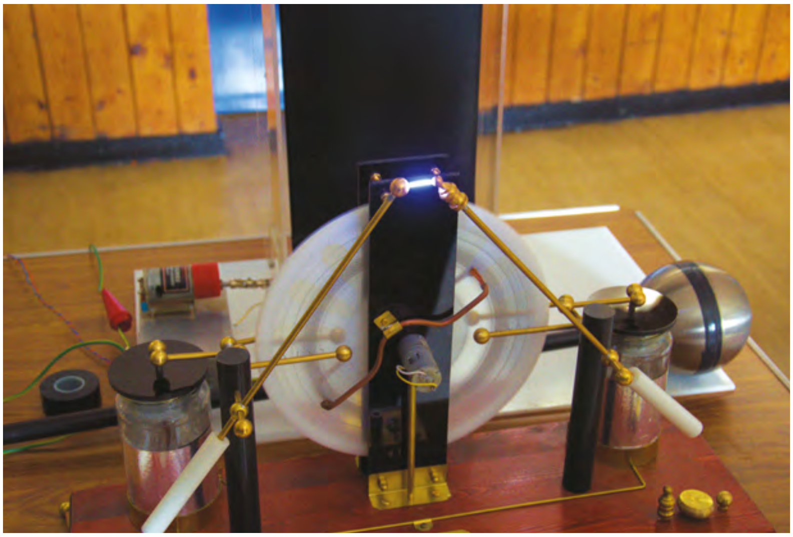
"The talk that I'm doing for the universities is based solidly around Tesla coiling. A little bit of the history, a little bit of 'this is a small Tesla coil, this is an electronic Tesla coil, this is a spark gap Tesla coil, this is something Tesla would have recognised, this is a modern-day one that plays music'. I do a bit of background with each of them, try to get them interested, and get a bit more knowledge out there. The talk in Nottingham was brilliant because a minibus full of 14- to 15-year-olds turned up. That's why I'm there.

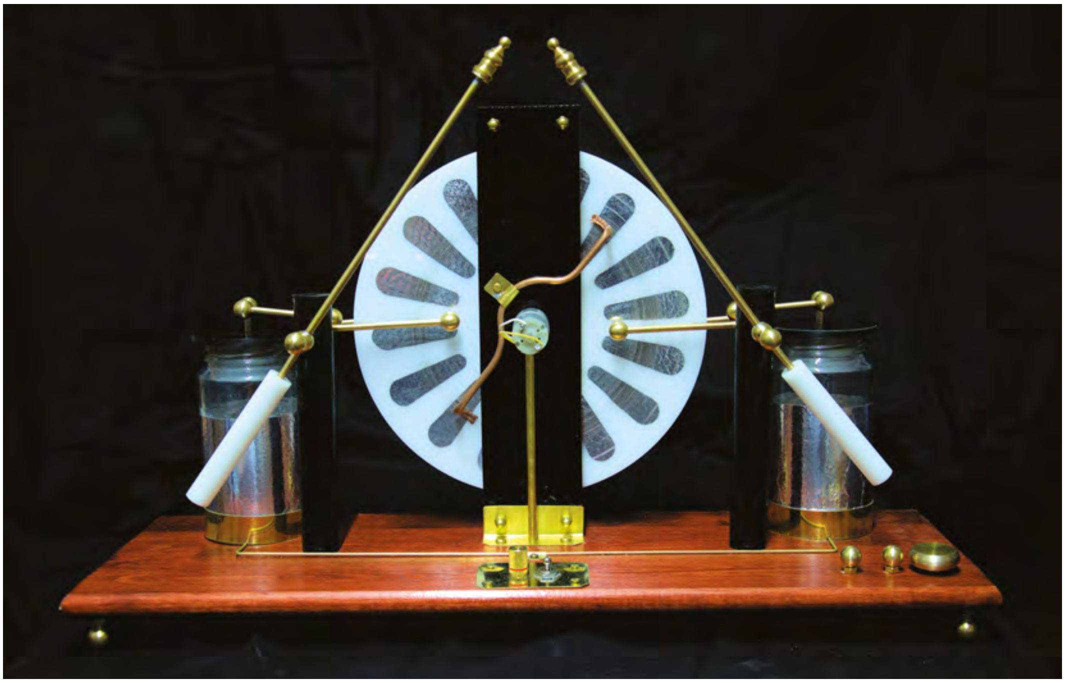
"People tend to look at the circuit and go, 'Oh that's really simple,' and not give it a second look. You get two coils, two capacitors, and a spark gap and you'll soon have some of the most complex relationships in electronics just with those components.

TESLA AND MARCONI

"Originally, they were for power transmission and radio communication. Marconi and Tesla sort of batted ideas between them – it turned out to be a lot more complex than that, and involved patents – but they had all sorts of interesting ideas for transmitting power and radio waves. Marconi beat Tesla on the radio waves by, interestingly enough, taking up one of Tesla's ideas and applying to what he was doing, and it just happened to fit really well. Tesla actually eschewed his new idea and went off in a different direction, which is bizarre, but that's the way history works. He backed the wrong horse.

"Tesla then went down the line of thinking, 'If I can't do radio, I'll transmit power,' but that was really at the point that his funding ran out. Whether he would have got something to work or not is a whole new question full of other intricacies. My own belief is that it wouldn't work, and it definitely wouldn't be a commercial success, because there was no way to charge for it. Which is also unfortunately where the free energy stuff that's attached to Tesla comes from, because when people said we'd get free energy from this, what they meant was, 'We couldn't charge for it,' which isn't the same as free energy." □





Above This beautiful object is a Wimshurst machine. As the disc spins, the collectors accumulate static electricity, which discharges across the spark gap

GAUSS FEST

Do you like the sound of high-voltage electronics, without the danger of killing yourself? Have a look for Nottingham Gaussfest, which is a meet-up of Tesla coilers and interested high-voltage people in Nottingham every year, currently at the Papplewick Pumping Station. (Planned for 13 April.) Come along and ask loads of questions!

The art of illusion

Beware of your senses – they lie to you

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**

Without measuring it, what do you think the diameter of a British one-pence piece is?

I was looking to get some pin-badges made and wanted them to be 'about the size of a penny'. I was surprised when I measured the coin – I was way out in my guess. I wondered if it was just me, so I asked my followers on Twitter, using a poll: **hsmag.cc/mgevky**.

What's the diameter (approx.) of a British penny?

- 10 mm (1 cm)
- 15 mm (1.5 cm)
- 20 mm (2 cm)

Over 450 people responded. Less than a third got the answer right. The actual diameter of the coin is 20.3 mm (**hsmag.cc/tIVpYF**). Some of those who responded can tell at a glance the size of a nut or bolt – and were also surprised that they got the answer wrong. One theory is that if it had been a stack of pennies, or a rod, or bar of the same diameter, then they probably would have got it. Maybe it is because it is so thin (1.52 mm for the older bronze versions, 1.65 mm for the newer copper-plated, steel version) that our senses were deceived.

Optical illusions have always fascinated me – from the 'which is the longer line' with arrowheads pointing in or out to circles that spin, if you stare too long at them. I also enjoy the impossible construction illustrations by M.C. Escher (**mcescher.com**), and the duck/rabbit-style illusion (**hsmag.cc/zDzWMM**), where if you think of a duck, you see the duck, think of a rabbit and you see the rabbit – but you can't see both at once.

I've never really considered physical illusions until I was thrown by the penny size. I remember at school, putting one hand in cold water and the other in hot for a while, and then putting both hands in lukewarm water. The lukewarm water felt hot to the hand previously in the cold water, and cold to the hand previously in hot water. This was in a chemistry lesson – although I can't now relate it to any part of the syllabus.

I've never really considered physical illusions until I was thrown by the penny size

I've only just discovered the Aristotle illusion (**hsmag.cc/lHsDma**). Cross your fingers, close your eyes, and touch a small ball, such as a marble or ball bearing, with the inside part of the crossed fingers. You should feel like you are touching two objects and not one (the closing of the eyes part is to stop your logical brain overriding the sensation).

The senses are amazing things – but it's fun when you can fool them. □

Easier hardware testing in production

Mistakes happen – you need to spot them quickly

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.

I often say the 'product behind the product' is the tester used for quality-checking a hardware product. Reputable factories will test 100 percent of every product shipped, and the more complex a product, the more complex the tester. Without an automated tester, confirming every solder joint quickly becomes an error-prone exercise in tedium, even for a run of a couple of dozen boards for an art or hobby project meant to be shared among friends.

Surprisingly, there is no convention for dealing with even the most common issues facing the production of simple IoT-style devices, such as loading firmware, testing radios, and checking all the GPIOs are wired up correctly. Time and time again, I had found myself frustrated because I was spending more time writing test code than I was building the product itself. Sean 'xobs' Cross, was similarly frustrated when developing the test programs for projects like Novena (**hsmag.cc/vMblMf**) and Chibitronics (**chibitronics.com**), so he created a testing infrastructure called Exclave (**hsmag.cc/xBVSgP**). The idea behind Exclave is to create a standardised tool for stringing together sets of small POSIX-compliant scripts that can be quickly and safely rearranged to accommodate various production testing scenarios.

By relying on small scripts that can be written in any language, Exclave makes it easier to reuse the single-purpose scripts inevitably written during board bring-up for debugging and validation of point features. By creating a system that defines dependencies between the scripts and wrapping them into scenarios, Exclave makes sure that important tests don't get accidentally skipped or commented out as the overall test flow is optimised for production throughput.

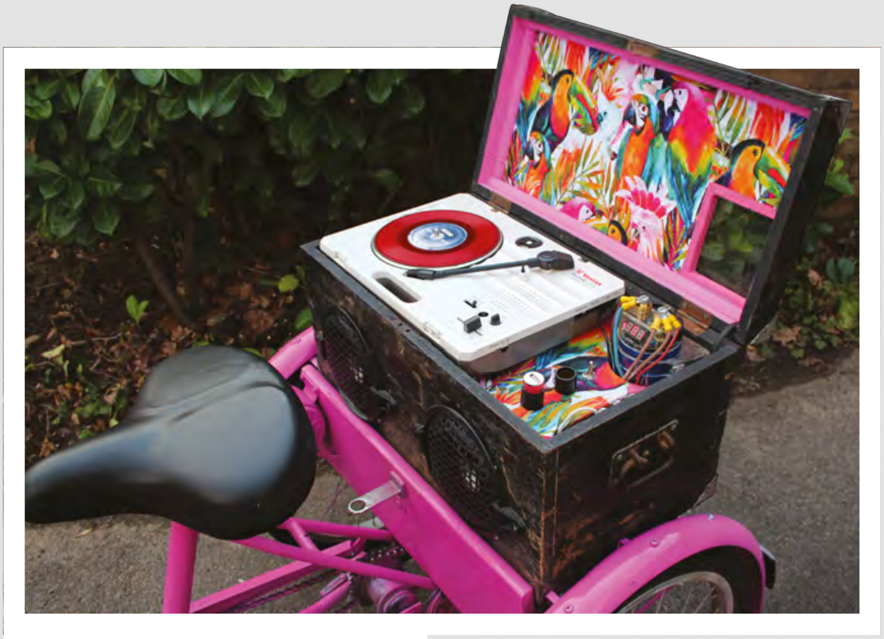
I've used Exclave to develop testers for about a half dozen products and, despite its rough edges, Exclave has shaved months off product development cycles. Exclave enables me to quickly build production testers for even fun art projects meant to be shared only among a few friends. While Exclave can run on any POSIX-compliant OS, I usually just install it on a Raspberry Pi and take advantage of the native GPIOs to handle most of the tasks, from uploading firmware over OpenOCD, to testing GPIOs, to driving simple ADC/DACs on a Pi HAT to verify analogue signals. If you're considering building more than a few of anything, I encourage you to take a look at this freely available, open-source project. For a more detailed look at designing and running a test jig, take a look at my blog on the subject: **hsmag.cc/rXxUtU**. □

> **Exclave enables me to quickly build production testers for even fun art projects**

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**



RIDE ON

I loved Caz Ryves' story of making things out of old bikes. It feels like it's an area ripe for hacking and I'd love to try something similar, but I've no idea where to start. How do you go about modifying a bike? Can you get started without knowing how to weld?

Paul
Cardiff

Ben says: As a regular cyclist, it's a subject close to my heart. We don't have anything planned in at the moment, but we'll get out our Reynolds 531 and see what we can come up with. If readers have any suggestions, get in touch.

MAVERICK

Colin Furze is an absolute inspiration. Sometimes his attitude to his own safety is a bit … er … lackadaisical, but his enthusiasm for making things is infectious. I've loved watching his YouTube videos, and it was great to hear a bit more about what goes on behind the scenes. Any chance of a HackSpace magazine / Furze co-build in the future?

Jordan
Inverness

Ben says: I'm hugely jealous of our photographer, Fiacre Muller, who got to ride on the bike of springs. We've got nothing planned in at the moment, but who knows what the future holds?

SUFFICIENTLY ADVANCED

I really enjoy your regular *Improviser's Toolbox* articles. They show that you don't need to have the latest bit of technology to have fun making things. In the candles article [issue 15], you have a candle-powered smartphone charger, but I couldn't see how it worked. Can you shed any light on it?

James
London

Ben says: Peltier elements are one of those technologies that seem magical at first. If you create a heat differential across it (that is, if you make one side hot and the other cold), it generates electricity, which is fairly impressive – and this is how the candles created power for the charger. The bit that really blows my tiny mind, though, is that it also works the other way – if you put a voltage across it, one side gets hot and the other cold.

REGULAR

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.

Cthulhu Shield

Give yourself extra senses

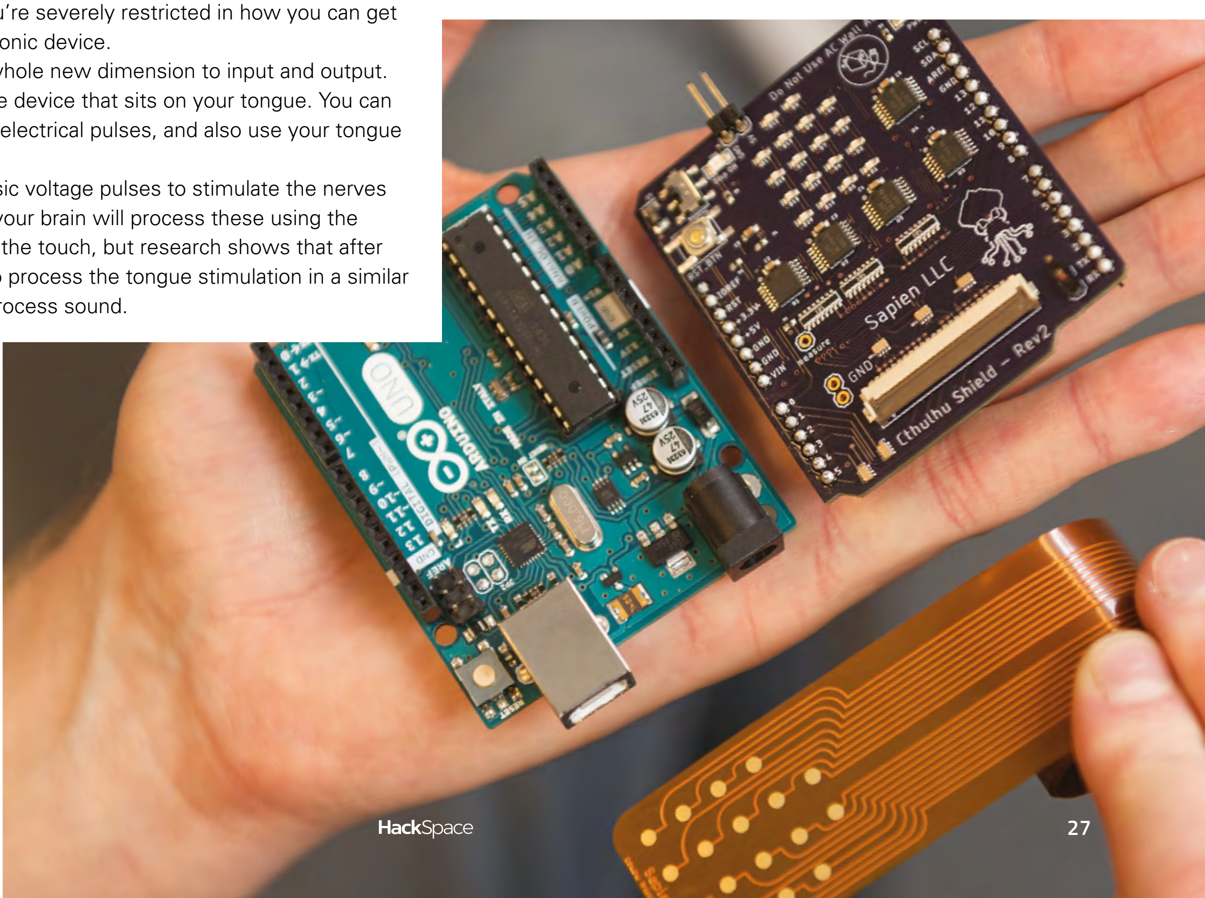
From $75 | hsmag.cc/lkTSQL | **Delivery:** July 2019

How do you interact with electronics? A screen? LEDs? Buttons? Sound? These are all great options, but they rely on just a few senses. If you have limited eyesight or hearing, then straight away you've got far fewer ways of receiving information from computers. And if you've got limited movement in your hands (or if they're just occupied elsewhere), you're severely restricted in how you can get information into an electronic device.

The Cthulhu brings a whole new dimension to input and output. It's an Arduino-compatible device that sits on your tongue. You can get information via small electrical pulses, and also use your tongue movement as an input.

This shield uses biphasic voltage pulses to stimulate the nerves on your tongue. Initially, your brain will process these using the part of it associated with the touch, but research shows that after regular use, you'll start to process the tongue stimulation in a similar fashion to the way you process sound.

Once you get past the slightly odd idea of electronics that you put in your mouth, you start to see the possibilities of this. Sensory augmentation is the practice of adding extra senses to humans. For example, you could use the Cthulhu Shield to give yourself the ability to sense ultrasonic sounds, magnetism, or anything else you can read with an electronic sensor.



Right The grid of electrodes gives you 18 different points on your tongue that can be stimulated

Space of the month: MakerLabs, Vancouver

MakerLabs

Alyssa Wongkee

makerlabs.com

WHAT WE ARE

MakerLabs is a makerspace in Vancouver, Canada that provides people with the tools, space, and skills to make almost anything. After five years and several expansions, we are now 42,000 square feet between two buildings, which is filled with workshops and studio space, including a wood shop, metal shop, CNC machines, laser cutters, electronics, and more. We have about 200 members and 100 studios, made up by a diverse mix of makers; including, artists, builders, entrepreneurs, and designers. Our members range from hobbyists to small business owners, start-ups, and non-profit societies.

WHY WE DO IT

Our mission is to provide access to tools, empowering members of the community to create and make. We do that in four main ways: through education, membership, studios, and fabrication services.

Our education program includes regularly scheduled classes, providing the foundations for almost anyone to use our tools and space. We also hold specialised workshop and events, such as team building events, and school workshops at the K–12, to university level. We are launching an intensive two-week digital fabrication program.

Wood and metal shops, as well as other trades have been historically male-dominated and our membership reflects that. We have a subsidised residency program to address this issue: Tools for Women, which offers classes and membership for two months. Throughout the last eight months, 32 new female-identifying members have graduated our program, 20% of which continue their membership today.

We also offer a stewardship program where people can volunteer in exchange for membership and/or classes. Stewards can either be a receptionist or shopkeeper. Not only does this program make tool access financially accessible, but it quickly immerses stewards in the MakerLabs culture. →

Above
Through its Tools for Women program, MakerLabs is trying to redress the balance in this often male-dominated field

One of our strengths is that, as a private business, we have been able to quickly address the needs of our members and focus on improvements to our spaces and tools. This also allows us to have a fabrication department, which is a team of full time staff that take on almost any fabrication project, however big or small, for our members and the wider community. By being housed within the makerspace, it allows us to keep our membership and fabrication prices low and competitive. Fabrication projects range from full service laser cutting and CNC projects, to design, furniture, signage, and everything our makerspace offers.

Outside of those main streams, we also try and remain active leaders in our community by partaking in various initiatives. We host many events, from maker meet-ups to community BBQs, and rent or donate our space for private events. We showcase and provide fun exhibits at local events, such as a sustainability biodome or vortex fountain. □

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.

Above
MakerLabs is the hub of makers, designers, and collaborators in Vancouver, Canada

Build a Makerspace for Young People

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

Sign up today: rpf.io/makerspace

Raspberry Pi

The Raspberry Pi Foundation. UK registered charity 1129409

#RPiLearn

HackSpace
TECHNOLOGY IN YOUR HANDS

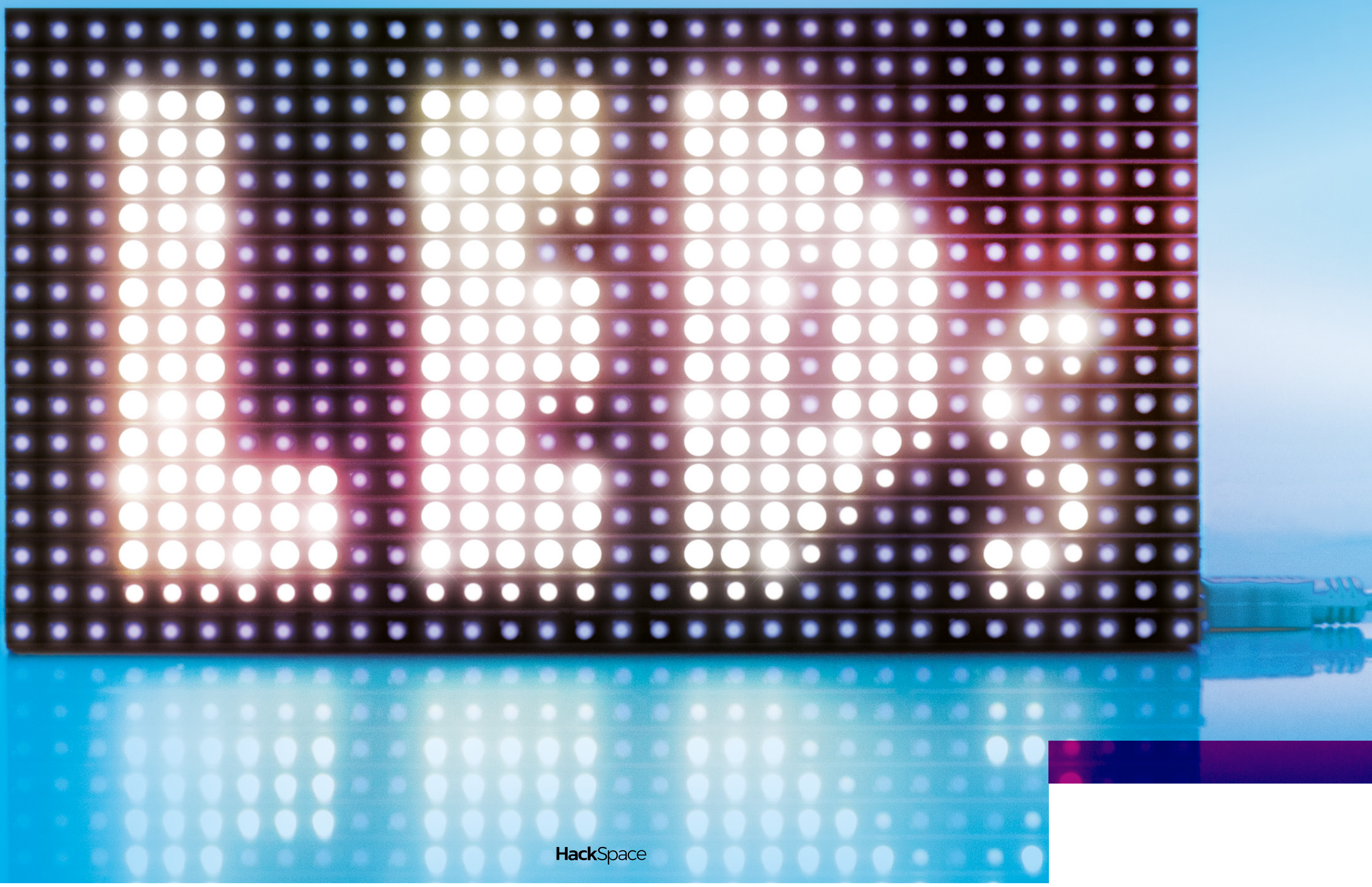
LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

MAKE ART WITH

Create something beautiful with silicon, electricity, and your endless imagination



The word ‘art’ means many things to many people. It can mean ‘pretty things to look at’, it can mean ‘things that make you think’, and it can mean ‘seemingly meaningless things that people wearing berets pontificate about’. Each of these, though, has a common theme – they enhance our life. We should smile at beautiful objects, and being shown a new way of looking at a subject can help you understand it more fully. And there’s nothing wrong with a bit of pontificating, if that’s what you enjoy.

A bit of electronics can help make art, whichever definition you choose to use. By adding programmable electronics, we can make art that’s reactive – it can change in response to users’ input, environmental conditions, online data sources, or just over time. This interactivity gives us the opportunities that have been impossible for artists throughout most of history. If the purpose of art is to look pretty, then changing light and shadows gives us the ability to make art mesmerising. If the purpose of art is to make people think, then the ability to see how different inputs change the state gives us tools for understanding the implications of actions far more deeply than static pictures. Changing patterns also gives us a whole new set of ways to make people in berets pontificate. Oh, and it doesn’t hurt that LEDs look awesome.

You might choose to incorporate the electronics into the aesthetic of the piece and show off the functional heart of the creation, or you may obfuscate it away and use it for control behind the scenes – you’ve got artistic licence to put it where you like. As long as it’s there, you can take advantage of it to enhance your work.

Art should be for everyone and, as makers, we think that the best way to start appreciating art is to have a go at making it. Turn the page to get started. →

ART

LEDs

The makers of light

Right ⇨
As a NeoPixel is moved through a long-exposure photograph, you can see it blink on and off. DotStars – which have a faster PWM rate – avoid this

Below ⇘
A strip of NeoPixels – each LED is the same component, but they can be programmed to display different colours

At their simplest, LEDs are components that convert electrical power into light. Connect them the right way around to a power source (with some form of current limiter) and they'll glow. Each LED will shine a single colour.

Fortunately, manufacturers can pack three LEDs (red, green, and blue) into a single component. By varying the power to each different LED-within-an-LED, we can produce a wide range of hues. However, these can be a little tricky to control as each LED requires three inputs, so a simple 10×10 matrix would require 300 inputs. A range of tricks and techniques can be used to greatly reduce this, but it does increase the complexity. But there's a further trick electronics manufacturers have that makes it even easier for use.

Addressable LEDs have microcontrollers built into them. These aren't powerful, programmable microcontrollers, but they're just powerful enough to handle a simple communications protocol. There are quite a few different types, but two are most popular with makers: WS2812s (often called NeoPixels) and APA102s (often called DotStars). They're both widely available from both maker stores and direct-from-China websites. NeoPixels use a single data line, while DotStars use a signal and clock line. Both, however, are chainable. This means that you connect one (for NeoPixels) or two (for DotStars) pins to your microcontrollers to the Data In connectors on the first LED, then the output of this LED to the input of the next and so on.

Exactly how many you can chain together depends on a few different things, including the power of the controller and the intended refresh rate. Often, though, the limiting factor for most hobbyists is the amount of electricity you need.

SAFETY

This is just a whistle-stop guide to the basics of powering LEDs – it's not a comprehensive guide to all things power-related. Once you go above a few amps, you need to think about what you're doing with power. Once you start to approach double figures, you need to make sure you know what you're doing and, if you find yourself shopping for an industrial power supply, then you really need to make sure you know how to use it safely.

WHICH TYPE TO USE

While many types of addressable LEDs are available, NeoPixels and DotStars are by far the most popular with makers. The big difference comes down to the speed of them. LEDs are made dimmer by turning them off and on very quickly. The proportion of the time they're off, the dimmer they are. This is known as pulse-width modulation (PWM). The speed at which this blinking on and off can have implications for some makes, such as when the LEDs are moving quickly.

NeoPixels	DotStars
• Cheap	• More expensive
• Slowish refresh rate	• Faster refresh rate
• Slowish PWM rate	• Fast PWM rate

POWER

Keeping the lights on

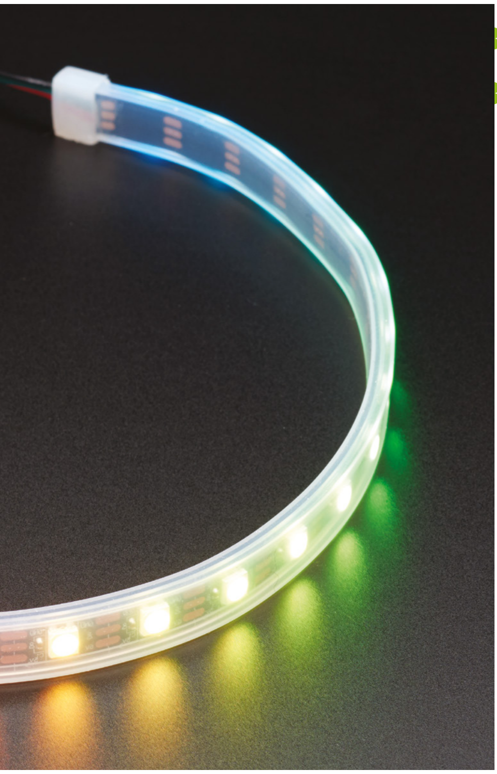
Most chainable LEDs (including NeoPixels and DotStars) have a 5 V power line. This voltage is easy and safe to work with, but you need to be careful about the amount of current you draw.

Each LED uses very little power – typically a maximum of 20 mA per colour. RGB LEDs use a maximum of about 60 mA. However, large displays can contain a lot of LEDs. A modest 32×32 grid contains 1024 RGB LEDs, so that's a maximum of 60 amps. This is the sort of current used to weld!

Of course, that level of current is only used if all the LEDs are turned on, bright white, at once. For most patterns, you're unlikely to draw more than about 20 mA per pixel, but this obviously varies. You'll need to experiment with your pattern and an ammeter (or multimeter) to see how much you actually draw. Start small, or at low brightness, and slowly ramp up to see what you're actually drawing.

The limit in your system isn't just the amount of current your power supply can source. Pulling too much current through thin wires and PCB traces will cause them to get hot and potentially melt or set on fire. You should find the 'ampacity' of your wire from your supplier. A significant issue is whether the wire is exposed to the air where it will be cooled naturally, or if it's enclosed in some material which will trap the heat.

Most LEDs come preconnected in strips, matrices, or other configurations. You need to make sure that the wiring on these can take the current. On matrices with more than 64 pixels, you'll usually find more than one power supply point. This means that you can supply power at more than one point and the load will be shared between them. Just because someone will sell you a 5 m strip of LEDs with 144 LEDs per metre, doesn't mean that it's safe to push 43 amps down it. These strips can have power supplied at intervals along them, or be cut and powered completely independently. →

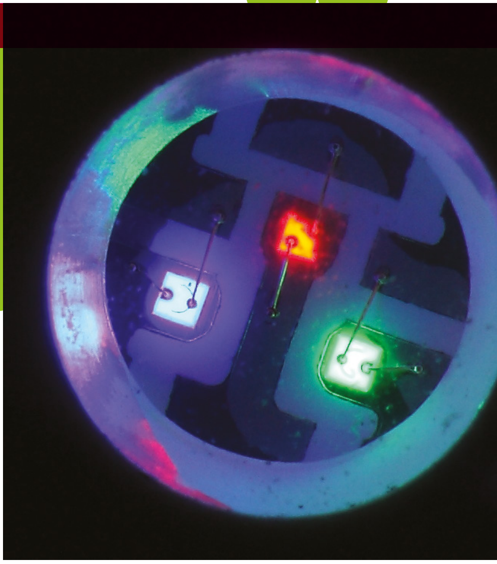


REDUCING YOUR POWER REQUIREMENTS

The easiest way to safely power your LEDs is to make sure that you keep your power requirements low. Here are our top three tips for keeping the power draw manageable. Not only will it simplify power management, but it'll make your project a little more eco-friendly.

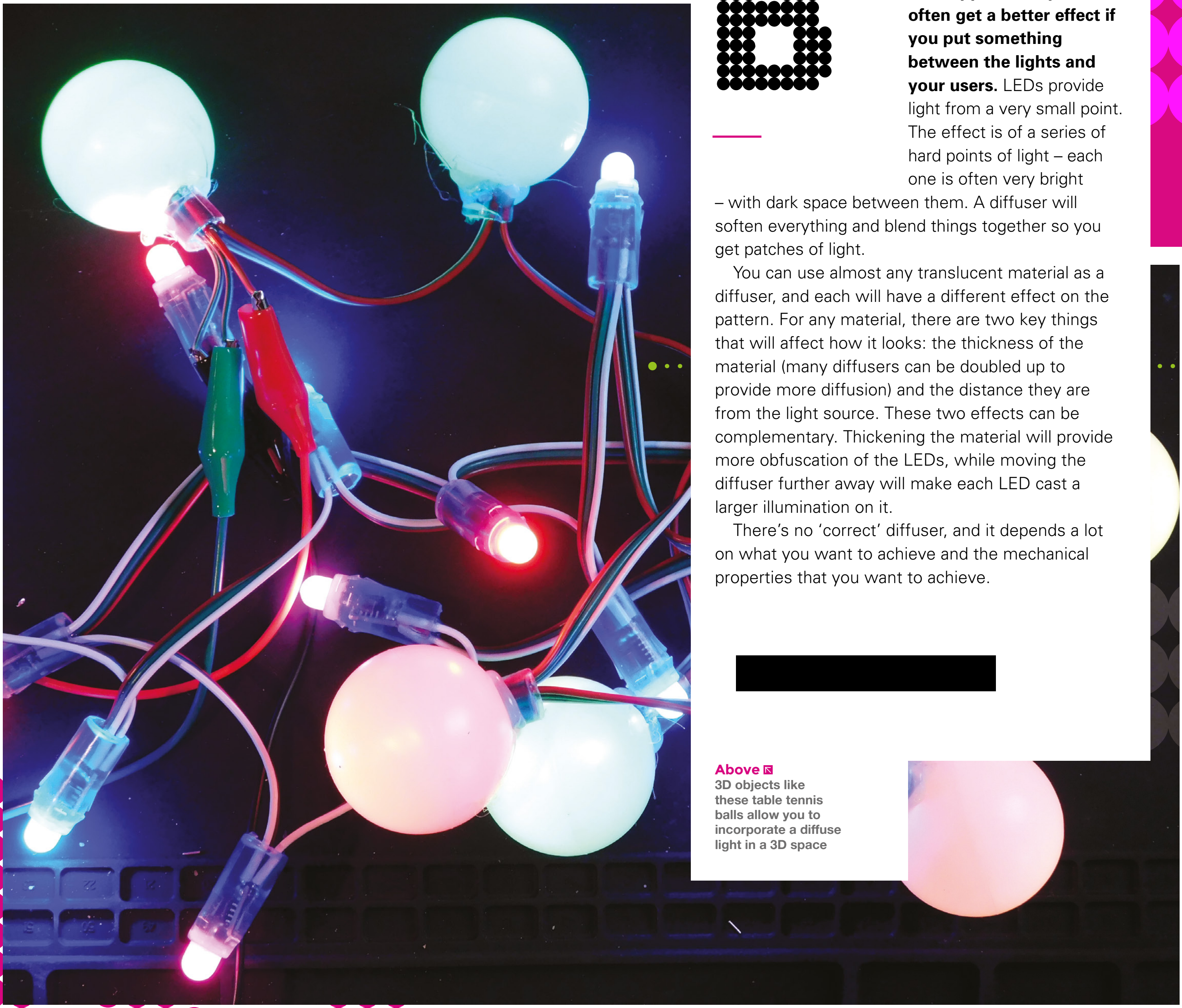
- Use fewer LEDs. Picking the right diffuser (see next page) can have more impact than pushing more LEDs in.
- Turn some LEDs off. Twinkling lights can have a bigger impact than always-on.
- Dim the lights. Our usual process is to start at around 10% brightness and turn it up if we need it.

Below A close-up view of an RGB LED, showing the three separate LEDs that make up the component



DIFFUSION

Give your lights more texture



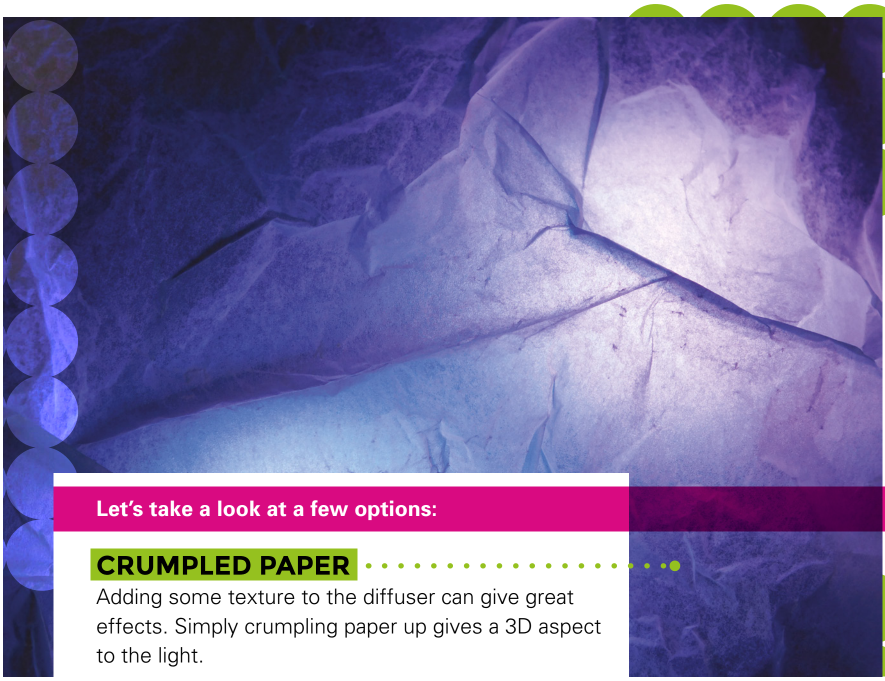
Bare LEDs have a certain nerd-appeal, but you'll often get a better effect if you put something between the lights and your users. LEDs provide light from a very small point. The effect is of a series of hard points of light – each one is often very bright – with dark space between them. A diffuser will soften everything and blend things together so you get patches of light.

You can use almost any translucent material as a diffuser, and each will have a different effect on the pattern. For any material, there are two key things that will affect how it looks: the thickness of the material (many diffusers can be doubled up to provide more diffusion) and the distance they are from the light source. These two effects can be complementary. Thickening the material will provide more obfuscation of the LEDs, while moving the diffuser further away will make each LED cast a larger illumination on it.

There's no 'correct' diffuser, and it depends a lot on what you want to achieve and the mechanical properties that you want to achieve.

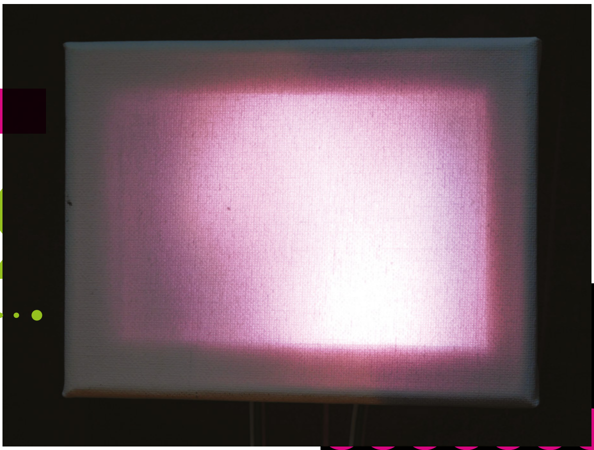
[redacted]

Above
3D objects like these table tennis balls allow you to incorporate a diffuse light in a 3D space



Left
Diffusers can impart a texture, as well as smoothing out the light. Here, crumpled paper adds a depth missing in some diffusers

Below
This artist's canvas has a very even diffusion and enhances the art-cred of the piece when hung on an easel



Let's take a look at a few options:

CRUMPLED PAPER

Adding some texture to the diffuser can give great effects. Simply crumpling paper up gives a 3D aspect to the light.

CANVAS

Artist's canvases are ready-made diffusers and add a certain flair to projects. The wooden frame is great for mounting to. Here, we've hot-glued an LED matrix behind a miniature canvas.

TABLE TENNIS BALLS

You don't always want to diffuse a lot of LEDs. Sometimes, just one at a time is all you need. Thankfully, there's a cheap, widely available diffuser that's great for just this. Cut a hole in the bottom of a table tennis ball (that's ping-pong ball to our American readers) and hot-glue an LED inside, and you've got a spherical diffuser. It's shown here with 12 mm WS2811 LEDs which are already diffuse. This double-diffusion provides a really even light. Other thin plastics can have a similar effect.

DIFFUSION FABRIC

Photographers need diffusion to provide a soft light source, and that's what diffusion fabric is for. It can be a little thin, so you might need to double or quadruple up, but it works well.

PACKAGING FOAM

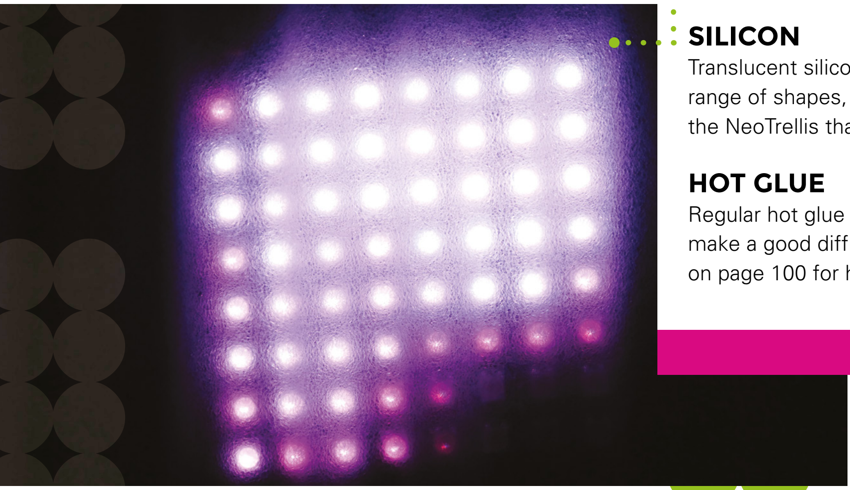
There are loads of translucent materials that can be salvaged from bins and saved from landfill. This packaging foam was used to ship some vacuum tubes.

SILICON

Translucent silicon can be mass-produced in a range of shapes, such as the square keypad on the NeoTrellis that we looked at in the last issue.

HOT GLUE

Regular hot glue is translucent when cold, so can make a good diffuser. Take a look at our tutorial on page 100 for how to do this. →



Left
Get creative with your diffusers! This packaging foam is thin, but the cellular structure picked out by the LEDs has a fog-like effect

CONTROLLING YOUR LEDs WITH FADECANDY

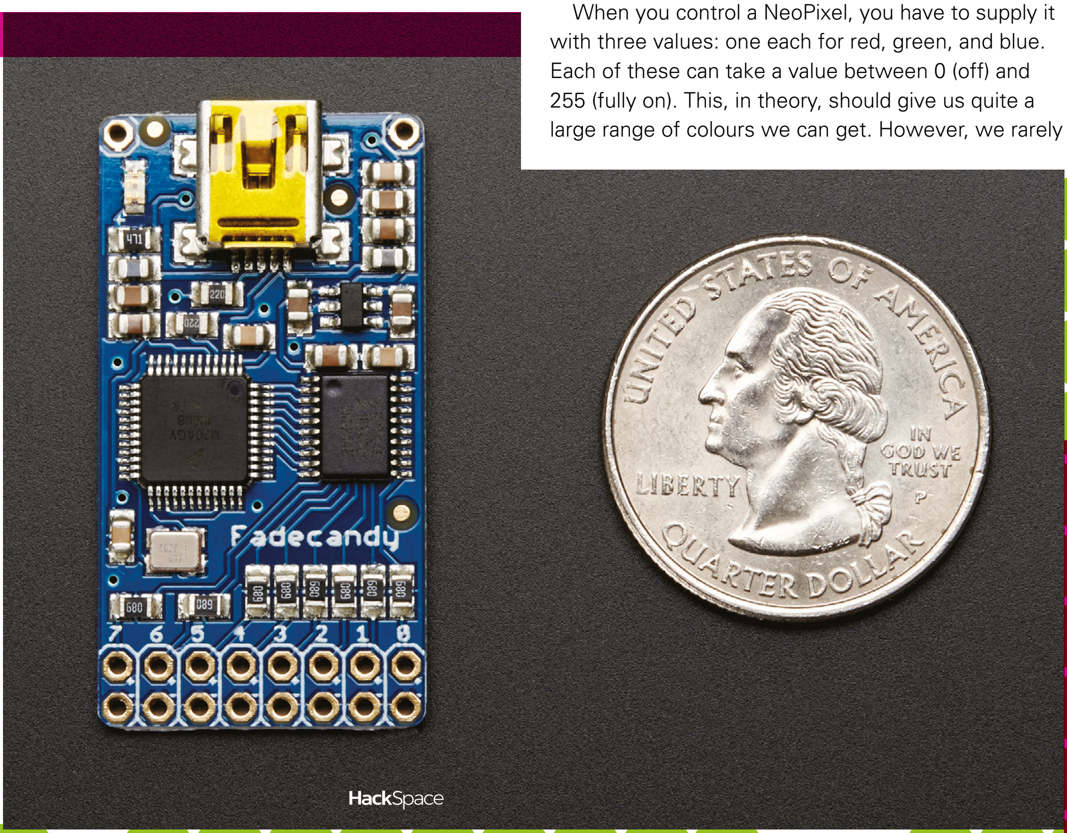
Give your LEDs extra depth range

So, you've now got your LEDs, whether that's as individual components, or amassed on a strip or matrix, and you need to control them. There are innumerable ways of doing this with modern hardware, and each method lends itself to different graphical styles. One of the most powerful is the FadeCandy board, developed in partnership by Micah Scott (aka ScanLime) and Adafruit.

There are a couple of things that we think make this board special. The first is that it manages to achieve far more dynamic resolution than other boards by using dithering, and the second is that it uses linear interpolation to achieve a smooth transition between frames. These sound technical, but everything is handled behind the scenes and you get great-looking LED projects easily.

When you control a NeoPixel, you have to supply it with three values: one each for red, green, and blue. Each of these can take a value between 0 (off) and 255 (fully on). This, in theory, should give us quite a large range of colours we can get. However, we rarely

Below
The FadeCandy requires just a USB connection and two soldered lines to control each chain of LEDs



Left There's far more definition in the low levels of light when using the FadeCandy than is possible with just using the PWM control

want fully on. In fact, a value of just 20 can often be as bright as we want to go. This means that we often find that we have far fewer colours to choose from than it first seems – particularly at the lower levels of brightness. The differences between 0, 1, and 2 are quite big jumps in colour and we get a stepping effect when trying to display a fade in this colour range.

The FadeCandy controller handles this problem by holding the frame rate very high (around 400 fps). At this rate, it can achieve extra levels of brightness by switching between different colours very quickly. So, for example, it can achieve a tone half-way between the usual 1 and 2 by switching between these two values very quickly (FadeCandy aims for 400 times a second). Our eyes can't see flickering this fast, so it will appear to us as blending the two colour levels together. This is similar to the process used in printing to create a smooth blend of colours, and is known as dithering.

The FadeCandy also smooths your animation between frames. When you send a few scenes to be shown, it doesn't flick straight to it – instead, it fades into it, interpolating between the two as it goes. This results in much smoother animations. For this to work, it requires the hardware to have very fast control over the LEDs, and this means keeping the LED numbers low. The reason for this is that shorter chains of LEDs need less data pushed down them, and less data per frame means more frames can be pushed out in a given time. Holding 400 frames a second requires chains of 64 LEDs or fewer. Obviously, many LED displays require far more LEDs than this, and the FadeCandy allows this by combining many chains, each of which has 64 (or fewer) LEDs. Each FadeCandy board can control eight chains (so 512 LEDs in total), and multiple FadeCandy boards can be used together to create even larger effects. The theoretical limit is around 20,000 LEDs per setup but, in reality, a computer is likely to struggle to control this many, and the USB controller is likely to have problems shifting this amount of data.

LET THERE BE LIGHT

Now, let's look at how this works in detail.

Many makers will be used to controlling LEDs from microcontrollers, so it might come as a bit of a →

DITHERING VS PWM

The on-board microcontroller uses pulse-width modulation (PWM) to achieve an 8-bit brightness level, where we send a number between 0 and 255 and the microcontroller turns the LED on and off rapidly, with the brightness level corresponding to the amount of time the LED is on.

Dithering involves the main controller on the FadeCandy sending instructions to the LEDs to switch between PWM levels very quickly. When you're controlling LEDs with a FadeCandy, you can use this approach to achieve more brightness levels than are possible with PWM using the NeoPixels.

surprise that FadeCandy is actually a USB device for controlling LEDs from a regular computer running Windows, macOS, or Linux. You can use a Raspberry Pi if you're after a small, embedded controller.

FadeCandy has a server that controls access to the hardware, and you can then write software in a range of languages that speak to this server and send frames to the LEDs. Before we get to the software, let's take a closer look at the hardware.

The FadeCandy board has eight outputs, and each output has two pins: one for data, and one for common ground. The reason we need a ground pin is because we need to power our LEDs separately. Although the FadeCandy gets some power from the USB, this can't provide enough power to run the LEDs, so we need a separate power source. This needs to be 5V, and needs to be able to provide enough power to run the LEDs (see previous Power section). This means we've got two power sources (the USB running the microcontroller on the FadeCandy and the external power running the LEDs) with the data line on the LEDs bridging between the two. Whenever we have two power sources connected like this, we need to ensure their grounds are connected.

So, the wiring is simple. The external power supply needs to send 5V to your LED strip, and ground needs to be connected to both the LED chain and the common ground pin on the FadeCandy. The data pin on the FadeCandy needs to be connected to the data-in on the first LED on the LED chain.

FadeCandy comes without any headers attached, so you could solder this directly onto the FadeCandy, but you may want to use a less permanent method of attaching the FadeCandy to the LEDs. We used wires on the FadeCandy, connected via crocodile leads to wires on the LED chain, but soldering directly onto the FadeCandy would provide a more robust link.

On the other side of the FadeCandy, the board connects to a computer via mini USB. If connecting many to a single computer, you can use a USB hub.

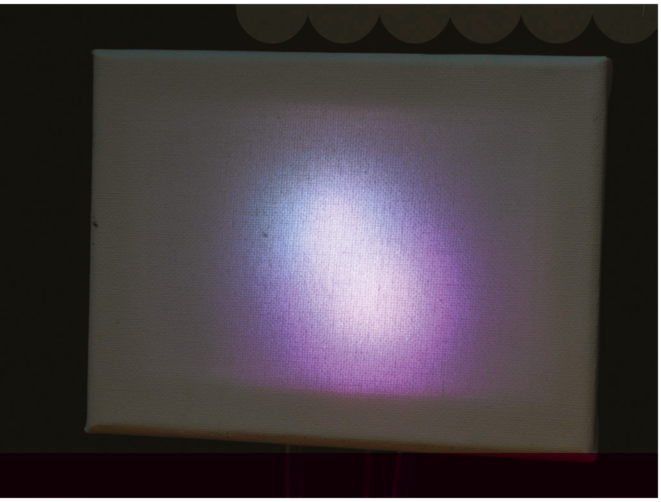
Now, with everything connected, we need to set up the software. The first thing we need running is the server. You can download it from: **hsmag.cc/AVAKjw**. Once you start this, point your web browser to **localhost:7890**, where you should see the server webpage, which should have identified your hardware. Here, you can run a test pattern to confirm that everything's set up and running correctly.

PROGRAMMING IN PYTHON

The FadeCandy uses the Open Pixel Control (OPC) protocol, and there's a Python module to handle this communication. Download it from: **hsmag.cc/JiuRdA.** (Note, this isn't the same as the OPC module in Pip, so make sure you get it from here rather than the package manager). Make sure that this is in the same directory as the Python script you're running.

As a simple test, the following code for an 8×8 matrix fades in and out rows of red, green, and blue, while a white pixel jogs through the grid.

Below ⧅
The Orbit example in processing shows two balls of light rotating around the centre



Above
When designing your animation, bear in mind that the diffuser will blur adjacent pixels together

```
#!/usr/bin/env python

import opc, time

numLEDs = 64
client = opc.Client(‘localhost:7890’)
line1=16
line2=31
line3=32
background = 0
increment = 1
pixels = [ (0,0,0) ] * numLEDs
white_posn = 0

while True:
        for i in range(8):
                pixels[line1+i] =
(background,0,0)
                pixels[line2-i] =
(0,background,0)
                pixels[line3+i] =
(0,0,background)

        if background == 255: increment = -1
        if background == 0: increment = 1
        background = background + increment

        pixels[white_posn] = (255,255,255)
        if white_posn == numLEDs: white_posn = 0
        white_posn += 1

        client.put_pixels(pixels)

        time.sleep(1)
```

The key parts for the FadeCandy are the line that establishes connection to the server:

```
client = opc.Client(‘localhost:7890’)
```

And one that sends an array of RGB values to the LEDs:

```
client.put_pixels(pixels)
```

The rest is code to control the main data structure – an array of RGB values.

You'll notice – particularly on the white dot – that it doesn't suddenly turn on and off, but fades in and out. FadeCandy interpolates between frames to make effects smoother, without you having to handle the fading in your code.

If you prefer to use a different language, it needs to be able to create a WebSocket connection to the FadeCandy server. There's example code for C++, C#, Java, JavaScript, Go, Perl, and Processing. →

PROCESSING

We've looked at Python because it's a great language for people new to programming and there's loads of resources to help you get started. However, FadeCandy works with a wide range of languages, and Processing is popular in the artistic community for its ability to work with colours and shapes.

Processing is currently at version 3.5, but the example sketches require version 2 to run. You can download this from **processing.org**.

Going into detail about programming in Processing is beyond the scope of this article, but we'll dip our toes into some code to see what's going on. There's a range of Processing examples that make a great starting point if you're looking to go down this route at: **hsmag.cc/JazjPC**.

These examples are named according to the hardware that they're designed to control. We tested on an 8×8 grid so, for example, the sketch at **hsmag.cc/DgxEWC** displays an image of two light sources orbiting each other. The **OPC.pde** file has the bits needed to create a canvas that's sampled at in 64 places. The **grid8x8_orbits.pde** file then draws on this canvas and the image is sent to the LED matrix based on the 64 samples.

The particular settings for the LED matrix are set up in the line:

```
opc.ledGrid8x8(0, width/2, height/2, spacing, HALF_PI, false, false);
```

Perhaps the most important entry here is the penultimate `false`. This is used to set whether your display is wired in the 'zigzag' fashion or not. If you find that the output on your LED matrix is muddled, this is the most likely culprit, so try changing it to `true`.

The code will look roughly familiar to Arduino users and is split up into `setup` and `draw` functions. The first is where you configure everything for your sketch. The second is similar to the `loop` function on Arduino and is called repeatedly. In this case, the `draw` function uses a little trigonometry to calculate the position of two images (that are loaded from PNG file in the `setup` method) as they follow a circular path around the matrix, and the position of the mouse is used to determine the size of the images.

If you want to find out more, peruse the other examples; or take a look at the library of excellent Processing videos at: **hsmag.cc/tPInFA**.

LASER LIGHT SYNTHS

Combining LEDs and lasers

Seb Lee-Delisle

Image
Rob Stanley

Let's now take a look at some awesome LED projects to see what can happen when you start to work with LEDs.

Seb Lee-Delisle is a BAFTA-winning digital artist and speaker, who specialises in large-scale interactive light installations.

His Laser Light Synths project is a touch-activated light installation comprising of thousands of super-bright LEDs. As you play the custom-made synths, powerful lasers project animated visuals that respond to the music and cover the entire building with shimmering dancing lights.

"I honestly think that everyone is innately musical. The brain power required to listen, process, and enjoy music is astonishing – and if you can enjoy it, then you can make it," says Seb. "These musical instruments remove all barriers – now everyone can express their individual musicality."

The synths have been designed to use a carefully selected musical scale, so that they remain perfectly in harmony with each other. This makes them easy to play (there are no wrong notes!), and yet expressive enough that you feel fully in control. →

Below ◈
For a smaller version of this light interface, we took a look at Adafruit's NeoTrellis last issue

Image
Becky Lu

LED CYLINDER

Shining a light through wood

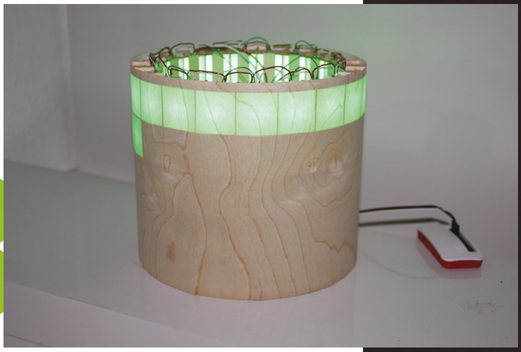
Martin Oehler

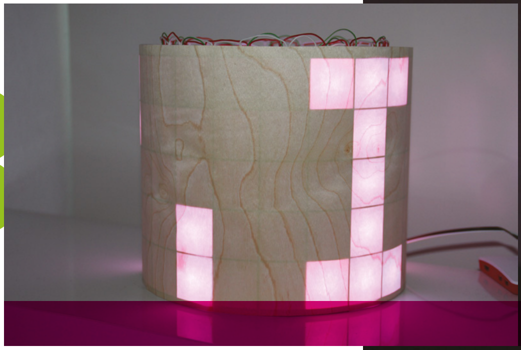
Martin is a 39-year-old electrical engineer working in the biomedical engineering field, widely interested in all sorts of tinkering, woodworking, and 3D printing.

For this project, he's used a Pi Zero W running a Python script based on the NeoPixel library. For simulation and animation scripting, he is using a Pygame-based simulator which he has written for a previous project (**hsmag.cc/AVYFIW**).

3D design was done in Fusion 360, and a 2D laser cutting file was preprocessed in Inkscape. Materials used include cardboard for the frame, veneer paper, and WS2812B strips. □

Below When not illuminated, the lamp looks like opaque wood



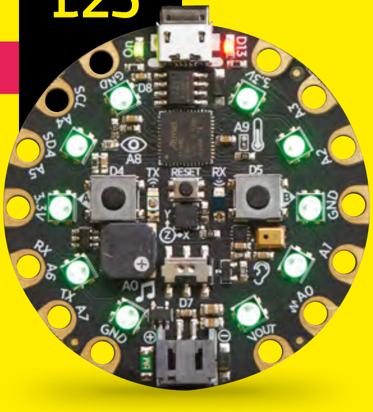


adafruit

CIRCUIT PLAYGROUND EXPRESS

WITH 12-MONTH PRINT SUBSCRIPTION

WORTH £25



MAKE BUILD HACK CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
March 2019
Issue #16
Extreme Electronics
Tesla coils, sparks, and high voltage
ACCESSIBLE CONTROLLERS
Helping disabled gamers join the fun
MAKE ART WITH LEDs
PLOTTER DRAWING
Beautiful pictures created with motorised pens
DRONE MAPPING
Taking to the skies to find the streets
ARDUINO SPONGES AMPLIFIERS RESIN

FROM JUST £55

12-month subscription from £55:

- → UK: £55 per year
- → EU: £80 per year
- → US: £90 per year
- → RoW: £95 per year

Offers and prices are subject to change at any time

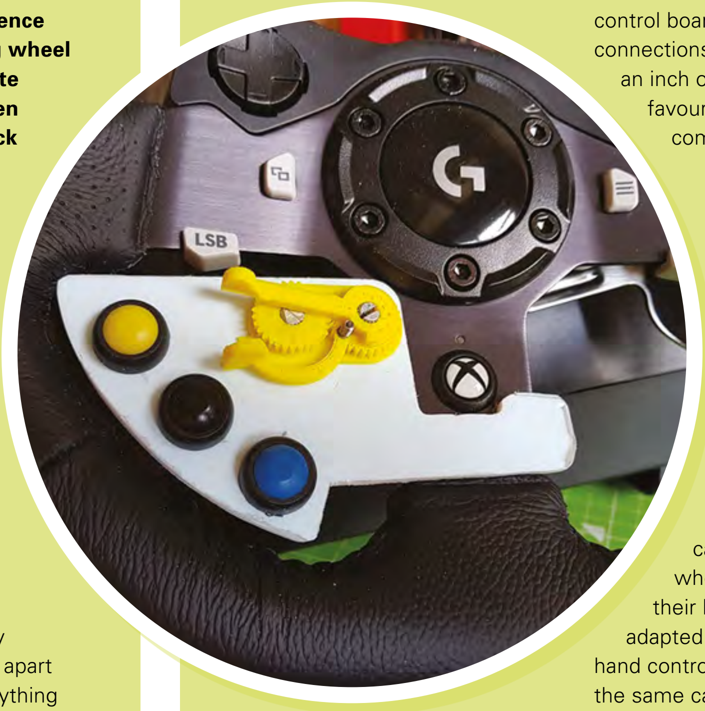
Visit: hsmag.cc/subscribe

How I Made
AN ACCESSIBLE GAMING STEERING WHEEL

Making games that more people can play

By Andrew Lewis

My first experience with a racing wheel was in the late nineties, when force feedback technology had just been released, and Need for Speed 3 was my favourite game. The wheel made the game so much better, but my wheel connected to the computer via a serial connection, rather than the new USB-style connection that was becoming popular. When I upgraded my operating system to the latest version, the drivers for the wheel were no longer compatible. With the technology now useless to me, my curiosity got the better of my judgement, and I decided to pull apart the controller, and see how everything worked. To my surprise, the controller was nowhere near as complicated as I had expected it to be. Even more surprisingly, I spotted an unsoldered jumper pad on the control board inside the wheel, with four connections labelled USB. An hour, and an inch of solder later, I was playing my favourite game again, with a USB-compatible race wheel.



Above
A modified race wheel pedal controller, fitted to a race wheel

Twenty years later, I had a conversation with my friend and colleague Chris Power. Chris is the vice-president of The AbleGamers Charity, and it turned out that my early experience with race wheels might be useful for adapting them for people who aren't able to use them in their default configuration. Like most game controllers, a race wheel (and an actual car) is designed for someone who has full control over all of their limbs. While cars can often be adapted to suit individual needs, with hand controls or levers instead of pedals, the same can't always be said of game controllers. Chris suggested that I might be able to help on one of their projects, and I was very happy to donate some time to a worthy cause.

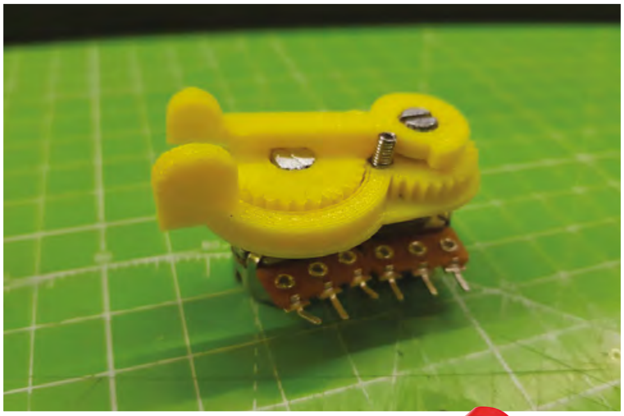
How the wheel works

This steering wheel modification replaces the accelerator and brake pedals of a Logitech race wheel, with hand controls positioned on the wheel itself. The main accelerator and brake are interlocked, so that pushing the brake lever down will also reduce the acceleration. This interlocking action can be disabled by removing the grub screw from the acceleration gear. In addition to a brake and accelerator lever, there are three additional buttons. These buttons are programmed to mimic certain common driving pedal states when they are pressed down:

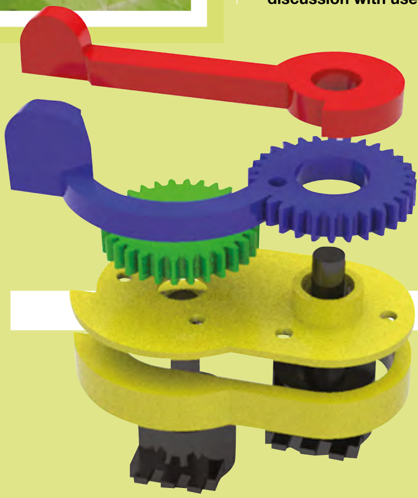
Turbo button
Sets maximum acceleration and zero brakes. In the default configuration, this button is located at the top of the column of buttons on the panel.

Coast button
Sets accelerator and brake to zero. In the default configuration, this button is located in the middle of the column of buttons on the panel.

Emergency Stop
Sets the accelerator to zero and brake to maximum. In the default configuration, this button is located at the bottom of the column of buttons on the panel.



Left
Final version of the lever design, before mounting in the control panel



Below
3D model of the modified lever design, after discussion with user

DEFINING THE PROBLEM

The temptation with any project is to dive in and prototype things. From experience, I know this is a bad idea. The first step in solving a problem should always be to define the problem clearly. In this situation, defining the problem meant having a conversation with the end-user, finding out the problem they're experiencing from their perspective, and seeing whether they have any ideas how that problem could be solved.

The problem was that the pedals weren't accessible to the user, and it wasn't possible to control the throttle and brakes effectively using the existing buttons on the wheel. I came up with the idea of a linked lever hand control, and I presented this to the user as a possible solution. After some more discussion, we decided that the lever control alone wouldn't be enough to solve the problem, so I added some more buttons to the design and asked whether the user liked my idea. I had some more conversations with Chris and the other volunteers at The AbleGamers Charity. We picked up on some issues that would make the project more flexible, and we came up with a workable specification that covered the installation, calibration, and use of the race wheel to play games. We wanted the wheel to be easy to install, easy to calibrate on different machines, and comfortable to use when gaming.

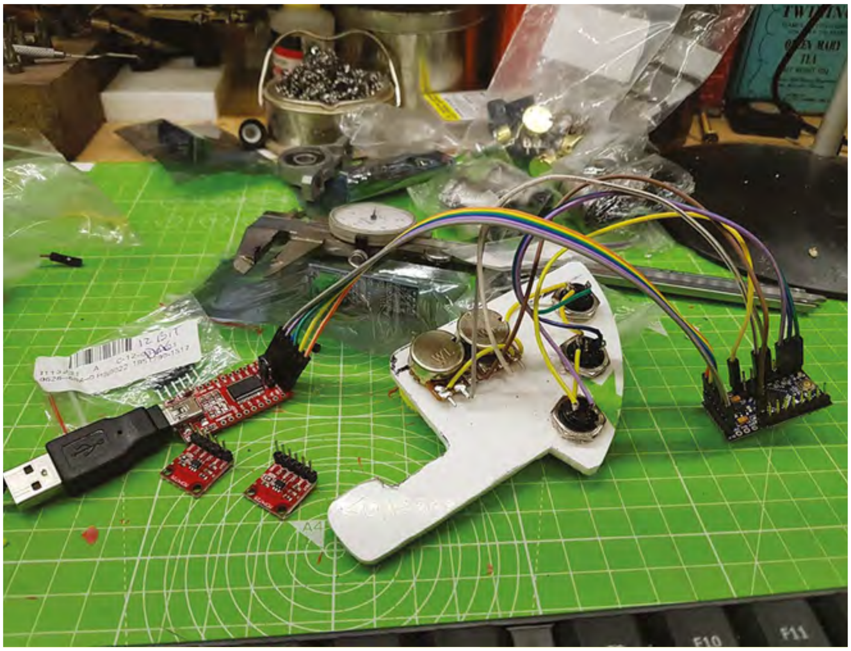
INVESTIGATING THE HARDWARE

Now that I had a design, I could look at the hardware and figure out how to implement it. The pedals on the race wheel were connected to the main wheel via a 9-pin D-Type connector, and I started to probe the pinouts from the pedals with my multimeter. I figured out that (as I had suspected) the pedal unit was entirely passive, and the pedals were just plastic shapes connected to potentiometers. The output from the potentiometers ran directly into the 9-pin connector on the race wheel. I recorded the maximum and minimum resistance of each pedal, and soon realised that the amount of movement needed would be too much to just move the potentiometers directly onto the wheel and connect levers to them. I could try to modify the movement range using gears, or I could use electronics to map the values from a different set of potentiometers to match the output range of the pedals. An electronic solution meant that I would be able to integrate button controls more easily so, after determining that the race wheel supplied 3.3V to the pedals, I ordered an Arduino Pro Mini 3.3V and waited for delivery. →

Left
The original lever concept described to the end-user

Using multiple MCP4725 boards

When using MCP4725 boards, the I²C address can usually be changed between two values by moving a solder jumper on the face of the board. This must be done on one of the MCP4725 boards so that each board has a different address when connected to the Arduino. You must also disconnect the pullup resistors from one of the MCP4725 boards by cutting the tracks to the resistors.



Above
The back of the panel during assembly, showing the switch and lever connections, and the MCP4725 boards ready to be connected to the Arduino Pro Mini

BUILDING THE SOLUTION

With some electronics hardware on the way, I started designing the lever assembly in Rhino 3D, and 3D-printing the parts. Now that I knew I was using electronics to convert the resistance levels, I had the freedom to choose good quality, compact potentiometers and start deciding what other components I would need to make the modification work as expected. The first problem to overcome was that the Arduino needed to mimic the effect of the pedals in a way that the race wheel could understand. I could have solved this mechanically by using servos connected to potentiometers, but I decided that more moving parts would increase the potential points of wear and failure. Instead, I set about designing a circuit that would simulate the effect of several potentiometers, using digital to analogue converters (DACs). A DAC takes a digital signal from the Arduino, and converts it into a voltage between 0 V and 3.3 V.

The MCP4725 is a common choice for this sort of application, so I connected two of them (one each for the brake and accelerator) to the Arduino, using the I²C system. The I²C address used on different brands MCP4725 boards can vary, so I made sure that I set a custom variable near the top of my Arduino code to make changing the address easy.

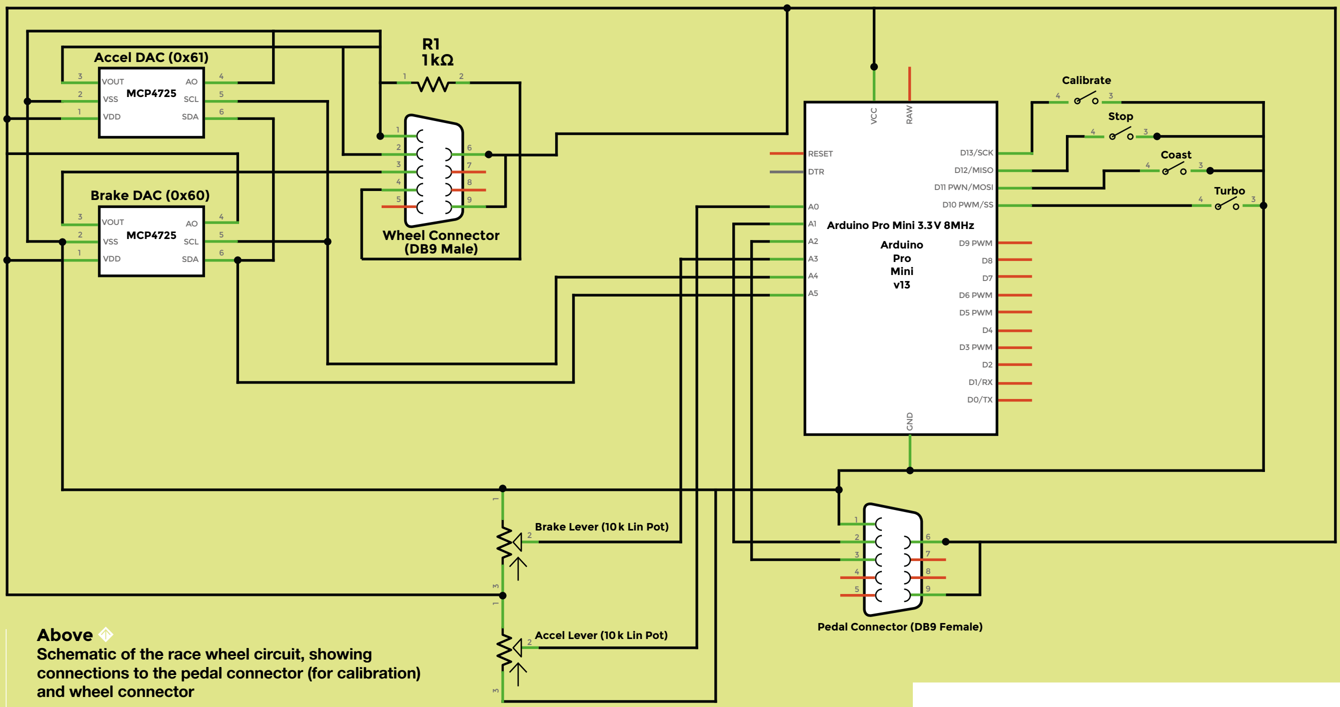
In addition to the throttle controls, the pedal modification also includes three buttons that mimic common foot pedal actions. The Turbo button sets the accelerator to maximum and the brake to zero. The Coast button sets both the accelerator and brake to zero. The Emergency Stop button turns the accelerator to zero, and the brake to maximum.

The next problem I dealt with was finding an easy way to calibrate the system. The modification to the race wheel would work if I hard-coded the conversion values for the pedals and levers, but it's possible that different race wheels will use different configurations. I wanted my modification to be easy to adapt and use on multiple devices, so I developed a software calibration system that only needs to be set up when the system is first fitted. The person calibrating the system attaches the pedals to the control box and flicks the calibration switch. Then, the user sets the pedals and levers to their lowest positions and pushes the Emergency Stop button. The pedal and lever positions are reversed, and the user pushes the Turbo button and turns off the calibration switch. This gives the Arduino everything it needs to detect the maximum and minimum values of the levers and pedals, and set the DAC values correctly when the user moves a lever or pushes a button. The values are stored in the Arduino's EEPROM, so it isn't necessary to recalibrate if the race wheel is unplugged from the computer.

> I set about designing a circuit that would simulate the effect of several potentiometers

FINAL TOUCHES

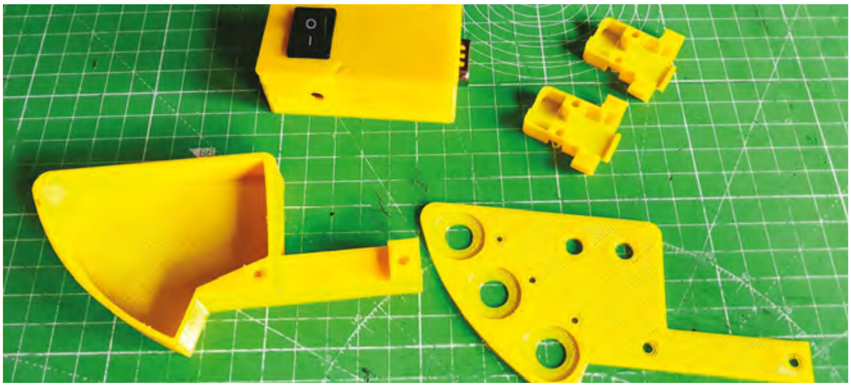
With the hardware working correctly, I started designing more presentable hardware. Up to this point, I'd been



Above
Schematic of the race wheel circuit, showing connections to the pedal connector (for calibration) and wheel connector

using a piece of scrap plastic to mount the switches and levers onto the wheel, so I 3D-printed a suitable panel and a box to hold the electronics. I used 9-pin D-type connectors to join the wires, so that replacing them would be easy in the event of a problem. I also made proper documentation for the project. It's unlikely that I'd be the person who had to remake or install anything, so I made sure that all of my code, parts lists, and diagrams were available for the volunteers at AbleGamers to remake and modify the project. I didn't need to make a final assembly of the project myself, because I knew each of the individual parts worked fine. It was more useful to hand the project over to AbleGamers, who could build a complete version from the instructions that I provided them with, and help find any mistakes in my notes. □

Below
The 3D-printed enclosures designed for the final assembly. I produced enclosures for the control panel, Arduino (with calibration switch), and also for the 9-pin connectors



The AbleGamers Charity

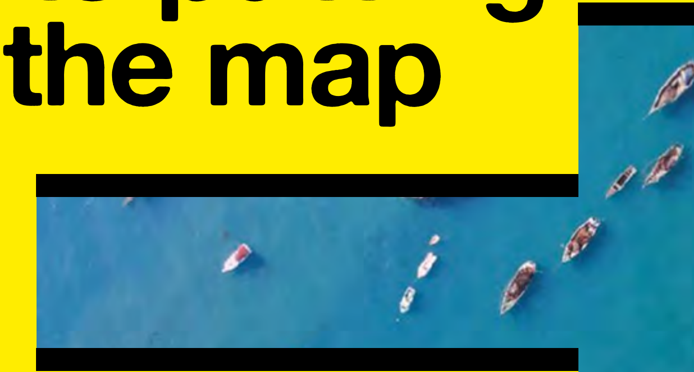
Founded in 2004, AbleGamers is a 501(C)(3) nonprofit charity that enables people with disabilities to play the games they love, or to play games for the very first time. The AbleGamers headquarters in West Virginia are home to Songbird Studio, a workshop with equipment for engineering and making assistive devices. AbleGamers has helped thousands of gamers through its grant program, by allowing people with disabilities to apply to receive free, custom-made assistive technology. The AbleGamers Expansion Pack program goes even further, by supplying hospitals and long-term living facilities with entire gaming room setups that include the assistive technology patients will need to game.

From working with the University of York on cutting-edge research, inventing innovative technologies with the world's best engineers, and deploying free industry-standard resources like **accessible.games**, AbleGamers continues to impact real change to a multibillion-dollar industry to be a more inclusive virtual playground.

Contact: ablegamers.org

The drone pilots putting East Africa on the map

The Zanzibar Mapping Initiative is the largest civilian drone mapping project in the world



Cameron Norris

@cameronsnorris

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

Seventeen spatial technology students from the State University of Zanzibar have spent over twelve months collecting thousands of aerial images to map the islands of Unguja and Pemba, in the Indian Ocean, off the coast of Tanzania in East Africa. The end result is a high-resolution 3D geographical map that covers more than 2300 km² of the Zanzibar Archipelago.

According to the United Nations, less than 10% of the African continent has been mapped at a local level, with many buildings, roads, and street names known only to local residents. Due to this lack of public geospatial information, urban planning has been incredibly challenging for Zanzibar, which experiences one of the highest population growth rates in the world.

Many urban areas have been constructed in previously undocumented flood zones, causing millions of pounds worth of damage to homes and infrastructure each year. Exposure to flood-water also puts Zanzibar's 1.3 million inhabitants at much higher risk of contracting life-threatening waterborne diseases, such as cholera, which is endemic to the region.

URBAN AREAS HAVE BEEN CONSTRUCTED IN ... FLOOD ZONES

AFRICA'S MISSING MAPS

To tackle this issue, the Zanzibar Commission for Lands and the World Bank launched the Zanzibar Mapping Initiative (ZMI). The goal of ZMI was to update Zanzibar's existing aerial imagery, using drones piloted by teams of students and local volunteers. Swiss company, senseFly, stepped in to provide a fleet of fixed-wing eBee drones, capable of mapping "areas of up to 7.5 km² in a single 50-minute flight". The data collected is now openly available for all purposes related to the island's conservation and development. This includes identifying areas that lack sufficient drainage, and deciding on the best locations to build new schools and hospitals.

"Traditionally, you would use satellite imagery, but it's not that effective in the tropics because of cloud-cover, and it's not very high-resolution," explains ZMI Task Team Leader, Edward Anderson. "Or you'd use manned aircraft. But that's not something you can do every six months because it is too expensive."

COMMUNITY MAPPING

The aerial imagery collected by ZMI's eBee drones has a spatial resolution of around 7 cm per pixel. In contrast, most areas covered by Google Maps satellite imagery have a resolution of only around 15 metres per pixel. The higher spatial resolution of the imagery gathered by ZMI's drones provides the detail required for accurate flood risk assessments, and can also be used to improve the accuracy of existing flood zone maps, which is not possible with low-resolution satellite imagery.

SenseFly eBee drones are constructed from ultra-light expanded polypropylene foam (EPP) and weigh just 700 g, despite a wingspan of 96 cm. Each drone includes a senseFly S.O.D.A. 20 MP RGB camera – the first camera to be designed for professional drone photogrammetry – and a 2.4GHz radio link, with a range of up to 4.97 miles. →

Below
A high-resolution aerial photograph above Zanzibar

Credit
ZMI

OUT WITH THE OLD

The last aerial mapping of the Zanzibar Archipelago dates back to 2004, and cost the Revolutionary Government of Zanzibar over £1 million due to the use of manned aircraft for aerial imagery. The data was not digitised, and quickly became obsolete due to rapid urban development.

BACK TO BASICS

A basemap is a map depicting geospatial information such as landforms, roads, landmarks, and political boundaries that help the user to orient the location of the map. The basemap also provides a framework onto which you can display dynamic data, such as seasonal water level fluctuations, or the expansion of urban areas over time.

Above ◈ **Khadija shows off the Cumulus One drone**

Credit Khadija Abdulla Ali CC-BY-SA

So far, the ZMI team has mapped all 239 zones on the island of Unguja, and 121 zones on the island of Pemba, with each zone representing a unique 9 km^2 parcel of land. All of the data collected is freely available from the ZMI GeoNode at **hsmag.cc/oJzXWv**, and can also be accessed on Android devices via the Zanzibar Mapping Initiative app, available from the Google Play Store.

SHARING IS CARING

Since being shared in the public domain, the data has been used to substantially improve Zanzibar's OpenStreetMap, and Municipal Councils have leveraged the insights gained from these mapping activities to develop sustainable urban planning strategies, in an attempt to reduce the impact of recurring floods, and avoid the spread of disease in densely populated areas.

The State University of Zanzibar has been widely praised for providing some of the most highly skilled drone pilots to participate in the ZMI project, including 27-year-old Information Technology and Application Management undergraduate, Khadija Abdulla Ali, known locally as the 'Queen of Drones'.

After volunteering, Khadija received training directly from senseFly on how to operate eBee drones for aerial mapping, using eMotion 2 flight management software. Now, she is a fully qualified eBee drone operator, with more than 1000 hours of professional flight time.

ALL OF THE DATA COLLECTED IS FREELY AVAILABLE

'QUEEN OF DRONES'

"I am very grateful to be part of the Zanzibar Mapping Initiative," explained Khadija, who last year signed a contract with Uhurulabs, in Dar es Salaam, to map mining areas in Northern Tanzania. "I want to be a mentor to other women by motivating them, and proving they can exist and work in the field of technology internationally."

Following the ZMI project, Khadija also volunteered with US-based not-for-profit organisation, WeRobotics, to help with disaster management in the Kagera Region of Tanzania, using Cumulus One and Parrot Bebop 2 drones, following a 5.9 magnitude earthquake that destroyed over 7500 homes and buildings. Khadija, and fellow volunteers, captured over 3000 high-resolution aerial images during a two-day trip to the worst affected area, around the lakeside town of Bukoba.

"I learned to believe in myself and work professionally, never scared to show the world what I can do to solve problems as a woman in technology," says Khadija.

The inspiration for ZMI came from a smaller open, community-based mapping project situated just 46 miles away, on mainland Tanzania. Dar Ramani Huria (Swahili for 'Dar [es Salaam] Open Map') was set up in 2015 to teach local volunteers the skills necessary to make highly detailed maps of the most flood-prone areas of the city. The maps were then combined with InaSAFE (free software that enables users to run realistic natural disaster scenarios), to provide insights into the likely impacts of future disaster events.

Right ◈ **Inspecting new maps during a Dar Ramani Huria conference**

Credit Dar Ramani Huria CC-BY-SA

Far Right ◈ **Testing new drones**

Credit Khadija Abdulla Ali CC-BY-SA

MADE BY THE COMMUNITY

"The map is a simplified way of providing open data, where everybody can see and learn. It's an easier way of communication, especially to the people at community level, who come from varying levels of education," explains Juliana Letara, Head of Department of Urban Planning for Kinondoni Municipal in eastern Tanzania. "Before this project, our basemaps were over 20 years old."

The 21 most flood-prone neighbourhoods of Dar es Salaam were mapped using the knowledge of local residents to identify buildings, street names, and flood-prone areas. "Many of those who are currently living here are new to this area," explained 69-year-old resident and volunteer, Monday Anthony. "They have no clear clue or idea of what floods can do to them when heavy rains occur."

Using a Bajaj Auto rickshaw, fitted with a 360-degree camera, volunteers were able to create 'street view' maps to more accurately identify and geotag the locations of those at greatest risk of flooding, particularly inhabitants living in unplanned settlements. "This is the best map made of this area, and it belongs to the community," says Juliana.

EQUITY AND OPPORTUNITIES

Dar Ramani Huria paved the way for future community mapping projects, like ZMI, by demonstrating how a few dedicated individuals can generate and manage large amounts of data that far exceeds anything previously available to 'experts' inside or outside of the region. "I think what is most important is that we have demonstrated proof of concept for how these things can be done," said Hassan Mshinda, Director General, Tanzania Commission for Science and Technology. "This will build stronger communities to promote equity and opportunities," adds Monday.

If you're interested in learning how to make your own high-resolution hyperlocal maps, visit **hsmag.cc/dOXytq** to access the training materials used by volunteers participating in the Dar Ramani Huria project. The training materials include details of multiple mapping techniques, and step-by-step instructions on how to use OpenStreetMap and InaSAFE with the data you've collected. □

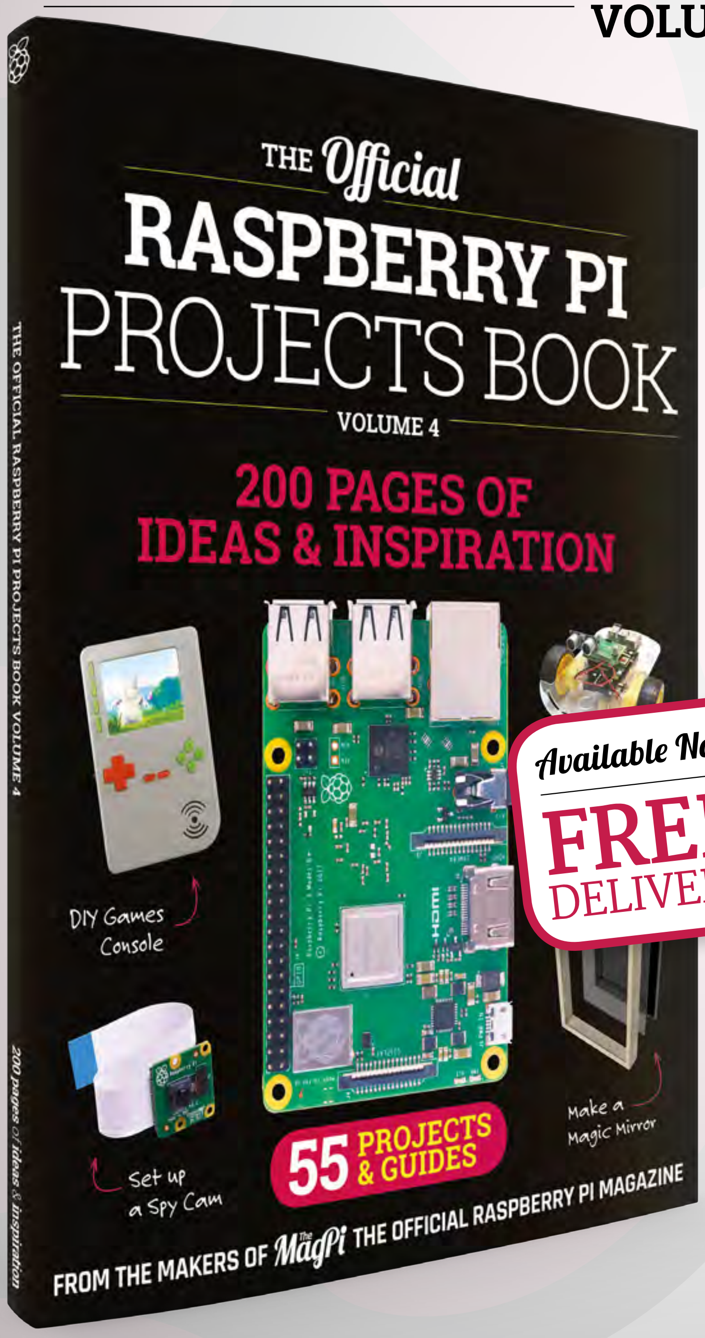
COMPUTER-AIDED GEOGRAPHY

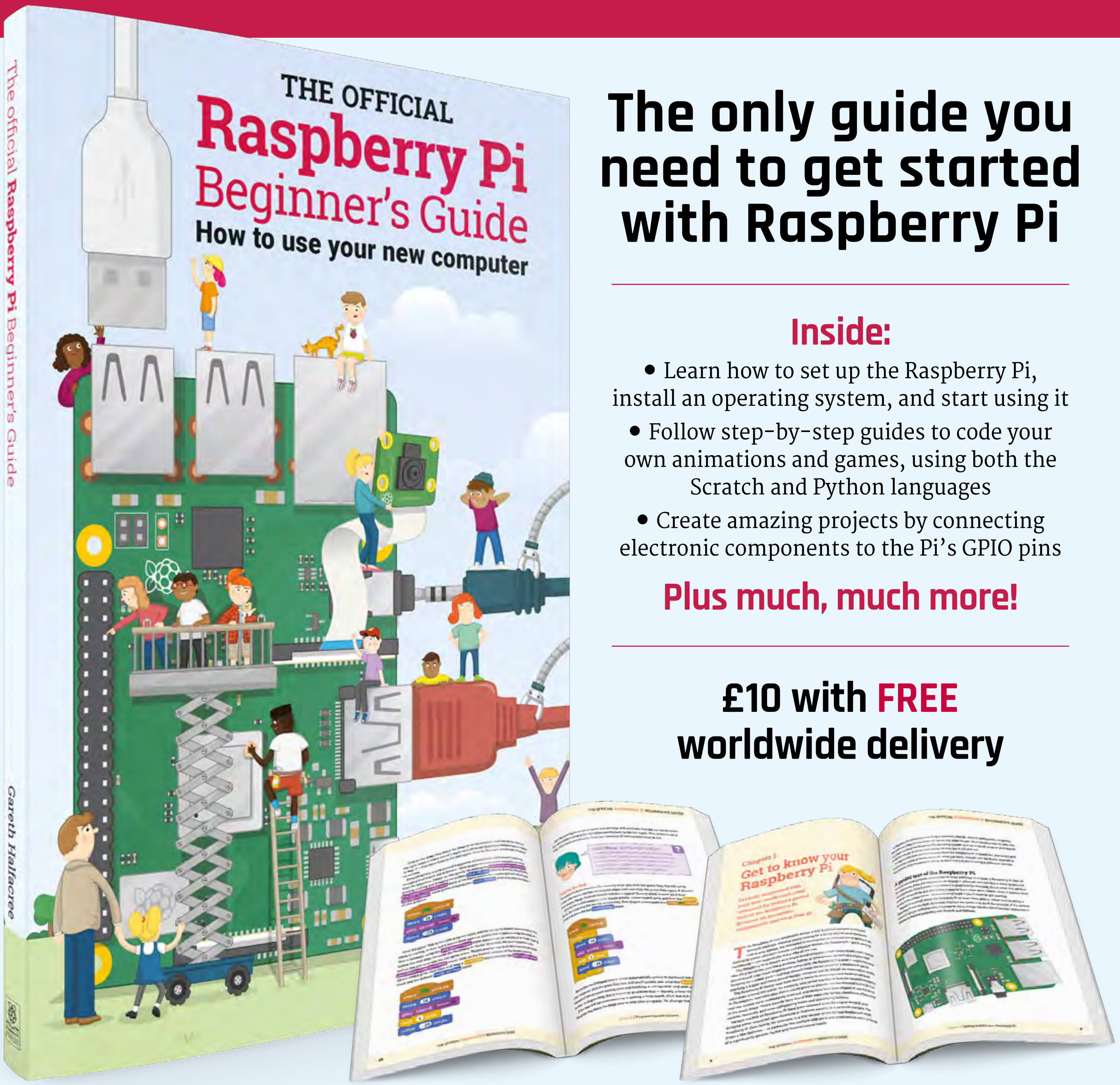
A Geographic Information System (GIS) is a framework used to gather, manage, and analyse data visualised in the form of an interactive map or 3D scene. The field was started in 1963 by Roger Tomlinson, who developed the first computerised GIS to provide the Canadian Government with a manageable inventory of its natural resources—Tomlinson also gave GIS its name.

Above ◈
Switching batteries ready for another flight

Left ◈
Looking for a suitable take-off point

Credits
Dar Ramani Huria
CC-BY-SA


THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 4
THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 4
200 PAGES OF
IDEAS & INSPIRATION
DIY Games Console
Set up a Spy Cam
55 PROJECTS & GUIDES
Make a Magic Mirror
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
200 pages of ideas & inspiration



THE OFFICIAL
Raspberry Pi
Beginner's Guide
How to use your new computer
The official Raspberry Pi Beginner's Guide
Gareth Halfacree
Chapter 1
Get to know your Raspberry Pi

HackSpace magazine meets...

Jasmine Brackett

The human face of Tindie tells us how to get our next product right

We're always thinking about our next side project, so this month we wanted to speak to someone who can help us sell things. As Product Manager at Tindie, the online marketplace for hobbyist electronics, Jasmine Brackett knows exactly what people are buying. She's got her finger on the pulse of the home electronics movement in the USA and around the world, so we thought we'd pick her brains on what Tindie, is and how we can make it work for us. →

Right Jasmine has left Britain behind for sunshine and sanity in California. Good work!

Images Rich Hogben

Left
Tindie's an ecosystem now; makers make products that support other makers' products

HackSpace We've mentioned Tindie quite a few times in the magazine without any real explanation of what it is. Let's start there…

Jasmine Brackett Tindie's a marketplace where people can buy DIY hardware and electronics. It works very much on a similar basis to Etsy. It's not an auction site or a crowdfunding site – it's for people who know what they've got, know how much it cost them to make, and how much they'd like to get for it.

For example, many hobbyists have an idea and want to create their own project – they then end up ordering more parts than they need because it's cheaper or only slightly more expensive. So they'll make the project for themselves, and have potentially the rest of the batch that they can sell on Tindie in the form of a kit.

And then there's other sellers, such as small businesses, who want to reach the maker market.

If you want to reach an audience that appreciates that some things are not always available off the shelf, or that require some degree of interactivity, you'll sell on Tindie – this is particularly the case for things for your house, such as home automation. For many of our buyers, it's a great point of interest that you can learn and play and tinker with stuff they've bought on Tindie, rather than having it in a tamper-proof smooth box. You're not just buying something with us; you're creating something.

HS How do you think it differs from something like Kickstarter? A lot of makers use crowdfunding sites to make sales, as well as for testing the market. How does Tindie compare to that?

JB We have quite a lot of sellers who use both Tindie and Kickstarter. With companies like Kickstarter or Indiegogo, there's a crowdfunding aspect, which we don't offer. We tried it once, but it didn't work.

The nice thing about Tindie is that you know that it's going to be shipped to you within two weeks – ideally straight away, but we do say within two weeks.

You find a lot of people selling on Tindie who are thinking about doing a Kickstarter or crowdfunding campaign; they'll test it out and maybe do some small batches to iron out the kinks with either their product or the production process, by testing out a couple of small runs. They can also beta-test the market and then, once they've got that going, they would launch on Kickstarter. The other thing that they might do is start a Kickstarter campaign and use Tindie to sell the extras. It's a good way for people to start off small.

> "The UK's got a great maker community with a lot of hardware creators; there are loads of people in their back rooms or sheds working on things"

HS Learning to scale is good.

JB People don't necessarily know how to scale. Tindie is a lot faster for people who are really interested in creating something, and just want a way to sell it.

Kickstarter and Indiegogo are all good, but they do lead naturally to setting up a business afterwards. And some of the people do want to do that on Tindie, but their approach tends to be more experimental. We've got quite a few sellers who will design a board, get maybe ten of them made, and then put them on Tindie to see how they sell. Then they will get more made as and when they sell out, scaling the production as they go.

HS I know that you're a US company; do you have many sellers based in the UK?

JB The UK's got a great maker community, with a lot of hardware creators; there are loads of people in their back rooms or their sheds working on things. There are definitely a lot of good things coming out of the UK and the rest of Europe. So, we have quite a strong community in the UK; whenever I go back there I try to meet up with a few of our sellers, some of whom you've featured. HackSpace magazine featured Spencer Owen, who makes the RC2014.

You've also featured Jason Hotchkiss, [who makes a load of music hardware under the name Sixty Four Pixels]. It's great to see what people make with stuff.

HS Do you have a support system for sellers? It seems like the average Tindie seller is a bit more switched on in terms of deliverability and knowing what's possible than some of the more pie-in-the-sky Kickstarter people. How much of that is down to Tindie itself?

JB It's down to the community. The community's really helpful. There's definitely a breadth of people who know about marketing, production, and also things like shipping. We do have help documentation available. Our team is quite small but we also have people available to answer questions that sellers might have. And then there's the forum, and also Hackaday.io, which is open to community content. And that's also generated products, because the people who run Hackaday.io publish projects, and other people read about them and say, "I really want that". Somebody else will say "Oh, you should put that on Tindie", so that generates products from people who didn't even know that their idea would be saleable.

HS What is the link between Supplyframe and Hackaday?

JB Supplyframe bought Hackaday maybe six years ago; then a couple of years →

after that they set up Hackaday.io. Then, in 2015, they acquired Tindie.

HS What's your role in Tindie? Product manager seems like a terrifyingly broad job description.

JB I make sure that everything runs. Our team consists of myself and one dedicated developer.

And then we've got one other part-time customer service person – I also help with customer service.

It involves working with various developers trying to make sure everything works, improving the package for sellers, and dealing with problems that come up. And then there's dealing with customer queries, and a bit of marketing too.

HS Can you tell us about any projects that are doing well on Tindie at the moment?

JB It's always interesting to go to **tindie.com** and just have a look at what's selling well. One of the products that's sold the most is the Raspberry Pi Camera HDMI extension cable. It's been on sale for years, but people are still buying it because there's a gap in the market for a quality bit of kit that accompanies an established hardware platform.

HS That sounds really simple; I'd expected it to be something like a retro computer or something smart-home-related.

JB The RC2014 is also a very good seller, as is Jason Hotchkiss's audio kit. Actually, Spencer's RC2014 has spawned a whole market segment of add-ons for compatible kit by other sellers. There are maybe five or six people making add-on modules for his computer.

HS Do sellers collaborate to help each other out, or is it more of a free-for-all?

JB Sometimes. Most of the [makers of] products to do with RC2014 work with Spencer; he wants to make sure that their kit is as good as it can be so that it improves the market for his product, and naturally the makers themselves want to be as good as they can be.

Occasionally people make clones of his work, so we have to be a little bit careful about that. But he's worked out a system where people can buy in confidence, knowing that any modules will be compatible with his stuff, even though they're not made by him.

Even though Tindie itself is a niche, there are other niches inside it, and the sellers are helping each other out as well.

> "Even though Tindie is a niche, there are other niches inside it, and the sellers are helping each other as well"

The Mooltipass is also worth looking at as an example of how things can work: that was developed on Hackaday.io. It's a hardware password wallet that came about on Hackaday.

The guy who created it [Mathieu Stephan] is an ex-Hackaday writer, and he's producing far more than he expected to. There are a lot of other sellers who reach the limits of their capacity and don't quite know what to do; he's got a factory working on it.

HS Right, that's kind of how I imagined it. I don't think anyone on Etsy has got a factory.

JB: There are some sellers who tend to be involved in electronics in their day job. They've worked out how to build and test things to a professional level. That's one of the things we're working on with our partners: how to get that knowledge spread out to the rest of our sellers.

Because there are a lot people with regular day jobs who don't understand →

Left

42.5% of Tindarians are in the USA; 7.5% are in the UK, and 26.5% are in Europe

Right ↗
When you buy on Tindie, you know you're buying something that exists. That's a big win

things like supply chains, they make a sale and it depends on a part, but then the wrong part comes and all of a sudden they're unable to fulfil their orders. They tend to unfold a little bit because there's other things going on with their lives. So, we're able to help them with production queries so they can carry on creating.

Definitely one of the things we tell the sellers is, we cannot compete with the large manufacturers; that's not why people are buying their things, because they're cheap. They're buying their things because they're unique, just like Etsy, because they're handmade and there's care taken over them. If the buyer gets a sense of who the person is who's behind the product, then you're more likely to buy it.

HS That brings me on to my next question: you mentioned a couple of things there, but do you have any top tips for someone who wants to get selling on Tindie?

JB The number one rule of selling anything on Tindie is to make sure you've thoroughly tested the product.

Documentation is key: clearly describe what your product is, and provide good instructions that show how to use it. Lots of engineers will tend to describe the capabilities – they describe features, rather than function.

Having good photos is also very important, so the buyer can see clearly what they're getting.

And videos are really helpful – it just helps to give people as much information as possible.

People love to hear about the person they're buying from – if you can get a few details about yourself on there, it shows that they're buying from a real person.

Understand your local shipping system – how much is it going to cost to ship that stuff? The system we have at the moment isn't perfect, but we're trying to integrate local shipping options.

Because sellers are based all over the world — our largest market is the US; UK [and] Europe aren't far behind – American sellers quite often don't understand that, when they are selling to Europe, Europe tends to levy import fees. Everybody gets a bit upset when that happens.

> "The number one rule of selling anything on Tindie is to make sure you've thoroughly tested the product"

Understanding how to ship things, package sizes, putting an invoice on the outside of the package rather than boxing it up; we do have help documents for that sort of thing. But only you can understand how much it's going to cost to post from your local post office.

If they don't want to ship insured, we encourage them to raise their prices a bit, just in case a package goes missing, or something gets broken – you never know what's going to happen, so it makes sense to cover for unexpected eventualities.

HS It sounds like you've almost got a small business advice centre for people who aren't used to it.

JB We do try where we can. But, like I said, because our sellers are based all over the world, it does make it a little tricky to know what's going on in some of those countries. We are usually able to put them in touch with groups within their own country, so people stay in touch with each other and help each other out. □

SYRINGES

Inject a dose of creativity into your builds

Mayank Sharma

@geekybodhi

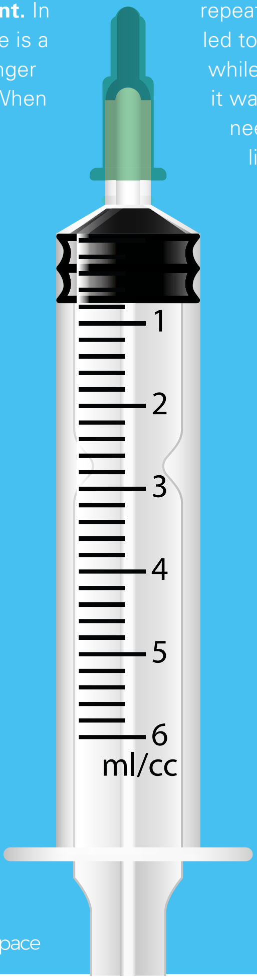
Mayank is a Padawan maker with an irrational fear of drills. He likes to replicate electronic builds, and gets a kick out of hacking everyday objects creatively.

The projects in the following pages all use needleless syringes. In case you can't find these at the local pharmacy, make sure you carefully remove and dispose of the needle from the disposable syringes. Also, reusing syringes (even for non-medical uses) is a bad idea, so make sure you always use fresh syringes.

No one has fond childhood memories of syringes, but no matter how many kids shriek at its sight, the syringe is an indispensable piece of medical equipment. In terms of its build, the device is a simple reciprocating pump that has a plunger that fits cosily inside a cylindrical barrel. When pulled, the plunger creates suction – and exerts pressure when it is pushed. With these two actions, syringes are used to either inject or extract fluids through the skin, flesh, or veins.

While syringes these days are mostly used to administer or inject medicine, they were actually designed as extraction devices to remove harmful substances like pus and poison. The exact date of the invention of the syringe has been lost in the annals of time, but there are records of Hero of Alexandria, the Roman mathematician and engineer, describing the construction of a syringe-like device he called a 'pyoulcus', which roughly translates to 'pus puller'. Perhaps the oldest record of the use of syringes dates back to the 1st century CE. Aulus Cornelius Celsus, in his medical encyclopedia titled *De Medicina*, described procedures to clean wounds and remove pus using a syringe.

The early syringes were made out of either metal or glass. The 9th century ophthalmologist Ammar ibn Ali al-Mawsili devised a method to use a glass syringe to suck cataracts from the eyes. But these early syringes were expensive, were used repeatedly and, despite regular sterilisation, led to the transmission of diseases. Also, while they were good at extracting things, it was the invention of the hypodermic needle that allowed doctors to inject liquids into something other than pre-existing openings.



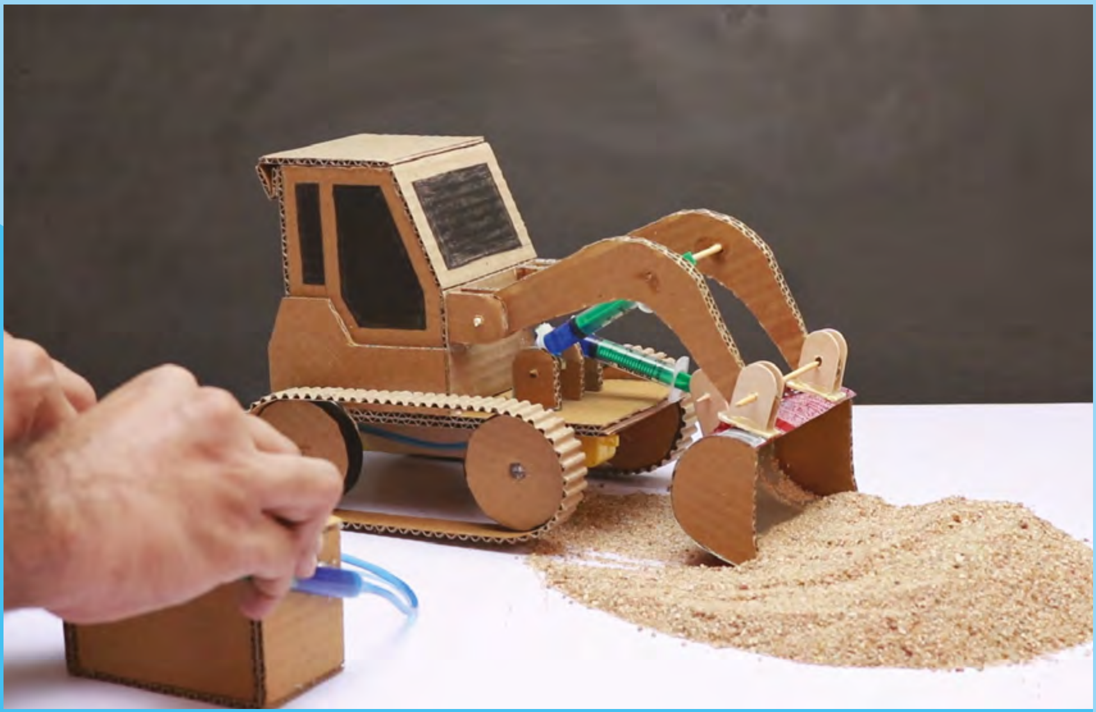
The first truly disposable syringes that were produced in large quantities were designed by James T Greeley in 1912. These were collapsible tubes made of tin (a bit like superglue or toothpaste tubes) that contained a certain amount of morphine for subcutaneous injection on the battlefield, and were very useful in World War I. However, it took several more decades before disposable hypodermic syringes entered the mainstream. Thanks to penicillin clogging up the glass syringes over repeated use, Charles Rothauser used polyethylene to make one that was cheap enough to be discarded after a single use.

Syringes have been a vital part of the medical field, but they do have non-medical applications as well – they have enough physics to be put to interesting uses.

SYRINGE-POWERED JCB

Project Maker
Akash Vaghani

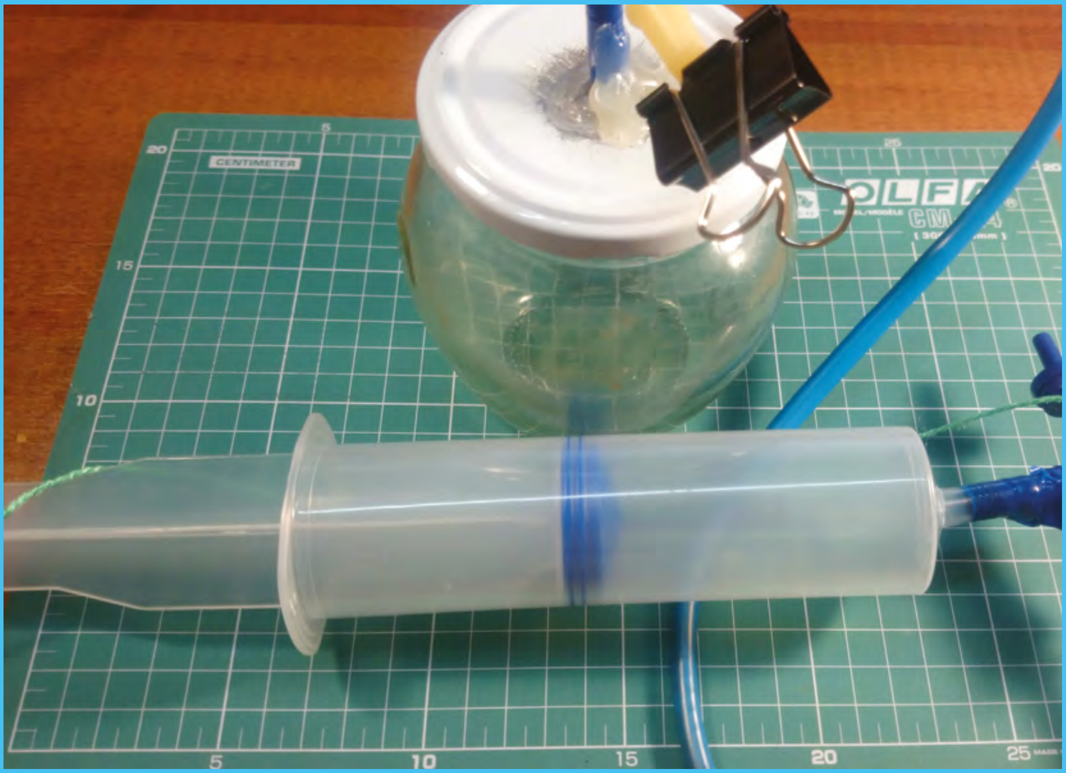
Project Link
hsmag.cc/ihkEYR



Left
The JCB sits on a crawler that's propelled by a couple of DC motors for forward and backward movements

We've had our fair share of RC cars, but none was as cool as this fully functional JCB. A major part of the build involves assembling the body of the digger, using pieces of cardboard and hot-gluing them together. The crux of the project, however, is its two hydraulic arms that move the bucket and lift the loader, and are made using syringes. Akash's system requires a total of four syringes. One is attached to the bulldozer's loader arm, and another to the bucket. The other two are used to control their movements. For the hydraulics, a pair of syringes are connected via a pipe and filled with coloured water. When you push the plunger on the control syringes, it'll push the plunger on the other that's connected to the bucket and the loader arm of the JCB. Inversely, when you pull the controller plunger, it'll pull the one connected to the JCB. You can now use them to mimic the motions of a real JCB digger. □

DIY VACUUM CHAMBER



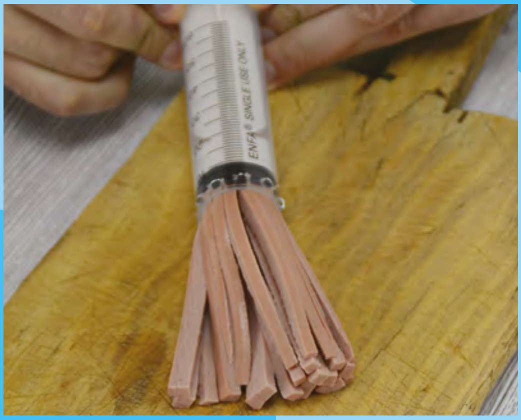
Project Maker
ALEXANDER MAYOROV

Project Link
hsmag.cc/RlufXR

Left You can use the pump to create enough pressure to inflate balloons and boil some water

This one is for all you STEM enthusiasts. If you need a portable DIY vacuum chamber to demonstrate what happens in a low-pressure environment, Alexander shows you how to build a hand vacuum pump that can create a low pressure inside small chambers. He uses a 150 ml syringe, a couple of two-way valves and T-connectors, a drop counter, and a length of silicone tube. If you can't find the tiny valves and T-connectors in your local hardware store, Alexander suggests looking for them in a pet shop that stocks or repairs aquariums. He explains the entire process of using these bits and pieces to assemble the pump, which is fixed atop a glass jar with a lid. Since it's unsafe to open the vacuum chamber while it's vacuumised, Alexander shows you how to create a vacuum release valve using a paper clip. You can now use the pump to create a low pressure of up to 0.2 bar, which is enough to demonstrate the characteristics of a low-pressure environment. □

DIY FOOD SLICER



If you love cooking, Denny has a useful hack for you.** He shows you how to use a syringe and a steel wire to julienne vegetables in a jiffy. Take a syringe with a detachable plunger and a wide barrel, lop off its top, and then sand the rough edges. Now, wrap electric tape around the freshly sanded top and use a calliper to mark equidistant holes around the syringe. Now, use a drill to make the holes and carefully weave a steel wire through them, forming a checked pattern. This is the most time-consuming and error-prone step of the entire process. Once you've woven through all the holes, loop the wire across the barrel, and make sure it's firmly in place before snipping off the rest. That's all there is to it. Now, remove the plunger, put in a piece of vegetable or sausage, and press the plunger to force it through the woven steel wire. □

Project Maker
Denny

Project Link
hsmag.cc/CnmpWI

Above At the beginning of the video, Denny shows you how to use a syringe to create a height-adjustable microphone stand

UNIVERSAL GRIPPER

Amazed by videos of the universal gripper made by iRobot, Charles Ford decided to build his own: "I kept thinking how useful it could be for people who can not easily deal with pills. The idea that a robot arm could pick up small items and deliver them to someone bedridden or physically handicapped motivated me." To complete the build with less power and complexity, he thought of using syringes, inspired by their use by special make-up effects creator Rick Baker for *An American Werewolf in London*. Charles has a detailed list of parts in his Instructable, and explains all the construction steps. He went through several revisions and has illustrated his progress with photos and videos. His universal gripper can pick up very irregular or smooth objects easily, despite the fact that it uses off-the-shelf syringes instead of expensive vacuum pumps and reservoirs. □

Project Maker
Charles Ford

Project Link
hsmag.cc/cEssbC

Left Charles got the large syringe from a veterinary supplier who was taken aback when told that it wasn't for his horse, but his robot!

MAKE BUILD HACK CREATE
HackSpace
TECHNOLOGY IN YOUR HANDS
hsmag.cc
March 2019
Issue #16
ACCESSIBLE CONTROLLERS
Helping disabled gamers join the fun
Extreme Electronics
Tesla coils, sparks, and high voltage
MAKE ART WITH
LEDs
DRONE MAPPING
Taking to the skies to find the streets
PLOTTE DRAWIN
Beautiful pictures cre with motorised
ARDUINO SPONGES AMPLIFIERS RE
REVEALED
JET ENGINES
FLAMETHROWERS
GIANT ROBOTS!
PLUS
THE EXPLOSIVE INVENTIONS OF
COLIN FURZE
BACTERIA NEOPIXELS MOTORS CANDLES

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 101.8: Amplifiers

Making signals stronger/louder

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**

An amplifier makes a signal 'bigger' in some way: bigger current, bigger voltage, or bigger power (which is voltage × current). We've seen transistors being used to amplify current when we first looked at transistors in HackSpace issue 13. We used a low current signal from a microcontroller (a milliamp or so) to control 1 amp for an LED strip.

An audio amplifier often amplifies voltage: turning a small signal into one large enough for further processing. The final stage of an audio amplifier will also amplify current for the purpose of moving a speaker to produce the required volume of sound. If you are familiar with audio amplifiers, you'll recall that they are generally rated in terms of power. From a few watts for a pair of powered speakers for your computer, to a 700 watt home theatre system, to a multi-1000 watt concert system; the principle is the same, just the voltages and currents vary. With higher power comes larger transistors to handle it. Depending on the efficiency of the design, those transistors can get quite hot. This is why you often see heatsinks on the back of amplifiers.

An electric guitar amplifier is a great example of a multi-stage audio amplifier. The signal coming from the guitar is very small, produced by the strings vibrating in a magnetic field and inducting a tiny current in the coils wound around the magnets (these are the 'pickups'). This signal has to be amplified before the signal is strong enough to drive a speaker, let alone a stadium sound system. There's usually a 'pre-amp' that amplifies the signal enough to feed into further stages. Between the pre-amp and the power amp (whose output drives the speaker), you will often find various effects units that modify the signal in different ways.

QUICK TIP

By cascading PNP and NPN transistors, we can cancel the base-emitter offset.

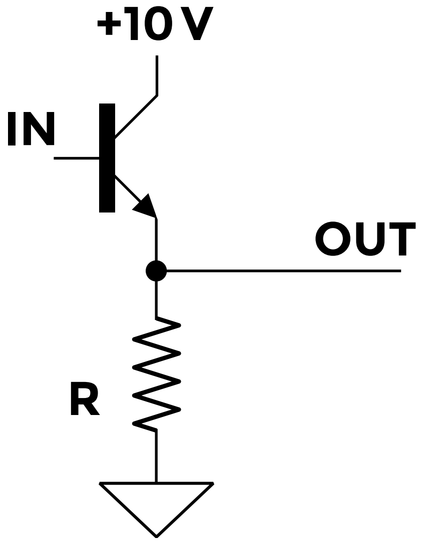


Figure 1 Simple emitter follower circuit

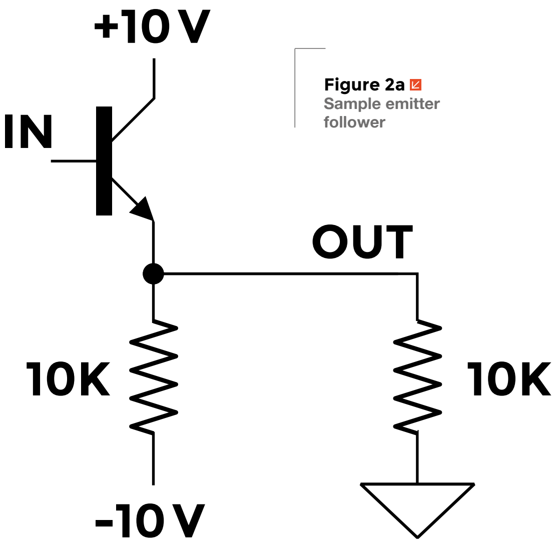


Figure 2a Sample emitter follower

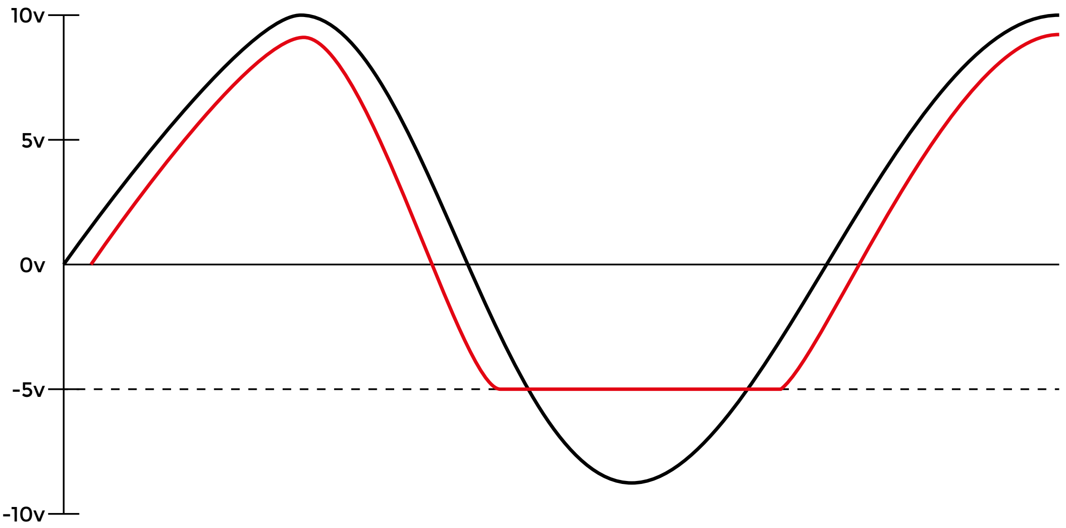


Figure 2b Input and clipped output of a naive emitter follower

The key point to remember is that amplifiers turn small signals into bigger ones.

THE EMITTER FOLLOWER

This circuit is a current amplifier. The output signal is actually lower in voltage than the input by 0.6 V (due to the base-emitter junction). However, the current available by way of the collector-emitter path through the transistor can be much larger. In our discussion of transistors, we saw that this is often around 100×. **Figure 1** shows a simple emitter follower circuit. Since this uses an NPN transistor, current only flows in one direction, which can cause some problems. Consider the circuit in **Figure 2a**. It assumes an effective load of 1 0kΩ, which is matched by the emitter resistor. These two form a voltage divider between -10 V and 0 V when the transistor is turned off. This means that the output can't go below -5 V. That, in turn, means that the transistor will turn off when the input (at the base) is -4.4 V. Since an emitter follower has a gain of about 1, the output vs. input voltages (given a nice sine wave as input) is shown in **Figure 2b**. This clipping is far from ideal, and means that a simple emitter follower can be problematic.

AC-COUPLING

In audio circuits, we typically want to deal with an AC signal. Recall that a capacitor will pass a varying signal while blocking any steady voltage in it. Essentially it's a high-pass filter: the component of the signal that doesn't change (i.e. has a frequency of 0) is filtered out. **Figure 3** shows an emitter follower with its input and output AC-coupled.

While we want to be working with this AC signal, it presents a wrinkle: we need an amplifier that uses both positive and negative voltages. A single supply amplifier will only work with the positive half of the signal. See **Figure 4** (overleaf).

If we use a dual supply emitter follower as discussed earlier (shown in **Figure 2a**), this will work with the condition that the signal stays near 0 V so as to avoid the clipping shown in **Figure 2b**.

> " **An audio amplifier often amplifies voltage:** turning a small signal into one large enough for further processing "

BIASING

An alternative to using a dual supply is to bias the transistor so that with the input at 0 V, the output sits at half of Vcc. **Figure 3** illustrates this. Resistors R1 and R2 form a voltage divider that biases the transistor appropriately. Because the output varies around half Vcc, the →

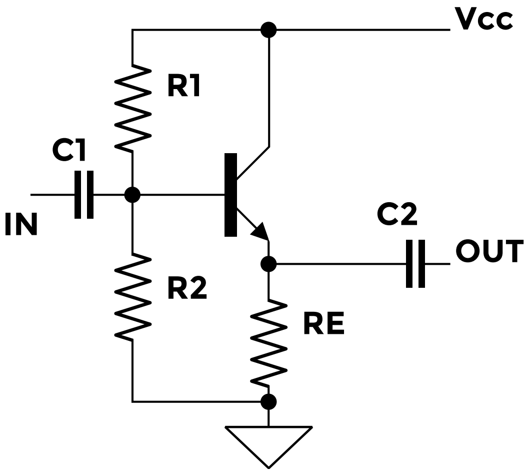
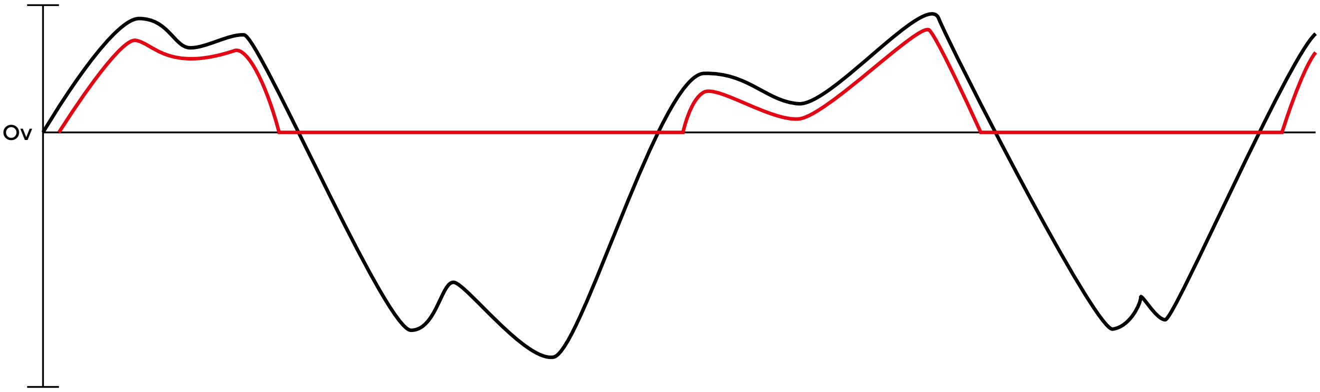


Figure 3 AC-coupled emitter follower

YOU'LL NEED

- **Solderless breadboard**
- **2N3904 and 2N3906 transistors**
- **Assorted resistors, as shown in the circuits you want to build**
- **Assorted capacitors, as shown in the circuits you want to build**
- **8 Ω speaker**
- **Signal source** (frequency generator or sound source)
- **Matched NPN/PNP power transistors** (e.g. TIP31C/TIP32C)
- **2 × 1N4001 diodes**
- **Power supply, dual voltage required for some circuits**



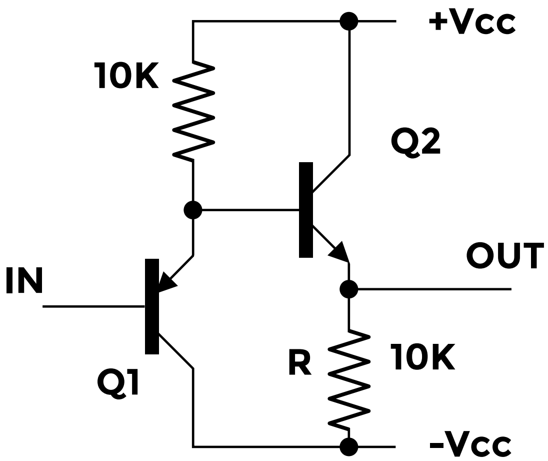


Figure 4 The result of a positive only amplifier

Figure 5 Vbe cancelling emitter follower

output has to be AC-coupled as well to remove the DC bias and pass the AC signal through.

AVOIDING THE VBE OFFSET

Recall that the output from a simple emitter follower is 0.6 V less that its input due to the voltage drop across the base-emitter junction. We can adjust for that by using a couple more parts. Essentially we use the opposite voltage drop of a PNP transistor to cancel the drop in the NPN transistor. **Figure 5** shows the updated circuit. The emitter of Q1 is 0.6 V higher than its base (higher since it's a PNP transistor). That feeds the base of Q2. Since it's an NPN transistor, its emitter is 0.6 V below its base. So the output signal (at Q2's emitter) is approximately the same as the input signal at the base of Q1. As before, this only works if the input is kept near 0 V so that the output doesn't clip.

COMMON EMITTER AMPLIFIER

Consider the circuit in **Figure 6**. The voltage at the collector is below Vcc by the voltage across Rc, which is IcRc, so Vc = Vcc – IcRc. The voltage at the base determines the current through the base-emitter, which determines the current through the collector-emitter. Since Vcc and Rc are fixed, if the base-emitter changes then so does the collector-emitter current and so does Vc: a small change in Vb causes a change in Ibe, which causes a larger change in Ic and thus a larger change in Vc. Note that increasing Ic causes a larger voltage drop across Rc which lowers Vc: i.e. changing Vb causes a corresponding (but amplified) decrease in Vc. Because of this we consider the gain of this amplifier to be negative.

Let's expand on the circuit in **Figure 6.** A concrete common emitter amplifier circuit is shown in **Figure 7**. R1 and R2 form a voltage divider that sets the voltage at the transistor's base. The stated values of R1 and R2 mean that 20 V / 120 kΩ = ~1.6 mA are flowing through them, resulting in a voltage of ~1.6 V (~1.6 mA × 10 kΩ) at the transistor's base. This is an

QUICK TIP

To amplify an AC-coupled signal, you can use a power source with both positive and negative Vcc.

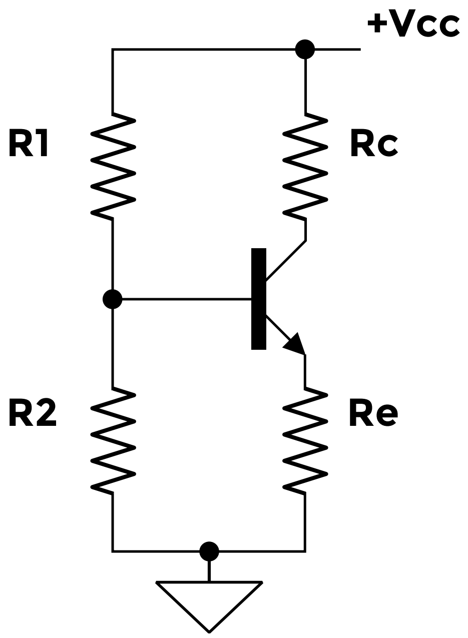


Figure 6 Current source driving a resistor (Rc) as a load

NPN transistor, so the voltage at the emitter will be 1 V due to the base-emitter drop. That means there is 1 V / 1 kΩ = 1 mA flowing through the emitter. That also means that there is roughly 1 mA flowing through the collector. 1 mA flowing through the 10 kΩ Rc results in 10 V across Rc. That means that Vc is the 20 V Vcc less the voltage across Rc: 20 V – 10 V = 10 V. That's the steady/rest state, with no input signal effecting Vb.

Notice the capacitor (C) that is used to AC-couple the input signal? We do this to block any DC bias on the signal so that it won't affect the biasing of the transistor, but will let the actual signal though. We want the signal to affect the biasing so that the transistor will amplify it. How is the value of C chosen? Why 0.1 μF? C forms a high-pass filter with R1 and R2, so a value is selected that lets the frequencies we want through. Remember that the goal is to filter out any DC bias in the input, so we want the filter cutoff to be quite low. The values given in the circuit give a cutoff frequency of approximately 17 Hz: f = 1/(2πC(R1||R2)) = 1/(2π×.000001×9167) = 17.36.

Now, because the transistor is biased to have the output sitting at 10 V with no signal, variations in the base voltage/current caused by a signal coming through the capacitor will change the collector voltage in a corresponding way, but in the opposite direction. Since the output will vary from 10 V, we will likely want another coupling capacitor on the output to remove that 10 V component.

> "The goal is to filter out any DC bias in the input, so we want the filter cutoff to be quite low"

DIFFERENTIAL AMPLIFIERS

There are cases where we send signals on a pair of wires. High-speed digital and RF signals are commonly transferred this way. An example is an Ethernet cable for a local area network. The two wires are quite often twisted together; you might have heard the term 'twisted pair' – this is what it means. **Figure 8** shows the wires inside an unshielded twisted pair UTP cable, typically used for Ethernet.

Why pairs, and why twisted together? Think about Ethernet cables: 3, 5, 8 metres long →

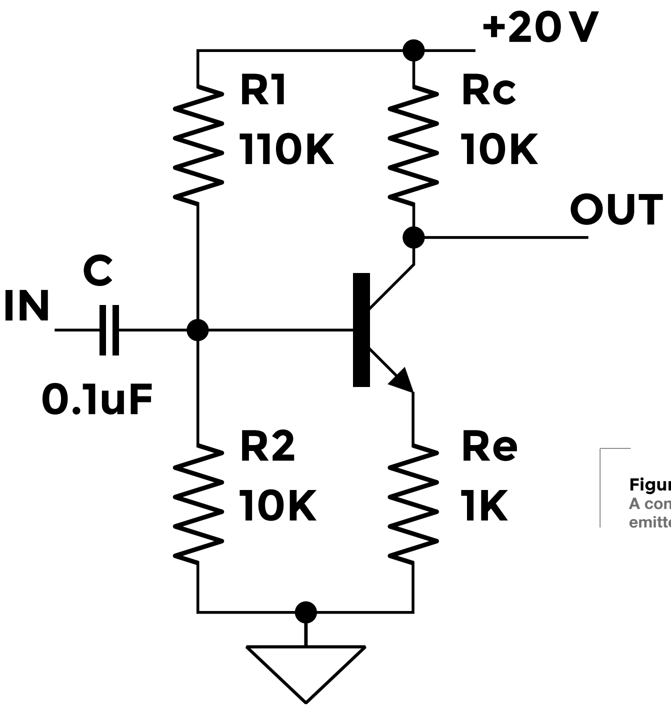


Figure 7 A concrete common emitter circuit

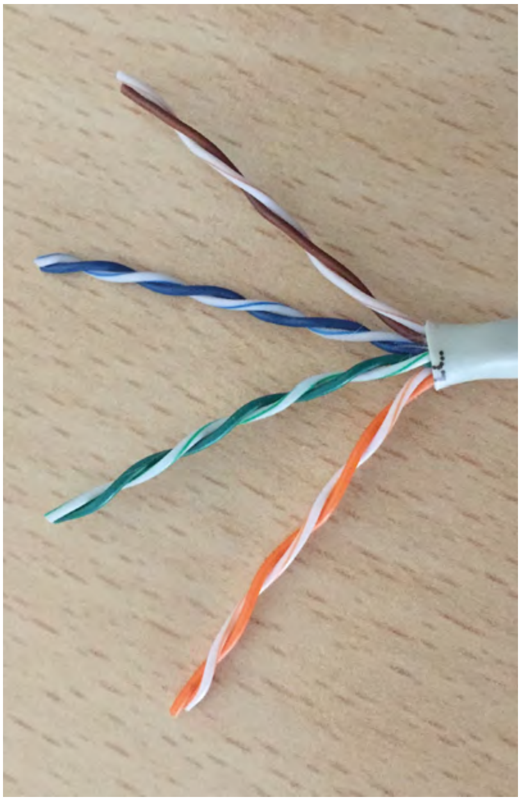


Figure 8 An unshielded twisted pair UTP cable

QUICK TIP

Using a voltage divider to set the base voltage is referred to as biasing the transistor.

COMPARATORS

Because its output reflects the difference between its inputs, a differential amplifier can be used to compare two voltages. Using its output to control a switch transistor results in a binary indication of the relative values of its inputs. A suggested exercise for you is to take the general differential amplifier and do this.

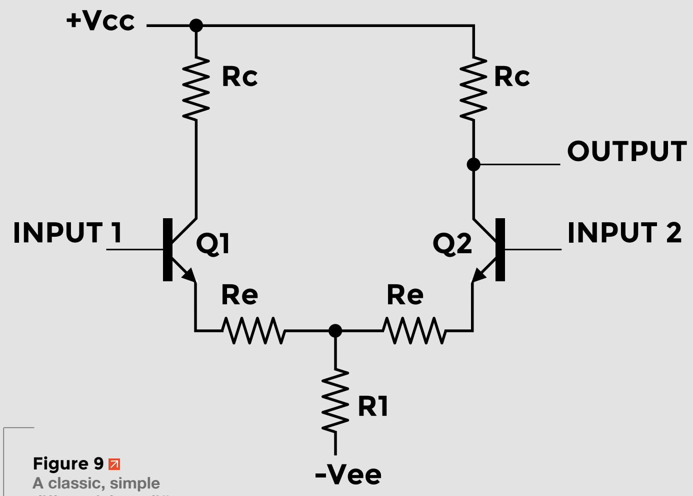


Figure 9
A classic, simple differential amplifier

cables are not unusual. The signals travelling along them are very high frequency, and not very strong. Why does this matter? Well, they are often used in noisy environments; noisy in the RF sense, not the sound sense. Remember our discussion of inductors? A varying electrical/magnetic field (which is what RF signals are) induces current in a conductor (which is what the wires in a cable are). This effect is the key to how antennas work. Unfortunately, a long cable is a very effective antenna. That means it can pick up any RF signals that are around. This presents a problem, since they can corrupt the signals travelling through the cable. This is where the twisted pair comes in. The same RF interference will be picked up by both wires. By twisting the pair of wires together, we can ensure that they both pick up the same interference. So, now we have a signal transmitted by a pair of wires; the signal of interest will be the difference between the two wires. For example, one could be a signal, and the other could be ground. Any RF interference will be picked up by both wires, so the original signal will still be the difference between them, even though the actual signal might not be recognisable due to the interference.

That's where the differential amplifier comes into play. Unlike previous amplifiers we've looked at, a differential amplifier takes two signals and amplifies the 'difference' between them. If we run the wires from a twisted pair into a differential amplifier, the output is a copy of the original signal, with anything in common (the interference) removed. **Figure 9** shows a classic, simple differential amplifier. In this circuit, the output is taken from one collector and this is the most common configuration. It takes the difference between the inputs and amplifies it, making it available at the output.

> "By twisting the pair of wires together, we can ensure that they both pick up the same interference"

QUICK TIP

A differential amplifier and twisted pair cables help deal with RF noise picked up by cables.

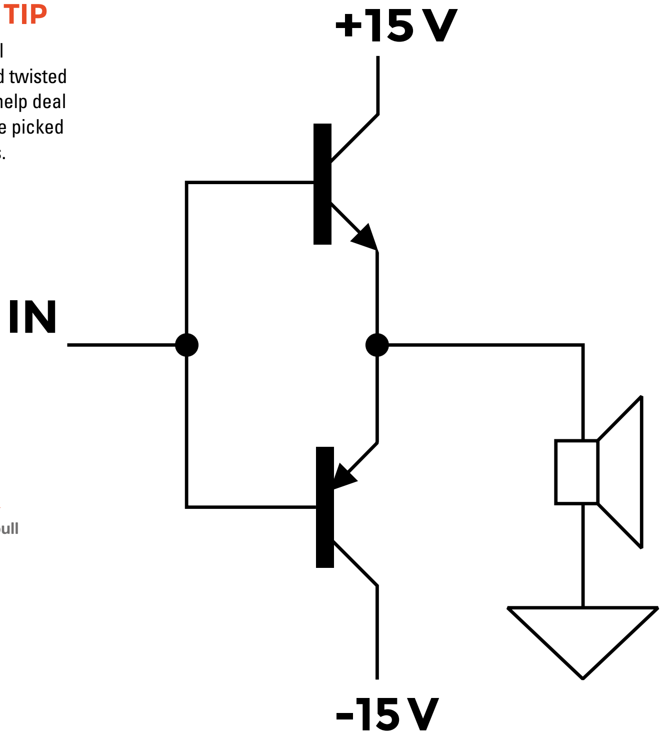


Figure 10
Basic push-pull amplifier

PUSH-PULL AMPLIFIERS

As we saw, some amplifiers aren't voltage amplifiers, but instead are current amplifiers. This is especially the case in power amps that drive speakers. Speakers work (as discussed in HackSpace #11) to make sound by moving a coil in a magnetic field. The coil moves

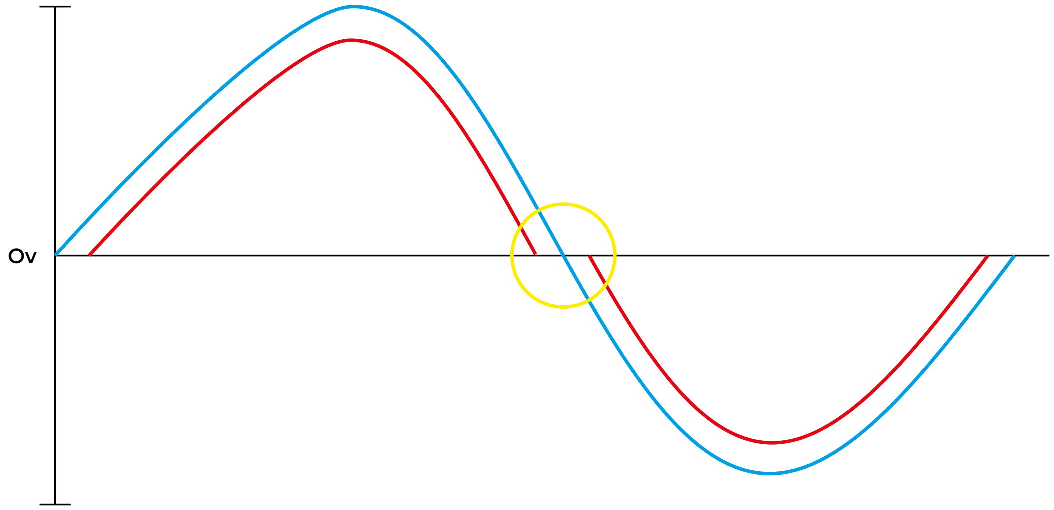


Figure 11 Input vs. output in a naive push-pull amplifier circuit

by controlling the current flowing through it. A big speaker needs a lot of current. So, amplifying the current of a signal is needed. We've seen above that a simple emitter follower needs to be biased so that the output varies around half Vcc. That means that the transistor is always conducting some amount, with a sizable current flowing through it. What happens when a lot of current flows through a transistor? It heats up. That heat is generated by power that could be used more productively. Also, heating up electronics generally causes problems, which means big heatsinks and/or fans to get rid of that waste heat.

Ideally we want transistors to conduct current only as it is required. A push-pull power amplifier circuit is a common approach to addressing this. **Figure 10** shows the basic circuit.

There's a problem, though. Remember that in the emitter follower, the output was offset by 0.6V (the drop across the base-emitter junction) from the input. The same thing happens here, but due to there being separate transistors of opposite type to handle each side of zero (an NPN for the positive part and a PNP for the negative), those base-emitter junction offsets don't line up at 0V. We get something that looks like the plot in **Figure 11**. This is because the emitter is 0.6V lower than the base in an NPN transistor, but 0.6V higher in a PNP. This is referred to as 'crossover distortion' and is generally not wanted.

This can be fixed by using a resistor and diode to bias each transistor, as shown in **Figure 12,** to keep them just at the edge of conduction. Using a diode (i.e. a P-N junction) to bias compensates for the base-emitter voltage drop of the transistor. This means that as the input signal leaves 0V, the appropriate transistor can immediately start conducting. That distortion at crossing 0V is gone.

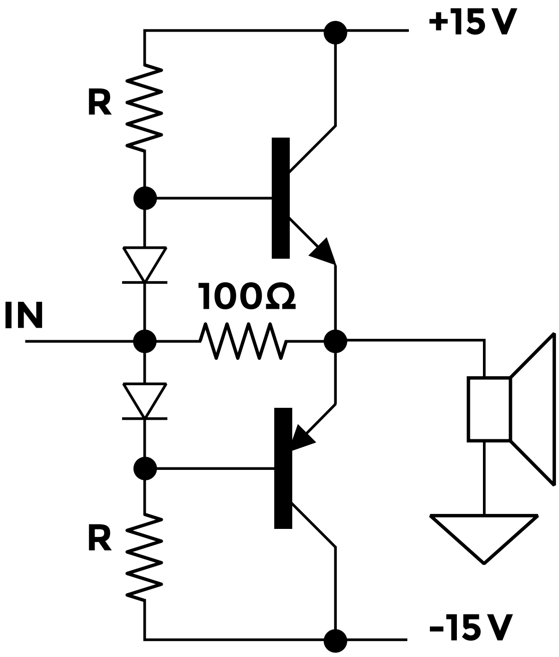


Figure 12 Diode-biased push-pull amplifier

QUICK TIP

Push-pull power amplifiers avoid wasting power as heat.

IN CONCLUSION

Amplifiers take signals and make them bigger, stronger, and more powerful. This can be by increasing voltage, current, or both. We have looked at several types of amplifiers: emitter follower, common emitter, differential, and push-pull.

As always, this is only a brief introduction. If you want more information, or more detailed theory, have a look at *The Art of Electronics* by Horowitz and Hill.

Next time, we'll move away from transistors, and will take our first look at integrated circuits. □

Make your own tools: Screwdriver

Make a knurled brass-handled screwdriver, with interchangeable 'locking' bits

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 and milling machines and CNC kit.

In the last two issues of HackSpace magazine we have looked at both milling machines and lathes, their component parts, uses, and the basic operations they perform. In this article we are going to apply what we have learnt and make a useful tool using a variety of techniques. This project uses the lathe and the milling machine, but doesn't need high accuracy and therefore no particular dimensions are given. Instead, following the principles of this build, it will hopefully enable you to make a similar tool suitable for your needs. If you read this article and can't follow some of the terms regarding lathes and milling machines, do check out the earlier articles in issues 14 and 15.

To begin, grab your piece of brass and place it into the lathe in a three-jaw chuck and tighten the jaws. For these first few operations, it doesn't matter if the chuck jaws mark the brass a little as this will get tidied up later. We want to leave a small section of the brass protruding from the chuck so that we can face the end of it, drill, and then create the short tapered section nearest the tool bits in **Figure 1**. Creating this with the work held close to the chuck means that

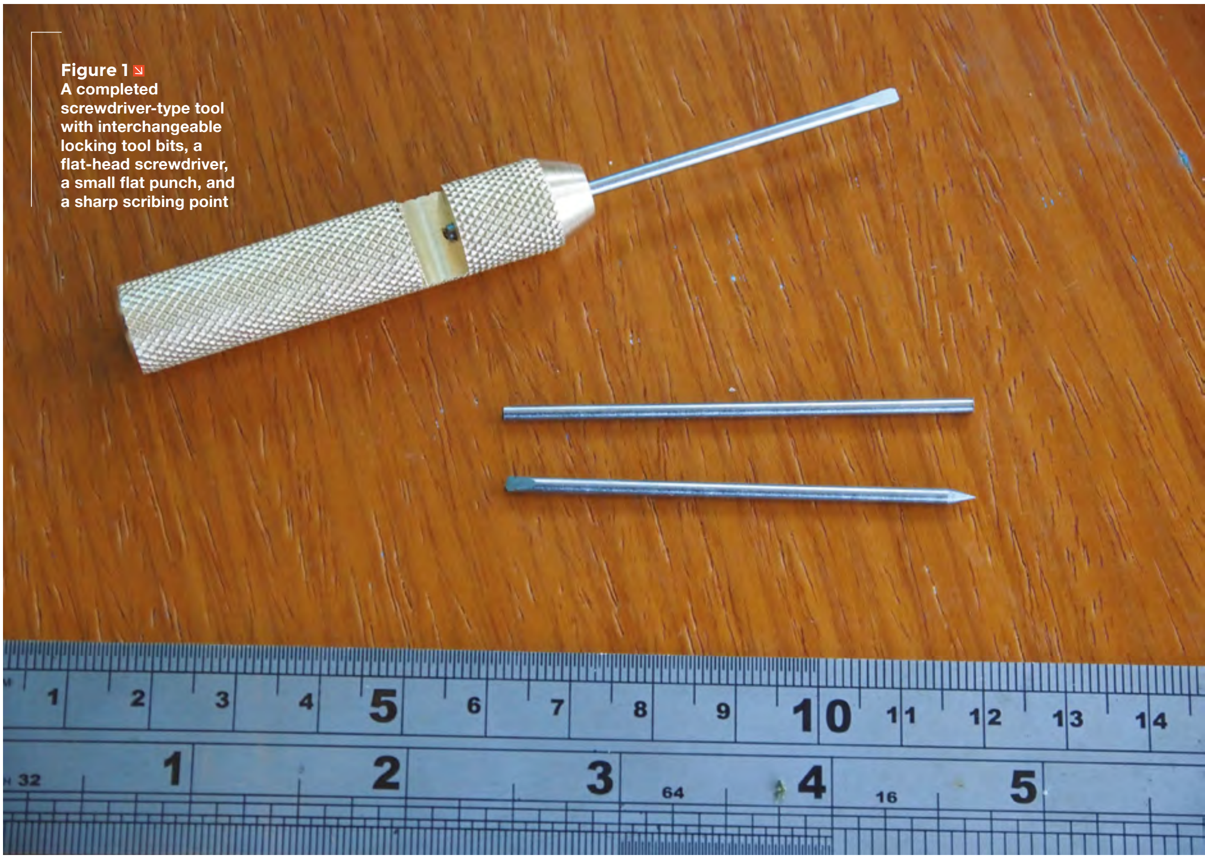
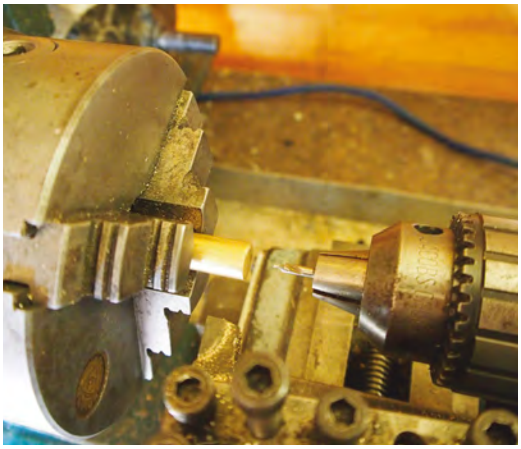


Figure 1 A completed screwdriver-type tool with interchangeable locking tool bits, a flat-head screwdriver, a small flat punch, and a sharp scribing point

YOU'LL NEED

- **Access to a lathe and milling machine** and tooling
- **Some round brass bar** around 12 mm in diameter
- **Some 2 mm silver steel rod**



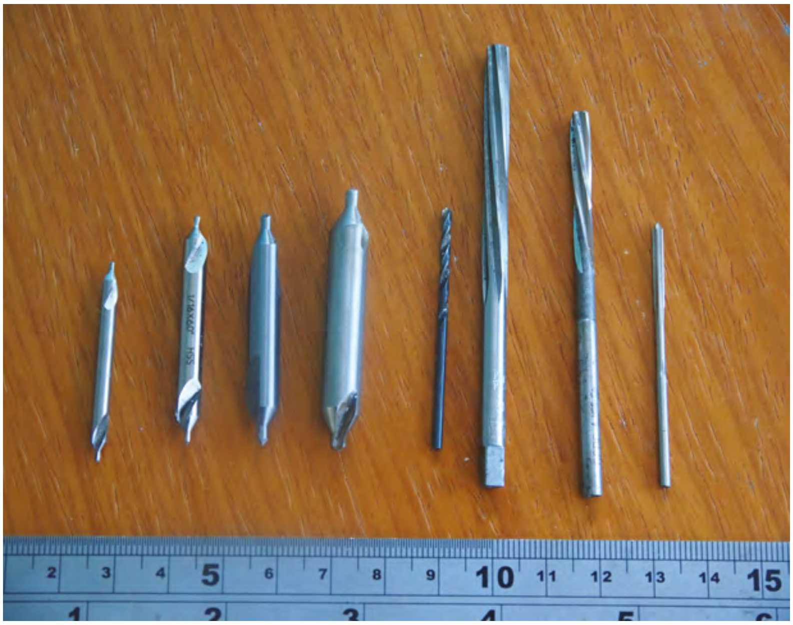
Above
Having faced the end of the brass, we are now set up with a small centre drill in the tailstock chuck

there is less chance of it deflecting or bending away from the cutting tool. Wherever possible on the lathe, you want to always work as close to the chuck as possible to mitigate this.

As this is smallish diameter brass, which is relatively easy to cut or 'free machine', we want the rotational speed of the lathe to be set high for these first few operations. First, use your right-hand cutting tool to perform a facing cut across the end of the brass to ensure that the surface is flat, then place the chuck in the tailstock quill, and fit a small centre drill to it. We want to use the smallest centre drill and only create a small 1 mm or so deep cut with it, as any deeper and the centre drill will create a hole wider than the 2 mm we are aiming for. Again, with the rotational speed high, create a small hole.

> **It's good practice, in drilling small diameter holes,** to use a small amount of cutting oil

Next, replace the centre drill with a 1.8 mm drill bit and drill slowly into the brass rod. I aimed for a depth of around 18 mm in my 12.5 mm diameter brass stock. It's good practice, when drilling small diameter holes, to use a small amount of cutting oil (or any fine oil will do) and only take small cuts, and then retract the drill bit and clean the tip with a brush to remove the chips. When you have reach the desired depth with the drill bit, replace it with a 2 mm reamer, and again move the reamer into the hole, accurately reaming the hole to 2 mm. →



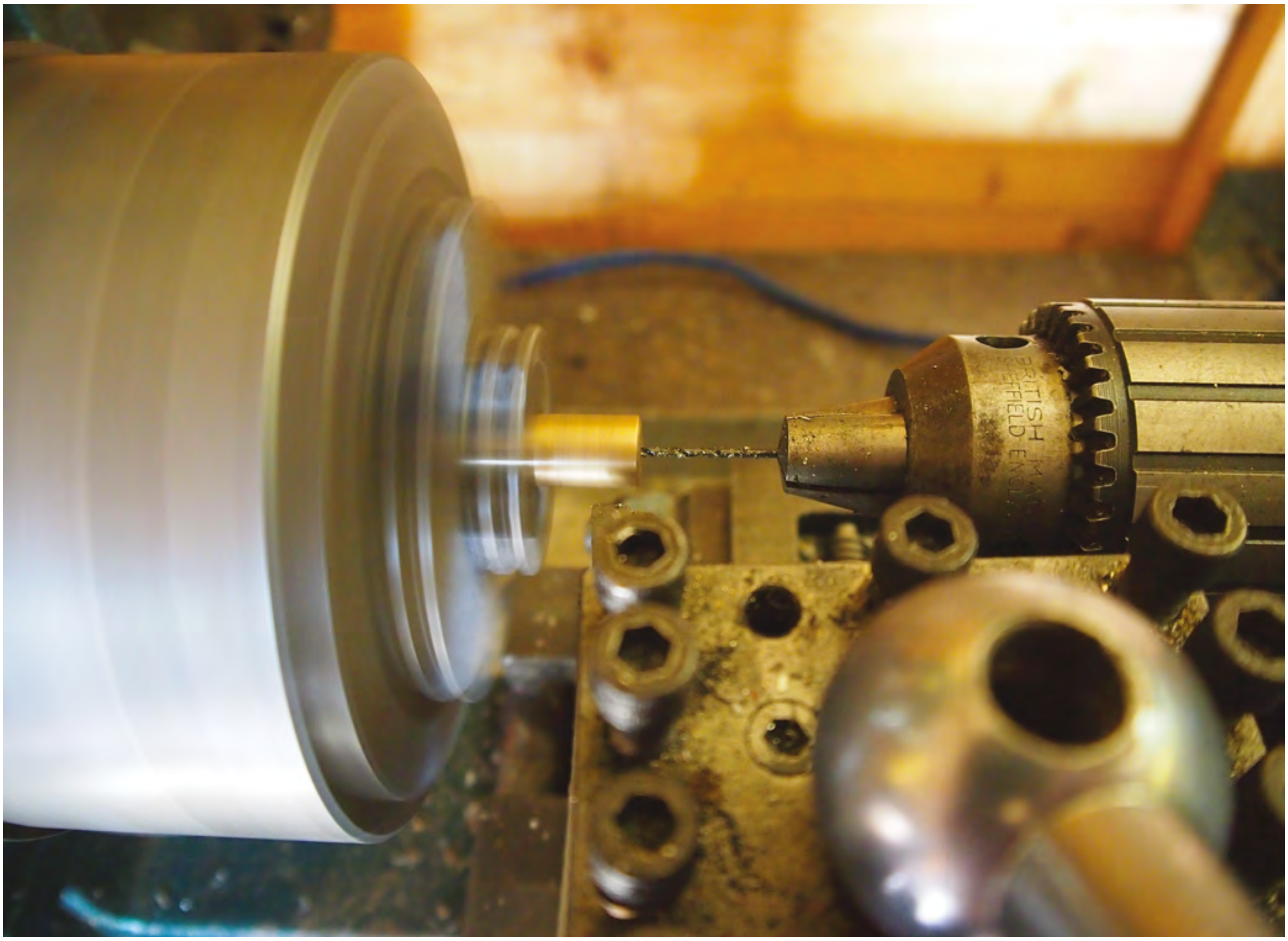
Above
Centre drills (left), regular drills (centre), and reamers (right). Centre drills come in a range of sizes, and are used in many drilling machines to start holes. They are very rigid and won't wander, so will start a hole accurately on position

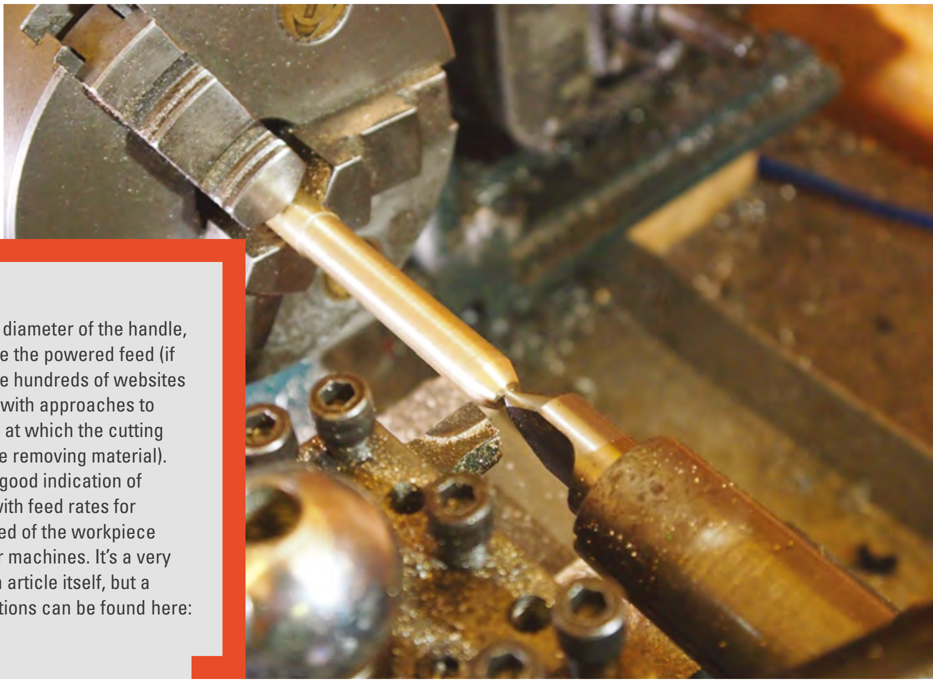
CUTTING TAPERS

Setting up to cut the tapered section involves setting the compound slide to an angle. Depending on your cutting tool geometry, you may need to rotate the toolpost for the back of the cutting tool to clear the work. Again, the angle you choose to cut the front taper of your tool is up to you. For the pictured complete tool in **Figure 1**, the compound was set to 30 degrees. If your lathe has a carriage lock, lock the carriage in position, and use the cross slide to add a depth of cut and use the compound slide to perform the cut. Make numerous passes, removing a little material each time. Again, choice of how deep you cut the taper is up to you. In the example shown in **Figure 1**, I continued until the remaining face of the tool was around 5 mm.

QUICK TIP

Let's get started! First of all, get your safety glasses on, tie up long hair and stow loose sleeves, test the machine's emergency cut-outs, and make sure everything is as safe as possible.





Above
Drilling the hole that will receive the tool bits to depth

Figure 2
Turning down the diameter of the brass handle

FEED ME

Having set up to turn down the diameter of the handle, we next need to set and engage the powered feed (if you have it available). There are hundreds of websites and online information dealing with approaches to calculating feed rates (the rate at which the cutting tool moves along the workpiece removing material).

These can give you a really good indication of where to start experimenting with feed rates for materials, and the rotation speed of the workpiece to find what works well on your machines. It's a very broad subject and worthy of an article itself, but a good example of these calculations can be found here: **hsmag.cc/nAUaOc**.

QUICK TIP

Using an undersize drill bit, and then a reamer, is a good way to achieve a really highly accurate hole diameter. If you don't have a reamer, however, you could drill with a 2 mm drill for this project.

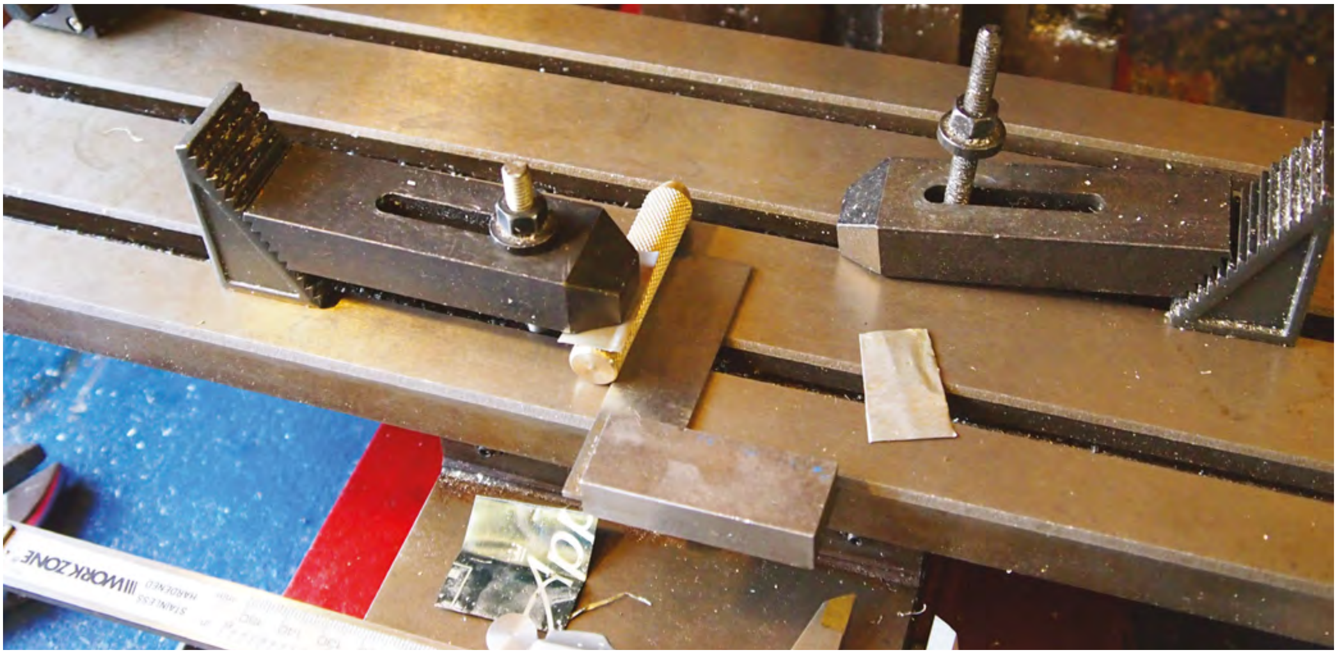
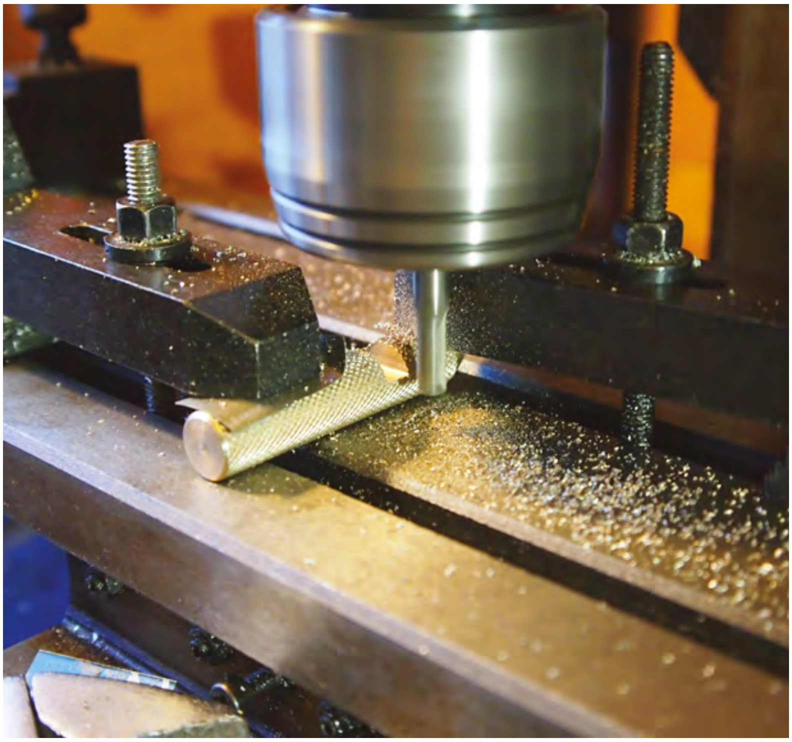


Figure 3
Using the technique of setting the workpiece square to the axis, using an engineer's square. Whilst there are more accurate techniques, this is accurate enough for this project

Below
Performing cuts across the tool handle to create the locking system

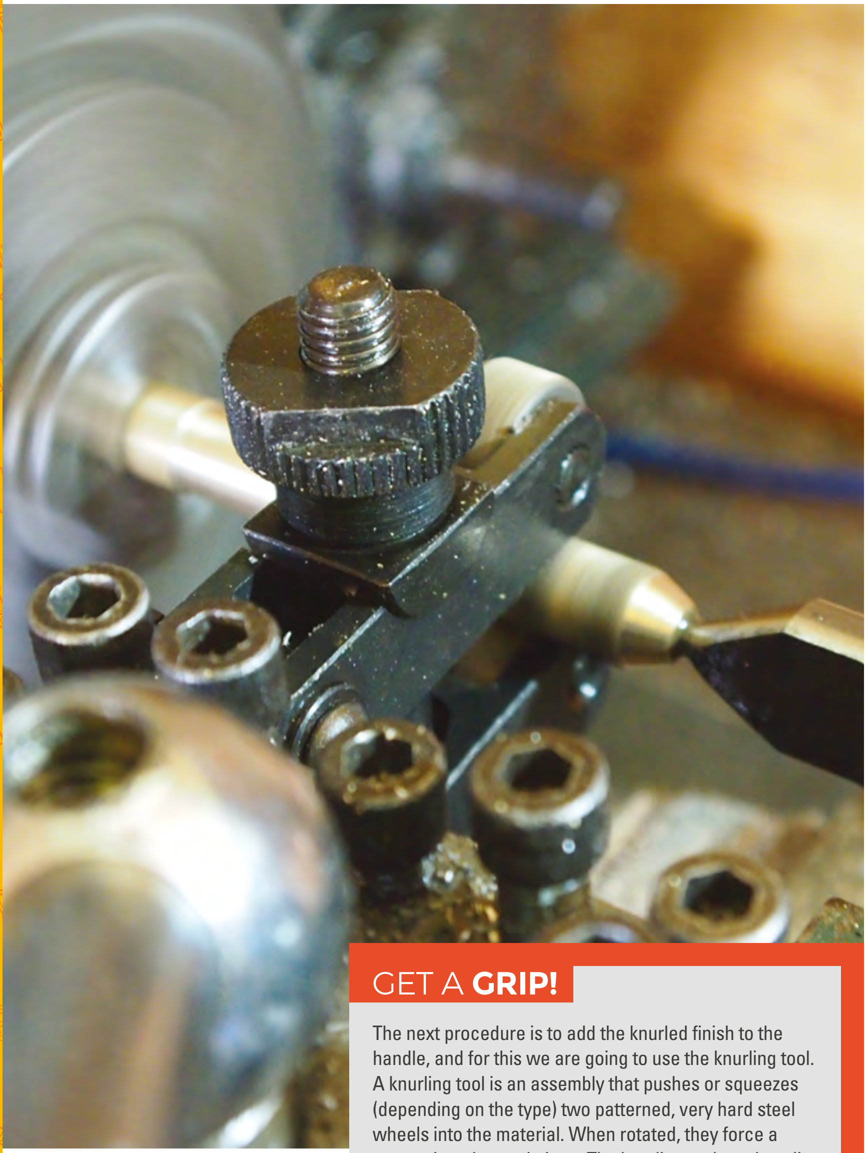


DON'T PANIC!

You may notice when you cut the taper that it isn't quite evenly distributed at the wider end where it meets the shaft. This problem occurs as it is rare for round metal bar stock to be perfectly round. We will solve this in the next operation, and make everything concentric (aligned with the centre).

Next, we are going to turn down the diameter of the stock for the length (plus a few centimetres extra for tool clearance later) of the tool handle we want to make. For around a 6 cm-long tool handle, reset the workpiece in the chuck, with around 9 cm of the bar coming out of the chuck, as in **Figure 2**. Place a dead centre in the tailstock quill, and bring it into contact with the hole in the end of the workpiece. The dead centre will help stop the, now longer, workpiece from deflecting away from the cutting tool, but we need to be careful not to fit it too firmly as it may accidentally ream/drill the accurate hole we made earlier wider. Adding a small spot of grease, or some oil to the tip, will also help avoid this. →

> "This problem occurs as it is rare for round metal bar stock to be perfectly round"



GET A GRIP!

The next procedure is to add the knurled finish to the handle, and for this we are going to use the knurling tool. A knurling tool is an assembly that pushes or squeezes (depending on the type) two patterned, very hard steel wheels into the material. When rotated, they force a pattern into the workpiece. The knurling tool used applies a squeezing force by the user drawing the two wheels together by tightening a large connecting bolt. Many prefer this type, as they distribute the forces evenly onto the workpiece, rather than a type that pushes the two wheels into one side of the work. Whilst stationary and unpowered, increase the tension on the knurling tool at the start point of the tool handle, at the edge where the tapered section meets the main diameter. Turn the lathe chuck by hand, with the lead screw feed disengaged, and inspect the workpiece until you see a good knurl pattern appearing. Using the lowest speed possible (or the lowest back gear speed if your lathe is equipped with a back gear), run the lathe under power, with the carriage travelling towards the chuck, sending the knurling tool down the length of the workpiece. Disengage the knurling tool travel before the tool hits the wider section of the bar stock, stop the lathe, and release the pressure on the knurling tool. You should then have an even, knurled finish along the piece.

QUICK TIP

The dead centre used in the tailstock, for the setup in **Figure 2**, is a 'half' dead centre, which usefully allows more access and clearance for the tool at the end of the workpiece.

Proceed to take some small cuts, turning down the diameter of the workpiece. Again, dimensions are not critical. Removing some material will make the outside face of the tool handle concentric to the hole we drilled and reamed, and the taper should now appear centred if it didn't earlier. In the example tool, the brass bar stock was around 12.5 mm, and it was turned down to just under 12 mm.

PARTING IS SUCH SWEET SORROW!

The next stage of this project is to 'part off' or remove the workpiece from the remaining stock. This can be done using a parting tool on the lathe, but this can be a tricky operation for many and in this instance, for ease, it can be readily cut to length with a hacksaw. Remove the workpiece, and cut the tool handle to the length you require, plus a millimetre or so extra.

> **"Take care not to overtighten the chuck** as that may damage the knurled finish"

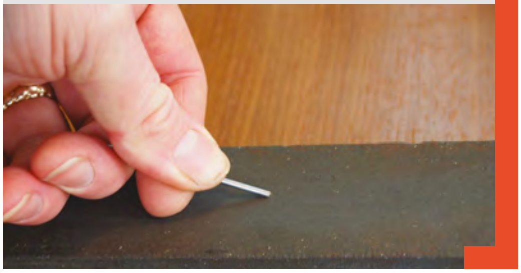
To then finish the lathe section of this project, mount the workpiece with the newly cut end facing out of the chuck, and face the end square, and then create a small angled taper on this end to make it more comfortable in the hand (see **Figure 1**), using the same method as we used earlier to create the main taper. Take care not to overtighten the chuck as that may damage the knurled finish. A good option is to place three small pieces of aluminium shim (fizzy drinks cans are a good source) in the jaws of the chuck so that the workpiece still stays centred, but is held by the aluminium.

TO THE MILLING MACHINE!

The last operation to perform on the tool handle is to mill a slot that is just over half the diameter of the handle deep, and at a position where it will intersect slightly with the end of the hole we drilled and reamed to accept the tool 'bits'. Using an engineer's square (see **Figure 3**), clamp the workpiece at each end, again using a bit of protective aluminium shim between the workpiece and clamp to avoid damaging the finish. The width of the slot is not too critical, and in the example piece a 6 mm diameter end mill was used. Perform some measurements to assess the diameter of your handle (it will have increased in diameter slightly after knurling). Similarly work out,

ITTY BITTY

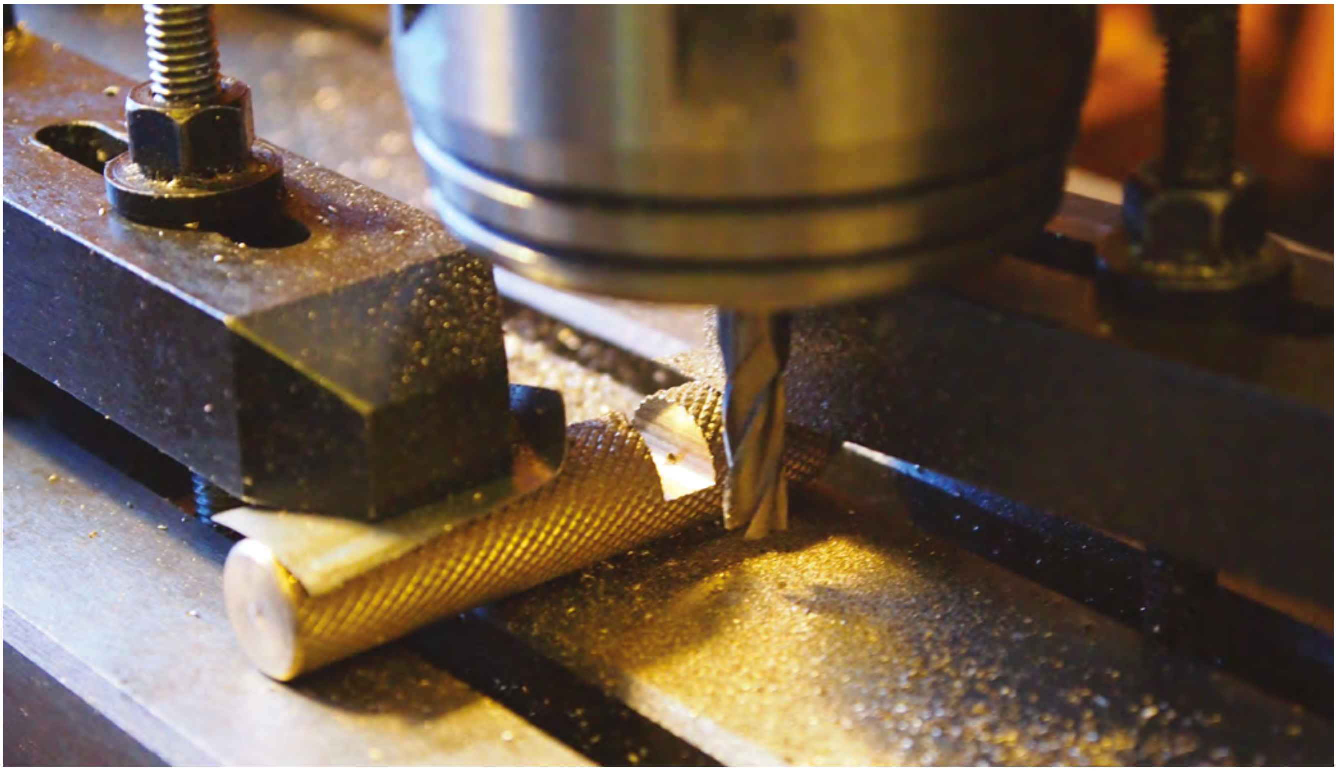
The final task is to make some tool bits. Cut some short lengths of the hard silver steel and, for the end that engages into the handle, grind an angle into the end and you should find that then when you insert the tool, it protrudes through the exposed half hole and locks the tool so it cannot rotate in the handle. You can see this angled cut on the lowest tool in **Figure 1**, at the start of this article. Grinding can be accomplished using a bench grinder, a Dremel-type tool, or even using a sharpening stone, as shown. For the other end, a simple small flat-headed screwdriver is simple to grind (take care to quench the silver steel in water after each bit of grinding, so it doesn't lose its hardness), as is a sharp-pointed bit to make the tool a useful scriber. The more adventurous perhaps can try to grind a small crosspoint screwdriver.

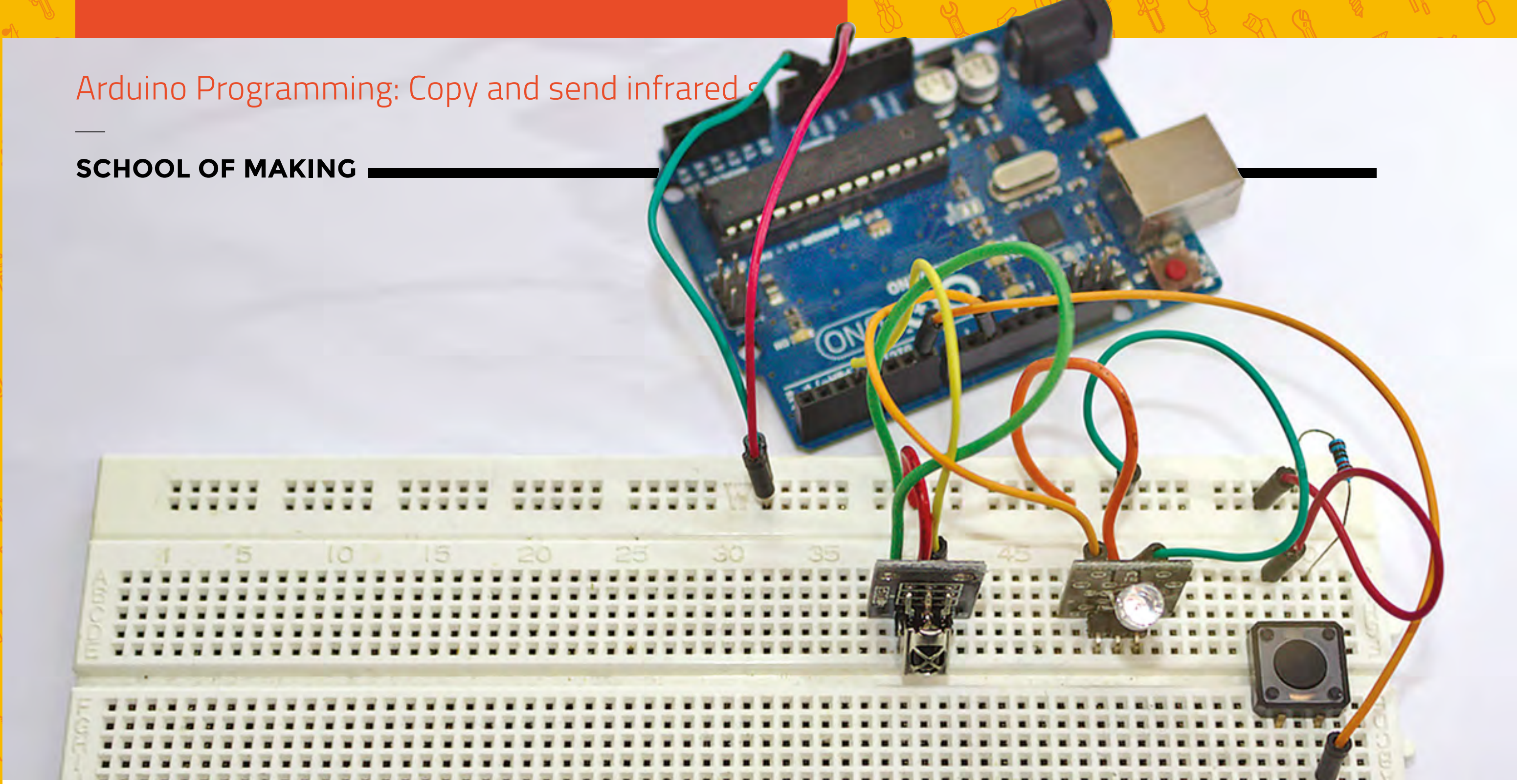


using a piece of your 2 mm silver steel and a marker, how deep your hole is. Touch on the tool to the holed end of the workpiece, and then zero the dials on the y axis and wind the tool back by the amount of the diameter of the tool, plus the depth of the hole. In the example, this was 6 mm for the tool, and 18 mm for the hole depth. To set the depth of the final cut, bring the milling cutter down to touch the surface of the milling table and then, using the graduated dials, lift it to the height you require to eventually form a slot just deeper than the midpoint of the tool. In the example, the milling cutter was lifted 5.9 mm up from the table and an end stop was clamped, so the milling head was unable to go past that depth. Then, lifting the milling head higher, perform a series of passes to cut the slot. On the Sieg SX2P, the machine would happily cut deeper, but cutting at around 1 mm per pass yielded good results.

We hope that you manage to create a tool for yourself that gives you many years of service or, at the very least, that you have learnt a bit more about simple machining approaches. If you haven't got access to a lathe or a milling machine, look up what equipment your local hackspace has, or see if a local college runs an evening class. It's a fascinating and rewarding skill set to slowly develop. □

Below
If everything has lined up correctly, your final cuts should reveal the top of the hole for the angled end of the tool bits to lock into, stopping them from rotating in the handle





Arduino programming: Copy and send infrared signals

Build a secret infrared repeater to turn off Sky Sports on the 65-inch screen in your local pub

Above While it's easy to wire everything together onto a breadboard, you might want to consider working this project into a small battery-powered container

We thought the humble infrared remote would be dead by 2019, the year of *Akira* and *The Running Man*. But infrared is still going, and has yet to be replaced by Bluetooth, WiFi, or Facebook. This means we still invariably point a piece of plastic at an invisible window on a television to change the channel. However, simplicity in this case is a good thing, because it means infrared is easy to subvert and easy to harness for your own evil projects, whether that's controlling your current equipment, or creating a line-of-sight communication channel between any of your own projects.

Despite being invisible to the human eye, infrared light couldn't be any easier to generate and to play around with. It behaves just like visible light, and can be produced with circuits no more complex than those using an LED, although infrared is usually output from a photodiode rather than a light-emitting diode. An infrared light has a light wavelength between 700 nanometres and 1 micrometre, whereas the human eye is sensitive to light between 380 and 750 nanometres, with the top of this range being red (followed by infrared). A larger wavelength means a lower frequency, which is why infrared has a lower frequency than visible light; this is why the term 'infra' – which means 'below' – is used.

We're going to create a super-flexible and generic infrared recorder and retransmitter that you can use to copy an infrared signal and resend it with many different types of generic hardware. You can use it as a single-button trigger for your own infrared commands, or at the heart of an aggregating infrared server you could use to send signals to multiple pieces of hardware from a remote source – much like Logitech's Harmony range of devices. It requires just a few components: the transmitter and receiver, a momentary button, an Arduino, and a smattering of programming, and we'll be touching slightly on both pointers and two-dimensional arrays.

HARDWARE

Even though you can get infrared LEDs to wire into your circuits, much like you would any LED, it's easier to use a pre-packaged module for both the transmitter element and the receiver. These packages are low-cost and help take some of the complexity out of the circuit, especially when it comes to decoding a signal. This is because the binary (digital) message that you send and receive with light, or even sound, needs to be modulated into a meaningful signal the analogue hardware can work with. This is what modulation does. This is where old-school modems get their name – they 'mod'ulate and 'dem'odulate signals between the digital domain of computers across the analogue (at the time) telephone network. We need much the same function to send signals by modulating signals with infrared light. The signal is modulated for sending and demodulated for receiving, with the end result being a string of binary digits appearing at one of your Arduino's pins. We then need to decode those bits into something we can understand, either by copying them and sending the same signal out on demand, or by looking for their meaning in a specification of a manufacturer's set of known infrared codes.

> **Other than the transmitter and the receiver,** we've added a simple momentary button

Both the receiver and the transmitter have three pins; two are connected to power and ground, which we connect to lines on a breadboard, and then data connections to pins on the Arduino. We've connected the receiver to pin 10 and the transmitter to pin 3. This is fixed because we're going to use a library to simplify sending and receiving signals, **IRremote.h**. This library requires the transmitter pin to be capable of pulse-width modulation, and is hard-coded to pin 3 for this purpose. The library will handle all the modulation complexity of sending and receiving infrared messages, as well as decoding them for lots of common equipment.

Other than the transmitter and the receiver, we've added a simple momentary button, the same we've already used in many projects. We're going to use this in two ways. First, by holding it down, we'll instantiate the 'receive and record' process for capturing an infrared message. And second, by pressing the button quickly, we'll send the message stored on the Arduino. Putting this into code is going to be an interesting challenge, so let's get started. →

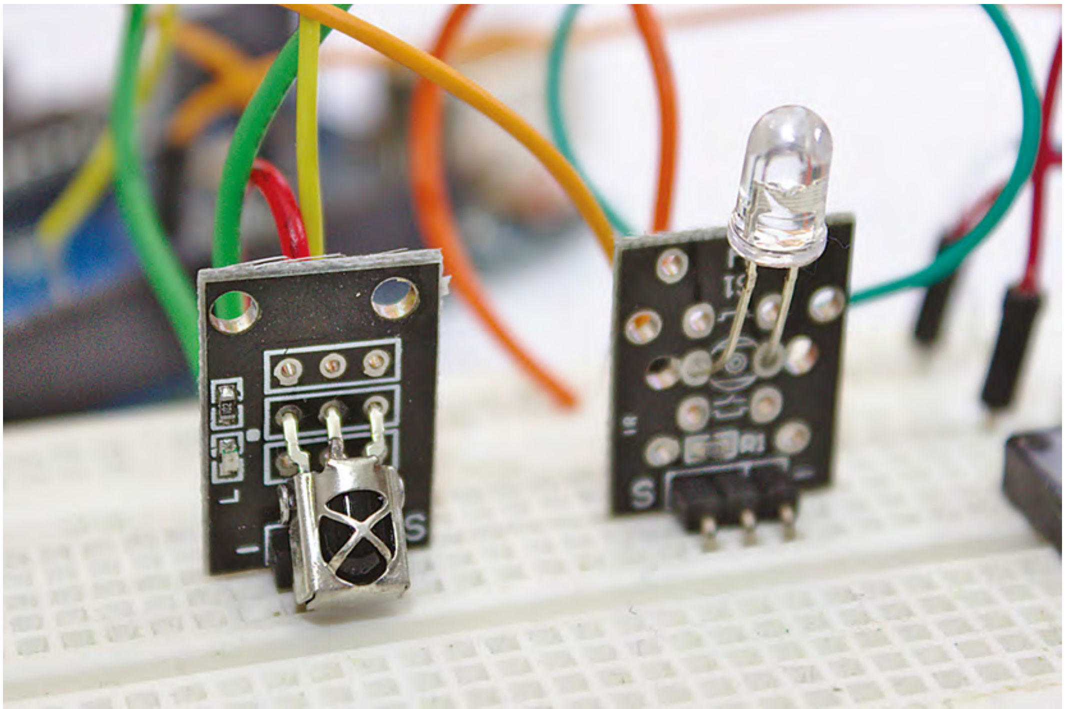
Graham Morrison

@degville

Graham is a veteran Linux journalist who is on a life-long quest to find music in the perfect arrangement of silicon.

YOU'LL NEED

- Arduino Uno
- Momentary push button
- 10 kΩ resistor
- IR receiver diode
- IR transmitter photodiode



Left
The receiver and the transmitter require only power and ground alongside a single data connection each to the Arduino

QUICK TIP

The IRremote header is located at **hsmag.cc/QJRKmw**. There's very little documentation, but there's some excellent example code if you wish to experiment.

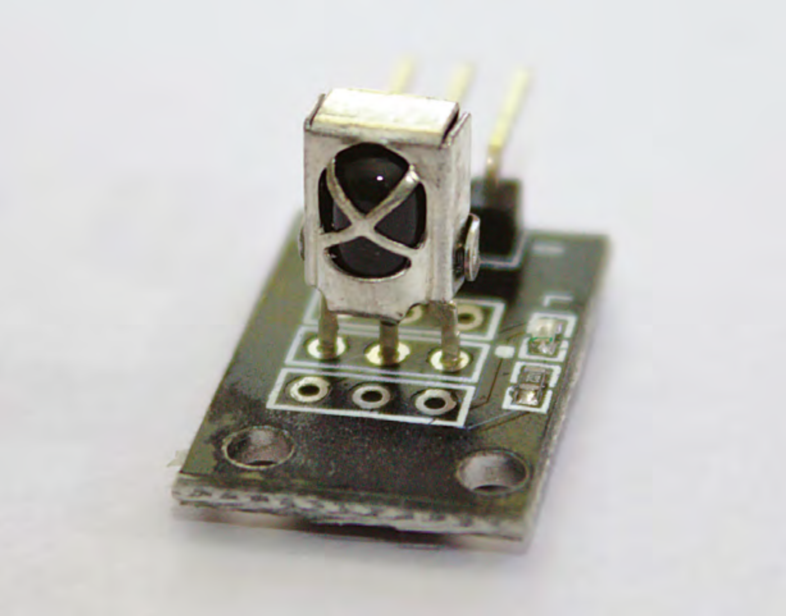
Above There are several different types of receiver and transmitter, but they're all low-cost, and mostly work in the same way as ours

Below Pre-made modules can be a bit more robust than just soldering wires to components

INFRARED LIBRARY

To a freshly created Arduino project in the IDE, the first line of code we're going to add is the header for the library we're using. We'll add this with the constant integers that hold which input pins we're using for the receiver and the button (remember, the transmitter is hard-coded to be pin 3 within the header files themselves);

```
#include <IRremote.h>
const int RECV_PIN = 10; // IR receiver input pin
const int BUTN_PIN = 7; // Button input pin
```



Don't forget that you need to first download and install any external header you use in your own project. This can be done easily from the Arduino IDE by selecting Sketch > Include Library > Manage Libraries from the menu and searching for 'irremote'. You need the package built by 'shirriff', which is nearly at the top of the search results. Click on Install in this result to install it.

We now add two sets of global variables:

```
bool buttonActive = false;
bool longPressActive = false;
int msglen = 0;
int khz = 38;
unsigned int receivedData[RAWBUF];
```

The first Boolean on/off values are going to help with the logic of the quick press/long press detection of the button. Implementing this is more complicated than it first appears, because simple momentary buttons like this suffer from jitter and false positive values during the transition from on to off and off to on. These values become true as the button press passes each state, so we know when a long press is active and can run the appropriate code.

> **Don't forget that you need to first download and install** any external header you use in your own project

The following three global values start with an integer for **msglen**. This will hold the size of a message we receive so we can make sure we store and send a message of the same length. After this is an integer we've mysteriously called **khz**. We'll be using **khz** when we transmit an infrared code because it holds the modulation frequency for the encoded data stream. The default is 38kHz, or 38,000 times a second, and is the most common frequency used by equipment manufacturers. This can obviously be changed if you need it to be. The final variable, **receivedData**, is an array of integers for the contents of the message. The size of this array is defined by a constant called **RAWBUF** which is, unusually, defined within **IRremote.h**.

We will now use three classes defined within **IRremote**:

```
IRrecv irrecv(RECV_PIN);
IRsend irsend;
decode_results results; // decode_results class
is defined in IRremote.h
```

We use one class variable for communicating with the receiver, one for communicating with the transmitter, and one for processing any results. This is a good example of using a class to hide, or abstract, the functionality of what's happening, such as the modulation and demodulation of a message. The header simply presents to the programmer an interface for controlling the hardware. We'll be using **irrecv.enableIRIn();**, for example, to initialise the receiver within the **setup()**, alongside the usual pin configuration, which is the next piece of code:

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTN_PIN, INPUT);
  irrecv.enableIRIn();
}
```

PROGRAM LOGIC

With the boilerplate code out of the way, we're now ready to tackle the logic of the code itself. First, we'll cover the logic behind detecting both short and long presses of the button. The main idea is that, as long as we know the button isn't already being pressed, we store the time when the first press is detected, and we can use this to work out whether it's been a long press or a short press. Here's the beginning of the code that detects when the long press is active (**longPressActive = true;**):

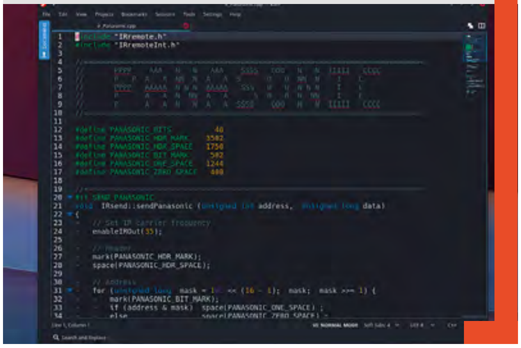
```
if (digitalRead(BUTN_PIN) == HIGH) {
    if (buttonActive == false) {
      buttonActive = true;
      buttonTimer = millis();
    }
    if ((millis() - buttonTimer > longPressTime)
&& (longPressActive == false)) {
      longPressActive = true;
    }
  } else {  // EXECUTED ON RELEASE
```

We can only detect a short press when the button is released, as it's only then we'll know the duration of the press. This is why the release code comes after the **else** statement above, indicating that →

REMOTE CONTROL **DECODING**

We've used a simple raw mode for recording and playing back infrared signals, but IRrecord can also decode and send signals to specific hardware manufacturers, which is needed when they're using their own protocols, or using a non-standard carrier frequency. It can do this because it has a large library of protocols from common manufacturers, including Sony, JVC, Panasonic, and even Lego. If you look at the header file for each manufacturer, you also get hints on how to best communicate with the equipment.

Using a function from each library, you can extract the command code from the manufacturer-specific code used as a container for the commands. This means you can theoretically control elements like volume or channel numbers using variables, combining them with the code you know controls volume when you send them from the Arduino. You could even chain commands together for different pieces of equipment, enabling you to set up a home audio/video configuration for a movie, or for music, for example, and then control playback from something that can then talk to the Arduino.



```
#include "IRremote.h"
#include "IRremoteInt.h"

//==============================================================
//      PPPP   AAA   N   N   AAA   SSSS   OOO   N   N  IIIII   CCCC
//      P   P A   A  NN  N  A   A  S     O   O  NN  N    I    C
//      PPPP  AAAAA  N N N  AAAAA   SSS  O   O  N N N    I    C
//      P     A   A  N  NN  A   A     S  O   O  N  NN    I    C
//      P     A   A  N   N  A   A  SSSS   OOO   N   N  IIIII   CCCC
//==============================================================

#define PANASONIC_BITS          48
#define PANASONIC_HDR_MARK    3502
#define PANASONIC_HDR_SPACE   1750
#define PANASONIC_BIT_MARK     502
#define PANASONIC_ONE_SPACE   1244
#define PANASONIC_ZERO_SPACE   400

//+=============================================================
#if SEND_PANASONIC
void  IRsend::sendPanasonic (unsigned int address,  unsigned long data)
{
	// Set IR carrier frequency
	enableIROut(35);

	// Header
	mark(PANASONIC_HDR_MARK);
	space(PANASONIC_HDR_SPACE);

	// Address
	for (unsigned long  mask = 1UL << (16 - 1);  mask;  mask >>= 1) {
		mark(PANASONIC_BIT_MARK);
		if (address & mask)  space(PANASONIC_ONE_SPACE) ;
		else                 space(PANASONIC_ZERO_SPACE) ;
```

QUICK TIP

Encoding signals by varying the width of a pulse (Pulse Width Modulation) is exactly how 'PWM' audio synthesis works, and also how some audio communication protocols work.

Left ⬖ Using one of the manufacturer-specific profiles in the library can help solve compatibility issues. Panasonic, for instance, uses a carrier frequency of 35 the button event is not setting the button to HIGH. We then reset the **longPressActive** variable if this event has already been detected as a long press. And if not, after another **else**, we finally get to play with some infrared code:

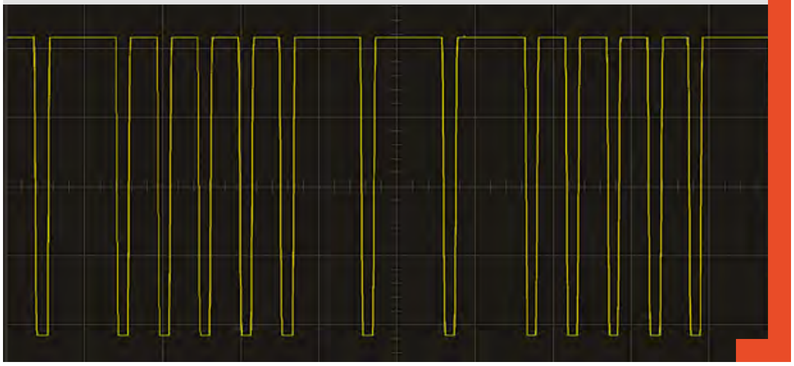
```
 if (buttonActive == true) {
   if (longPressActive == true) {
    longPressActive = false;
   } else {
    if (msglen > 0) {
     irsend.sendRaw(receivedData, msglen,
khz);
     delay(50);
     irrecv.enableIRIn();
    }
   }
   buttonActive = false;
  }
```

INFRARED AND OSCILLOSCOPES

You can use an oscilloscope with an infrared receiver to calculate the frequency of the signal from an infrared source.

Despite being invisible to the naked eye, there are several ways to see an infrared signal (outside of joining the SAS and snagging some infrared night-vision goggles). The simplest is to use your smartphone camera. Looking at the real-time preview whilst pressing buttons on your remote should reveal flashes from the infrared LED. The front camera is usually best, as it's less likely to have an infrared filter because infrared is often used to help with motion detection and facial recognition. But if you happen to have an oscilloscope handy, you can study an infrared signal in much more detail. From the circuit we've created, just attach one of your oscilloscope probes to the output of the receiver – the same output that connected to pin 10 on the Arduino.

When you now fire a few infrared signals at the receiver, you should notice your oscilloscope come to life. In particular, if you set your update resolution to around 2 ms, you should see square waveforms with different widths. These changing widths are key to how different values are 'modulated' into the signal transmitted by the infrared LED. Notice, for example, that the 'off' width is always the same. This is the pause between transmissions, and it's constant. The actual data is carried in the variable widths of the on-time – hence, pulse-width modulation. Different widths hide different values, decoded automatically by any infrared receiver.

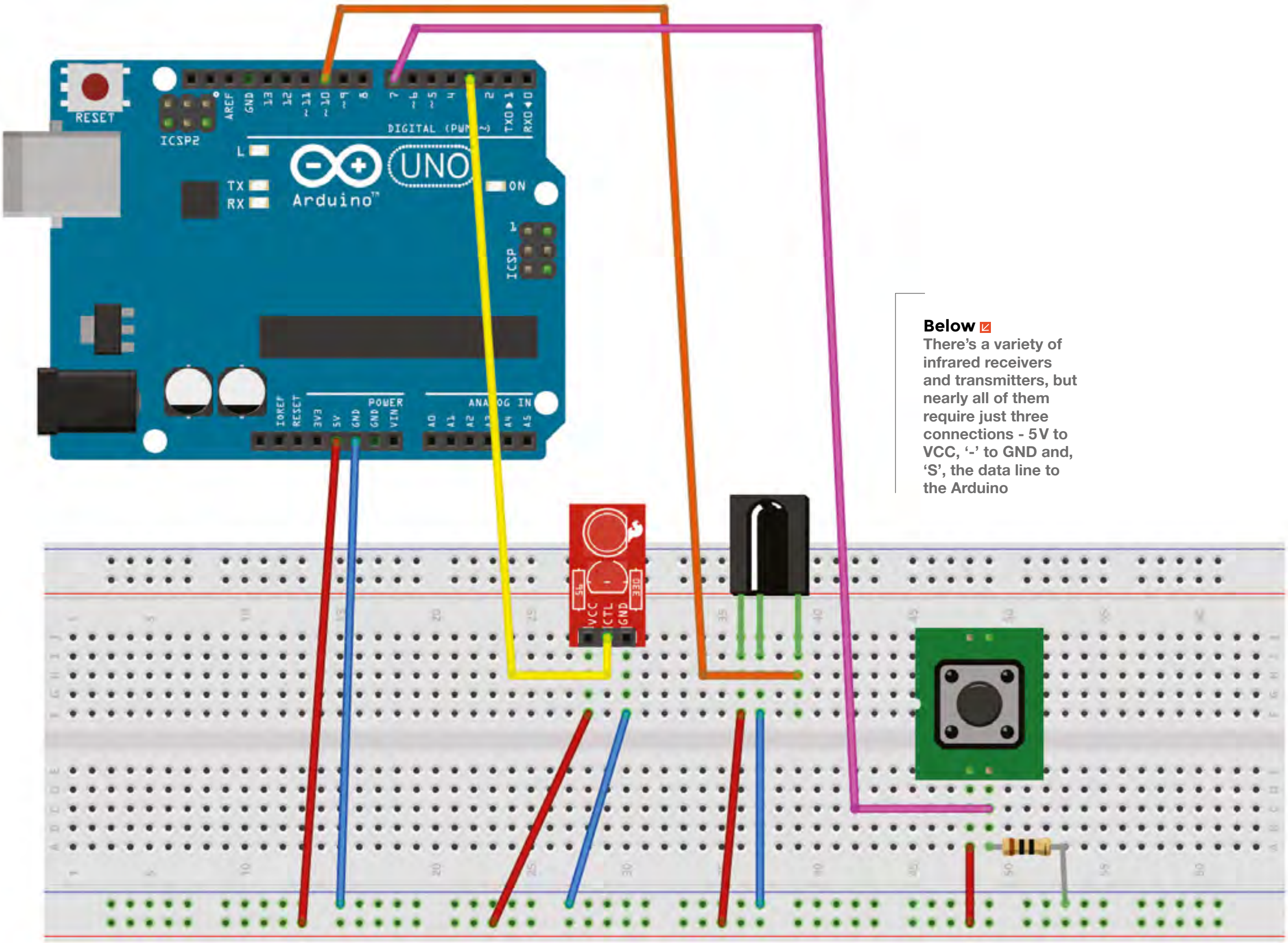


Thanks to **IRremote.h**, sending a signal is super-simple. First, we check to make sure there is a recorded message (**mesglen > 0**), and then send the message with **irsend.sendRaw(receivedData, msglen, khz);**. The data we transmit is in **receivedData**, but you may notice something. We created this variable as an array, but we're not including any brackets, or targeting a specific element. This is called 'passing by reference', rather than the more common 'call by value'. This works because it's only a reference to the array that's being passed to the **sendRaw** function, and this reference is really the memory address where the first element of the array is being stored. The relative address of each element can then be calculated by generating an offset from the amount of space required to store an element of the array's type. If this sounds exactly like what a pointer does, you're right – by not including an element identifier, we're implicitly using the array variable name as a pointer.

The final piece of code is also the most functional because it's responsible for receiving the data and decoding it into something we can use. This code runs outside of all the previous code because this improves the response time of our program when an infrared signal is received. This is how we start this block of code, quickly followed by a check to see whether the **longPressActive** Boolean is true. If so, this means the button is being held down and we can go ahead and record the infrared signal being received. If not, we can ignore the signal until next time.

> **We can only detect a short press when the button is released,** as it's only then we'll know the duration

```
if (irrecv.decode(&results)) {
  if (longPressActive) {
   msglen = results.rawlen - 1;
   for (int i = 1; i <= msglen; i++) {
    if (i % 2) {
     receivedData[i - 1] = results.rawbuf[i]
     * USECPERTICK - MARK_EXCESS;
    }
    else {
     // Space
     receivedData[i - 1] = results.rawbuf[i]
     * USECPERTICK + MARK_EXCESS;
```

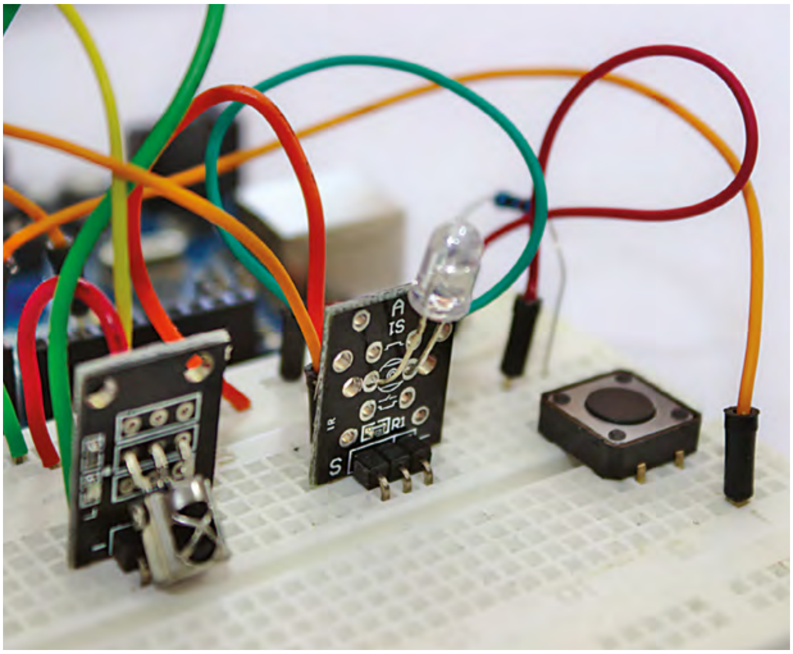


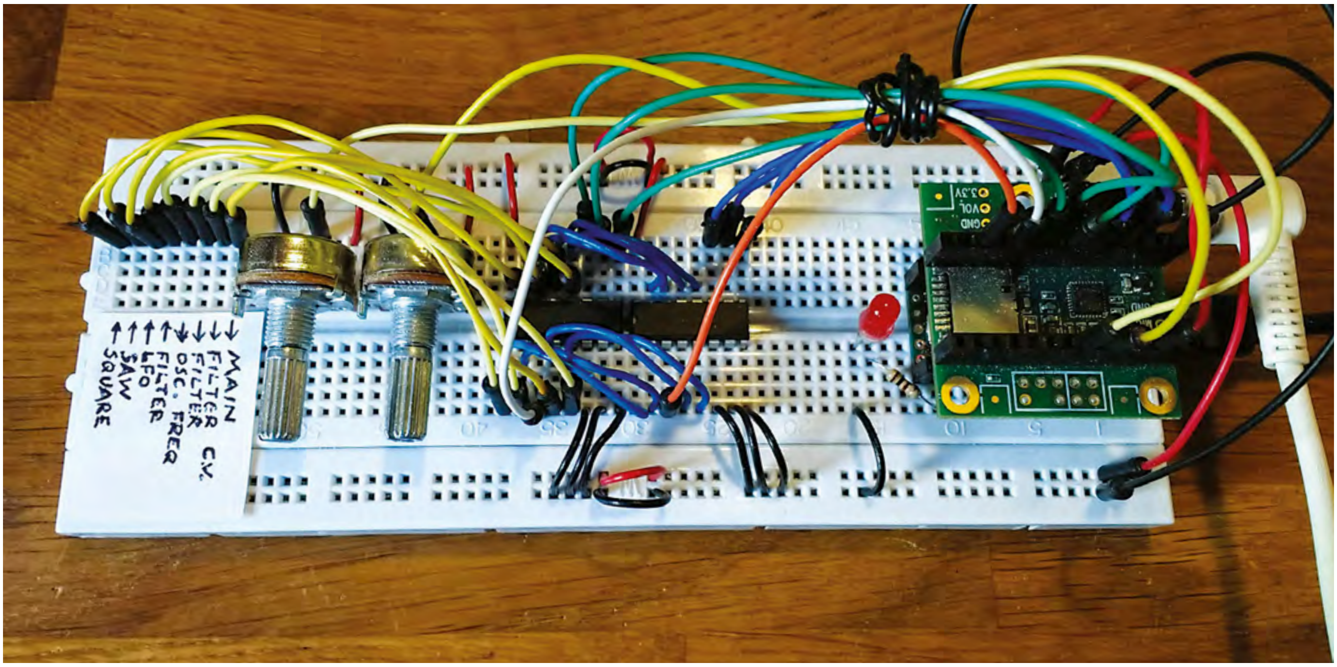
Below
There's a variety of infrared receivers and transmitters, but nearly all of them require just three connections - 5V to VCC, '-' to GND and, 'S', the data line to the Arduino

```
    }
   }
  }
  irrecv.resume(); // resume receiver
 }
} // End bracket for project
```

The decoding code itself is taken from the IRremote library, using **`%2`** to work out whether we're receiving either an odd or even element for the array and then using this to tweak the gaps between the received elements to cancel out receiver distortion. The results are placed into the array and the size of the message stored in **`msglen`**, which is the message we can now send whenever we perform a short press on the button. And that's all there is to it. Build it and send it to your Arduino! □

Right
Everything set up and running on a breadboard, but you might want to use protoboard to make this more permanent





Polyphonic digital synthesizer Part one

Build a full-featured polyphonic digital synthesizer in our two-part guide

Matt Bradshaw

mattbradshawdesign.com

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

Analogue synthesizers have made a big comeback in the last few years, but building a synth that can play multiple notes (i.e. chords) using only analogue circuitry is a big challenge. In this tutorial, you will see how to build a versatile synthesizer with a 'patchable' signal chain, but where the sound is generated digitally by code that you can write yourself. This is a two-part tutorial, but even by the end of part one you'll already be able to make some great sounds.

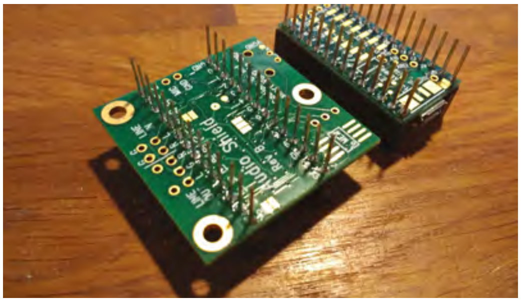
GOING DIGITAL

Modular synthesizers are awesome. They let you create your own signal chain by plugging cables into different points in the circuit, giving you the freedom to create any sound you can imagine. A true modular synth is basically a box which you populate with individual modules that you can either buy or build (see the excellent tutorial from HackSpace magazine issue 14 for an example). Some modules generate signals, while other modules take a signal and change it in some way.

This tutorial will show you how to create a miniature, digital version of an analogue modular synthesizer. The process of 'patching' different signals into each other will be done on the breadboard with jumper wires, and that information will be processed by the Teensy microcontroller.

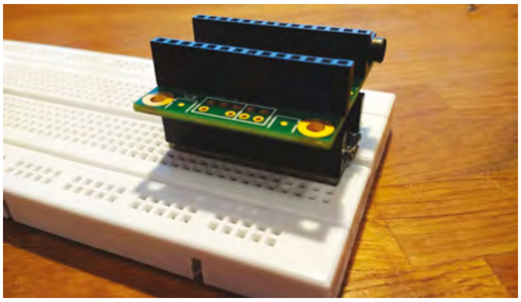
Firstly, we need to set up our Teensy 3.2, which is a bit like an Arduino but powerful enough to process audio. When you buy a Teensy, it usually comes

Above This design combines aspects of digital and analogue synthesizers, to give you a versatile, but cheap-to-make, instrument



Above

The extra-long headers are soldered underneath the Teensy and the audio board



Above

Here's how the Teensy and audio board should look on the breadboard – make sure that the pin numbers line up

without any headers, so you'll have to do a bit of soldering. The audio board, which sits either directly above or below the Teensy, also requires soldering. Using stackable headers is a good idea (female headers with long male pins on the other side), as these will make both the Teensy and the audio board compatible with a breadboard.

Once your Teensy is ready, download the Teensyduino software from **hsmag.cc/aRWmgD**, and try running an example audio sketch, such as File > Examples > Audio > Synthesis > PlaySynthMusic. You should then be able to hear music through the headphones jack.

> "The Teensy has its own library of code for adding audio to projects"

START SMALL

Before we can connect lots of modules together, we should try a simple sketch to get the hang of writing code to produce audio. The Teensy has its own library of code for adding audio to projects, and it works a lot like a modular synthesizer.

For instance, in the **sine_wave** example sketch, an oscillator is connected to an output via two `AudioConnection` instances (one for each stereo channel), meaning that a sine wave is heard through the headphones. Download this sketch from **hsmag.cc/issue16**, and try it for yourself.

Our synth will consist of eight sockets, and we will need to know which sockets are connected to each other. For instance, if the oscillator socket is connected to the main output socket, the Teensy needs to be able to read this and then recreate the connection digitally, producing audio. On a 'real' synth, these sockets would be sturdy 3.5 mm connectors (basically headphone jacks), but for this synth we are simply going to use a row of breadboard sockets. For now, it doesn't really matter which socket corresponds to which input or output – we just want to know whether socket A is connected to socket B, and so on.

ONE THING AT A TIME

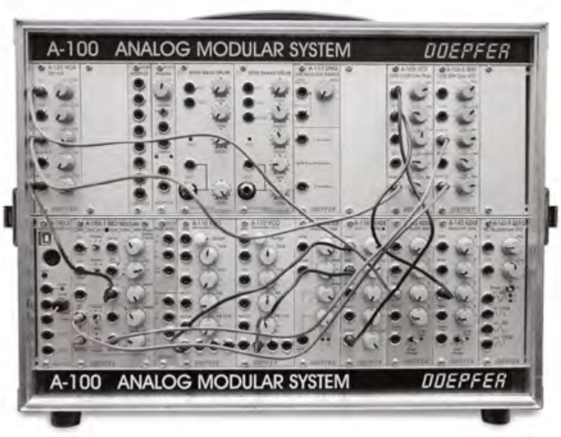
In order to test the connections between the sockets, we will use an integrated circuit called the 4051. This is an eight-channel multiplexer or demultiplexer; in layman's terms, eight 'things' are connected to the chip, and you can talk to them →

AVERTING THE SPAGHETTI

This synth, particularly once you complete part two, will involve a lot of wires in a relatively small space. If you use lots of standard-length jumper wires, you will quickly end up with an unmaintainable rat's nest (albeit a very pretty one). To alleviate this problem, it's worth making a batch of your own tiny jumper wires from single core wire, maybe 4 cm long each, with about 5 mm of insulation stripped away at each end.

YOU'LL NEED

- Teensy 3
- Teensy audio adapter board
- 4 × 14-pin stackable male/female headers (2 kits)
- 2 × breadboards
- Jumper wires
- 2 × rotary potentiometers (10 kΩ, linear)
- 5 × 4051 multiplexer chips
- 8 × tactile buttons
- LED
- 6N139 optocoupler chip
- MIDI socket
- Capacitor (0.1 μF)
- Resistors (various)
- USB micro cable
- Soldering equipment
- Headphones
- Computer

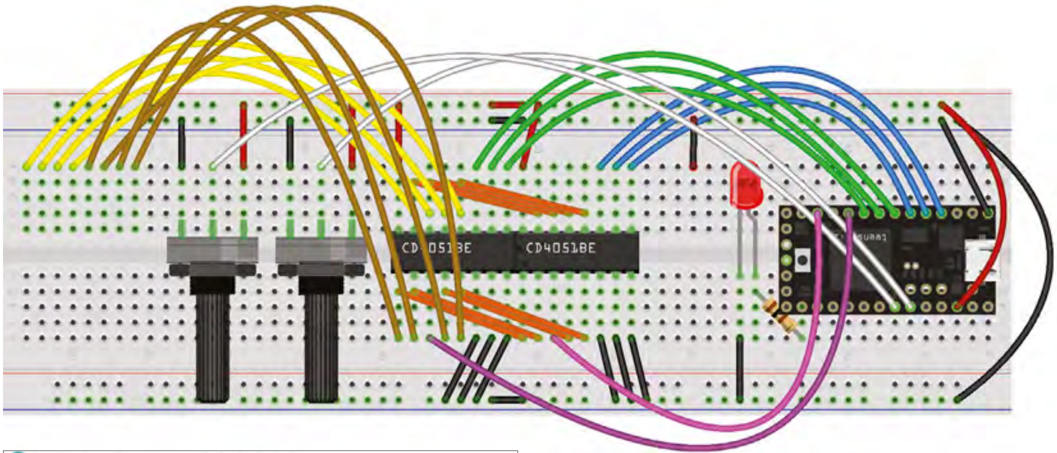


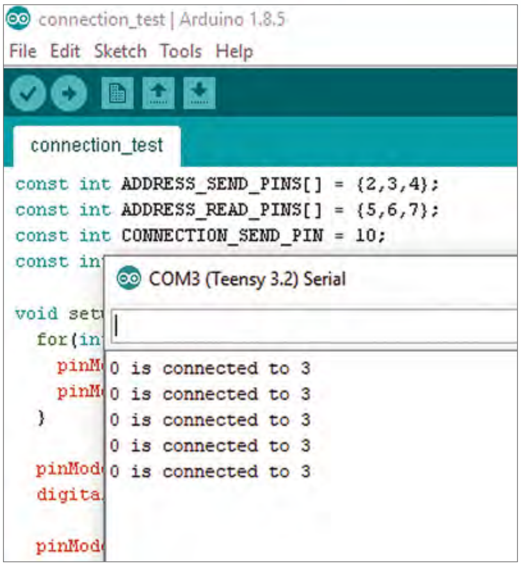
Left

This is what a full modular synth looks like, with removable modules and patch cables

Figure 1 ◇
The full breadboard layout, with the audio board omitted for clarity. Make sure to connect the channels of the two 4051 chips (see orange wires)

Figure 2 ↘
In the 'connection_test' sketch, you can check that your circuit is working correctly





```
const int ADDRESS_SEND_PINS[] = {2,3,4};
const int ADDRESS_READ_PINS[] = {5,6,7};
const int CONNECTION_SEND_PIN = 10;
const in

void set
  for(in
    pinM
    pinM
  }

  pinMod
  digita

  pinMod
```

```
0 is connected to 3
0 is connected to 3
0 is connected to 3
0 is connected to 3
0 is connected to 3
```

one-at-a-time. Three pins are used to select which 'thing' you want to talk to (these are connected to the Teensy), and eight pins are connected to the 'things' (in our case, the sockets).

Start by building the breadboard circuit, as shown in **Figure 1**. Notice that there are two 4051 chips, both addressed separately, but with their channels connected to common sockets. By using two 4051 chips in this way, you can send a test signal to each channel in turn on the first chip, then listen for that signal on each channel in turn on the second chip. If a signal is sent to channel A on the first chip, for example, and can be read on channel B of the second chip, socket A must be connected to socket B.

To try this out, download the **connection_test** sketch from **hsmag.cc/issue16**, and open it in the Arduino IDE. You will see a nested **for** loop, with the outer loop addressing the 'send' chip and the inner loop addressing the 'read' chip.

```
for(int a=0;a<8;a++) {
    setSendChannel(a);
    for(int b=0;b<8;b++) {
    setReadChannel(b);
    delayMicroseconds(10);
    if(a < b) {
      boolean connectionReading =
!digitalRead(CONNECTION_READ_PIN);
      if(connectionReading) {
        Serial.print(a);
        Serial.print(" is connected to ");
        Serial.print(b);
        Serial.print("\n"); }}}}
```

Upload the whole sketch to the Teensy and open the serial monitor. Now try connecting two of the sockets on the left end of the breadboard with a jumper wire. If everything is working, the serial monitor should report that a connection has been detected (see **Figure 2**), and we're ready to move onto the actual synth code.

THE INS AND OUTS

The 4051 gives us a maximum of eight sockets to use, which are allocated as follows:

- Oscillator output #1 (square wave)
- Oscillator output #2 (sawtooth wave)
- Oscillator frequency modulation input
- Low-frequency oscillator output
- Filter input
- Filter modulation input
- Filter output
- Main output stage

These sockets are worth explaining in a bit more detail, especially if you're not that familiar with synthesizers. The two oscillator outputs are simply tones with slightly different sounds (the sawtooth is a bit more 'buzzy'). The oscillator modulation input changes the pitch of the oscillator, meaning that when you connect the low-frequency oscillator

QUICK TIP

Once you've got your Teensy and audio board set up, search the web for 'Teensy synth' for more inspiration on what to make.

CHEAPSKATE **VERSION**

This project is a pretty cheap way into building your own synthesizer, but if you're willing to put in a bit more effort you could make it even cheaper. The audio board used in this tutorial is great, but there are less expensive alternatives. The PT8211 audio chip will give you 16-bit audio output for very little money, if you don't mind some very delicate soldering. Alternatively, you can get a lower-quality audio output direct from the Teensy via its DAC pin. Note that both of these options will require minor changes to the code.

(LFO) into it, you'll hear a tone that rises and falls like an ambulance siren. The filter is an effect which restricts certain frequencies while boosting others, and can also be modulated by the LFO. Finally, the main output stage represents the final part of the signal chain – you won't hear anything until you plug something into it. Don't worry if you don't understand all the ins and outs – once you start playing around with the synth, it should all start to make sense.

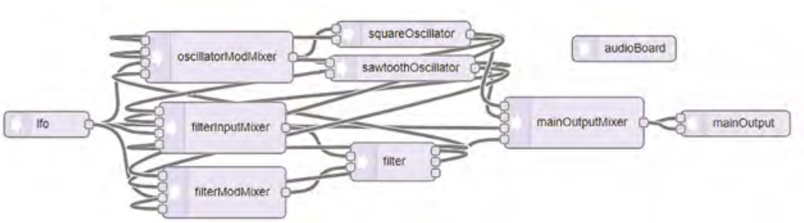
MAKE SOME NOISE

The easiest way to start writing audio code for a Teensy is to use the online 'Audio System Design Tool' at: **hsmag.cc/OiKbYH**. It's a simple drag-and-drop interface for connecting audio modules together, and it's definitely worth getting familiar with. For this synth, however, you can just copy and paste the code directly to make things a bit easier. Download the main sketch code from: **hsmag.cc/issue16**.

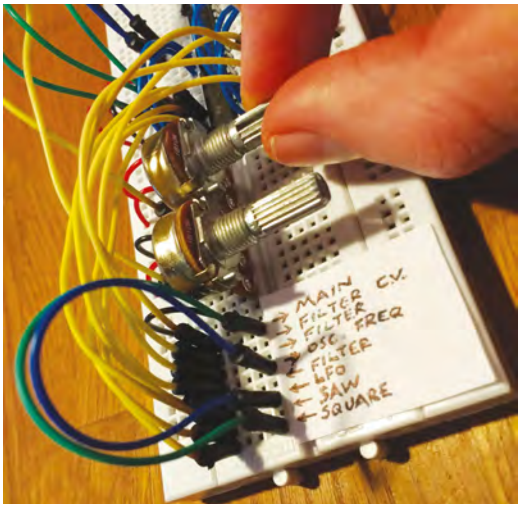
It's a good idea to look through the code to understand what's going on. The sketch basically combines the two simpler sketches from earlier, and adds a few extra features. It begins by declaring the various audio objects and how they are connected – this code was generated in the online design tool. Next, we declare an array of references to the four input mixer objects, so that we can easily reference them by number later on.

In the `setup()` function, we initialise the various input and output pins, and set some initial parameters for our audio objects – feel free to tweak these numbers to produce different sounds. The `loop` function works much the same way as in the earlier example sketch, but instead of sending a serial message when a connection is made (or broken), the volume of a relevant mixer channel is set to either one (for a connection) or zero (for no connection).

At the end of the loop, the LED is lit if a bad connection (input-to-input or output-to-output) is



Above This is what the synth looks like in the online Teensy Audio System Design Tool – the lines represent possible audio connections



Left Playing with a simple but fun patch where the oscillator frequency is modulated by the LFO

detected. Unlike on an analogue synth, making bad connections won't cause any harm in this design, but it's useful to know. Finally, the two potentiometers' values are read and used to control the LFO frequency and main oscillator frequency. Feel free to change this section of the code to customise your synth, by making the knobs control other parameters.

The last job is to label the patching area on the left of the breadboard. Either use a fine pen, or print a label from your computer in a small font, and use Blu Tack or tape to affix the label to the breadboard. Now you can start playing!

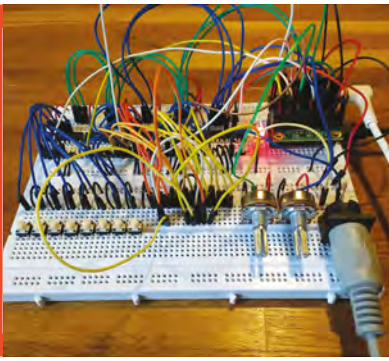
Try connecting different outputs to different inputs and see what happens. Turn the knobs up and down to control the sound. The synth is capable of dirty bass drones and *Doctor Who*-esque effects, but if you want to get really musical, you'll have to wait for part two! □

QUICK TIP

If this synth has piqued your interest, try the free, open-source software 'VCV Rack', which is a virtual modular synth.

NEXT **TIME**

In the second and final part of this tutorial, we'll be adding some features to really turn this project into a usable synthesizer. We'll be adding a second breadboard with a simple keyboard (allowing you to play melodies) and a MIDI input (allowing you to control the synth from another keyboard or a computer). We'll also double the number of connections you can make, and use a cunning trick to add polyphony to the synth, meaning you can play more complex music.



Below A sneak peek of what the synth will look like after part two, including a mini keyboard and MIDI input

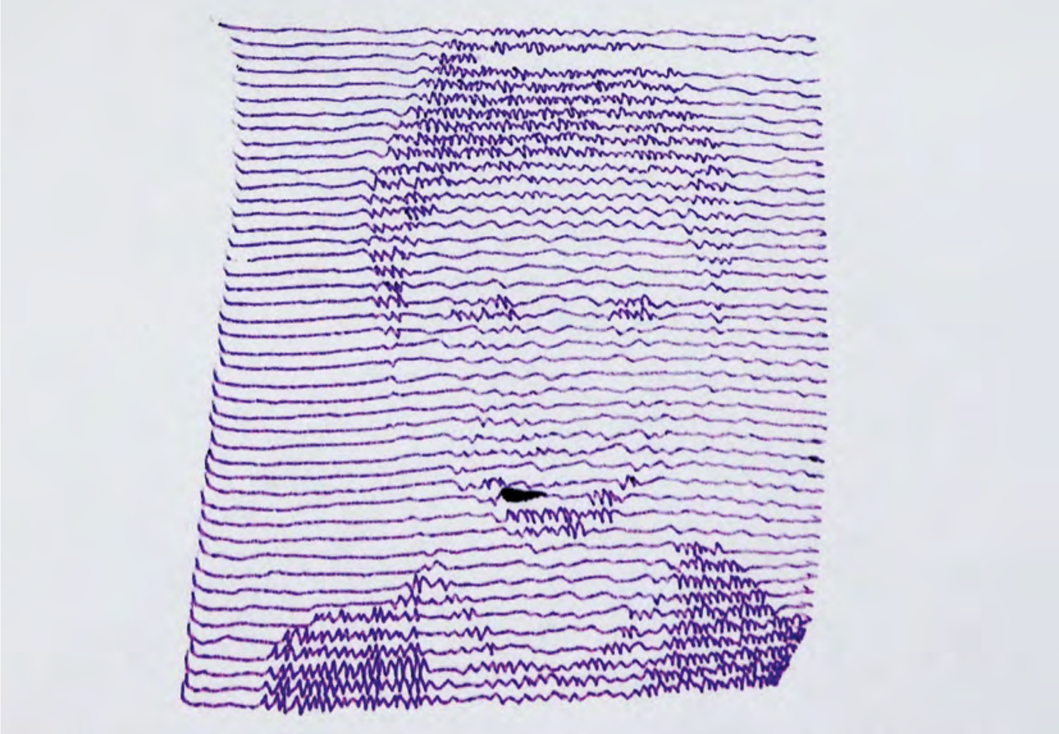


Figure 1 If you're planning a ghost movie involving polygraphs, this is how you can get your beings to appear

Plotter art

Take mechanical pen to paper and let your computer draw

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.

Plotters are devices for moving pens (or other drawing implements) around on paper. They come in many forms, including classic X-Y plotters (such as the EleksDraw that we review on page 124), and the one we'll be using in this tutorial – the Line-us.

Although we'll be using a Line-us, the same basic techniques should work on most plotters, but you'll have to use different software to control your machine.

We're going to look at two techniques for turning photographs into drawings. You can't directly plot a photograph because it's a raster image – that is, it's made up of a grid of pixels, each of which has a particular colour. Instead, you need an image made up of lines. There's no right way of doing this, and it all comes down to this fuzzy thing they call 'art'. We're going to look at two bits of software for converting images to line drawings, but first, we need to manipulate the image we start with.

We can fit far less detail in a line image than we can in a photo, so we need to make sure we're keeping things simple. We started with a selfie, but we took it against a plain background so there's no extraneous detail in there, then we used our phone's adjustment settings to make our face stand out as much as possible. This involved tweaking the contrast, light, shadows, and highlights settings in a highly unscientific manner until the face was as dark as possible, and the background as light as possible.

The two ways we'll manipulate this into lines are with the SquiggleCam web app and the Stipple Generator. First, let's look at the SquiggleCam. This draws your image in a series of horizontal lines. Each line shows the same frequency wave, but the darker the image at a particular point, the higher the amplitude of the wave (see **Figure 1**). Point your browser to **hsmag.cc/yLgCjQ** and it should load. You can take an image with your webcam, but we found that it worked much better if you adjusted the image as described above.

DRAWING SVGS ON THE LINE-US

The Line-us doesn't have in-built support for SVGs, but GitHub user, ixd-hof, has made a processing application that can send commands directly to the Line-us from SVG files. You can download this for Windows or Mac from: **hsmag.cc/qYQoXA**. Unzip the file, and run the **Line_Us_SVG.EXE** file. This will open a window with the commands to plot your SVG. Pressing A will let you enter the appropriate address for your Line-us (depending on your network, this will either be line-us.local or the IP address of the plotter). Pressing C will connect to your plotter (and the line will go green if the connection is successful). O lets you choose an SVG to plot. It doesn't automatically scale or centre it, so you need to use +/- and arrow keys to move it to the right place. You'll get more accurate plotting at the bottom of the available range than at the top.

If you load the image and the window stays blank, try zooming out a long way, as it could be in the wrong place.

Once you're ready to plot, hit P, and let your Line-us do its thing.

There's a range of things you can change in the Squiggle Settings. The two that had the biggest effect for us were the Line Count, and the Amplitude. The Line Count is the number of horizontal lines. More lines obviously makes for a slower plot, but we found that once we went below about 40, the image was no longer very clear.

Don't be afraid to be heavy-handed with the amplitude setting. The preview image on the SquiggleCam uses a thick line which makes it look more dramatic than it is, and there's a bit of wobble in the Line-us, so the amplitude is dampened a bit when drawn. An amplitude of between 1 and 2 worked best for us.

Once you've got your image how you want it, hit Download SVG, and see box above for how to plot it.

HEAD FULL OF BUBBLES

The second option is the Stipple Generator. This was created by Evil Mad Scientist Labs for their AxiDraw plotter, but it will work with other plotters on account of its outputting SVG files. It converts images into circles using some pretty clever mathematics. The technical details of how it works are detailed on its web page (**hsmag.cc/qqMQjv**), but if you'd rather just get stuck in, you can download the software at: **hsmag.cc/Mqcbss**. The interface is fairly self-explanatory. The main configuration option is the number of circles. The default is 2000, but this will take a long time to plot. It's worth dialling this down, at least while you're experimenting.

While the SVG will include perfect circles, this is beyond the Line-us's ability to plot, so instead you get small squiggles. We quite like this effect. The Line-us doesn't manage to make different sized squiggles very well, so it's best to keep the maximum size quite small, and increase the number of circles until you get the effect you want. If you're using a more accurate plotter, you might choose other options.

> "Anything that produces a line-based SVG **should work, so get tweaking**"

We've just looked at two options for generating plotter files, but there's far more than this. Anything that produces a line-based SVG should work, so get tweaking, playing with graphics tools, and see what you can find to make your own images unique. Show us what you've done on Twitter **@HackSpaceMag**, and don't forget to include #plottertwitter □

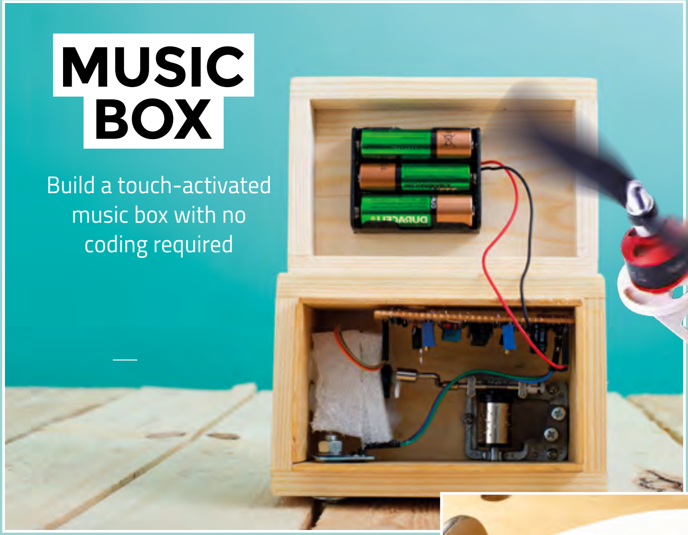
Below Like glitch art and music, the inaccuracies in the Line-us are part of its charm

50 PROJECTS FOR MAKERS & HACKERS
BOOK OF MAKING
VOLUME 1
BUILD A TREBUCHET
MAKE MUSIC
FLY YOUR OWN CUSTOM DRONE
BUILD AN A.I. WATER CANNON
CONTROL YOUR HOME: SMART TECH YOUR WAY
& LOTS MORE
FROM THE MAKERS OF HackS
BOOK OF MAKING VOLUME 1

THE BEST PROJECTS FROM

HACKSPACE MAGAZINE

THE ULTIMATE SKILLS, TRICKS, AND MAKES



MUSIC BOX

Build a touch-activated music box with no coding required

BUILD A DRONE

The ultimate guide to making your own quadcopter

LASER-CUT TURNTABLE

Create stunning 360° animated GIFs with this geared turntable

Add lights to a cosplay helmet

Colour-changing LEDs for special effects

So, you've got a cool cosplay helmet and now you want to put lights inside it? Let's do it! Whether you're starting with a purchased helmet or you've made your helmet from scratch, adding lighting can take your cosplay to the next level. There are many ways to add lighting to cosplay projects; sometimes a simple battery-powered tea light is enough to add a touch of magic to a costume piece or prop. In this tutorial, we'll take a cue from today's space movies (think Ridley Scott's *Prometheus* and *The Martian*) and add internal LEDs to light the face of our astronaut. We'll also add external LEDs that flash when our astronaut is talking: a cool interactive effect that will really bring the costume to life!

Note: adding interior lights to a helmet creates a reflective glare on the inside of the visor, especially in darker environments. Be careful when wearing this cosplay project: turn the lighting off, or remove the helmet, when moving around. Never install interior lighting in any helmet intended for safety.

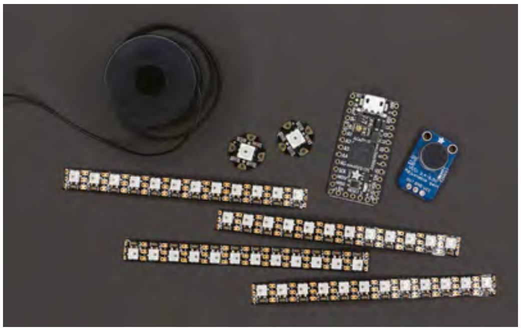
We'll work with two kinds of addressable RGB LEDs in this project: a high-density NeoPixel strip, and individual NeoPixels. The strip will be used for our internal lighting; we'll trim it into four pieces to distribute light evenly around the face. The individual NeoPixels will be our external lights, just one on each side of the helmet. We'll turn these into lights that are sound-reactive, and add a microphone on the inside of the helmet to detect when our astronaut is speaking.

Below
Rumours that this is Elon Musk's prototype helmet are entirely untrue

In this intermediate project, we'll do some light soldering, create a simple program with CircuitPython, and even craft custom diffusers for our external lights. To control the circuit, we're using the ItsyBitsy M0 Express from Adafruit. While the Gemma M0 board is our go-to for wearable projects (and would work just fine if you've got one to hand), the ItsyBitsy gives us room to grow if we want to add more functionality to the helmet in the future. Maybe a voice changer? Or sci-fi sound samples triggered by capacitive touch areas on the helmet? The possibilities are endless, and once you've got your ItsyBitsy installed, it's easy to reprogram with CircuitPython whenever you get a new idea.

For powering wearable projects, we often use battery packs or small LiPo batteries. But what if you'll be wearing your helmet all day at Comic Con and you don't want to be constantly changing batteries? For this project, our power source is a USB power bank. You may already have one for recharging your phone when you're out and about. Check the power stats, usually printed on the power bank – 5 V/1 A will do. The more mAh, the longer you can go before recharging. Find one with an on/off switch, slip it into the pocket of your spacesuit, and you're good to go all day at the con!

Because we'll be illuminating the inside of the helmet, anything that reflects light will be easily visible. Consider painting the inside of your helmet black, and using all black wires for your circuit. If you choose this method, be sure to label all of your wires as you work, and double-check every connection to avoid errors.



Above
Lots of lights ready for our helmet

For this project, your author started with a purchased costume helmet and modified it with craft foam and paint to create a custom look. The base helmet used here cost about $40 on **amazon.com**. We're focusing on the electronics here, but for techniques and tips on making cosplay helmets, check out these epic cosplayers:

Punished Props Academy
punishedprops.com
Kamui Cosplay
kamuicosplay.com/startingwithcosplay
Evil Ted Smith
eviltedsmith.com

DESIGN AND MEASURE

If possible, remove the visor of your helmet and any pieces that will be in the way during installation. Refer to the circuit diagram (not to scale) for an overview of how everything will be connected. Then, have a look inside the helmet and decide where the internal lighting will be installed. We are using the skinniest →

Right
Each helmet is different, so you'll have to measure for yourself

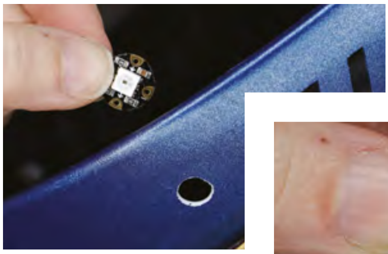
Sophy Wong

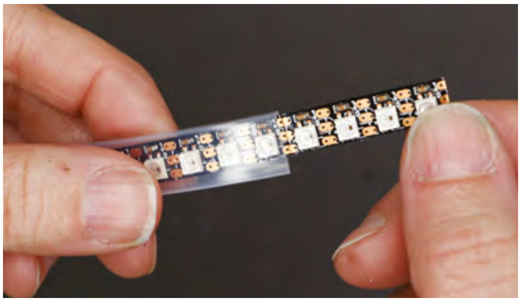
@sophywong

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**

YOU'LL NEED

- **Cosplay helmet**
- **1 metre × NeoPixel Mini Skinny Strips, 144 LED per metre** (Adafruit part 2970)
- **2 × Flora RGB Smart NeoPixel version 2** (Adafruit part 1260)
- **1 × Electret microphone amplifier** (Adafruit part 1063)
- **1 × ItsyBitsy M0 microcontroller** (Adafruit part 3727)
- **Silicone-coated wire, in black** (Adafruit part 2003)
- **Black acrylic paint**
- **Hot glue and glue gun**
- **Mould-making putty** (optional)
- **USB power bank and micro USB cable**
- **Soldering tools and supplies**





Right Individual NeoPixels are easy to mount in holes

Below The LED strip should come out of the weather-proofing

NeoPixel strips available, so you'll only need about 1 cm of flat space to mount them to. Ideally, you'll have two pieces of NeoPixel strip at the top of the face opening, and two more at the bottom of the face opening. You'll also need to find a good spot for the ItsyBitsy microcontroller, somewhere flat and accessible at the back of the helmet. Remember that our power source will be a USB power bank, so there will be a USB cable running from the ItsyBitsy M0 out of the helmet. Lastly, find a place for the microphone on the inside of the helmet, somewhere near where your mouth will be.

Measure the opening of your helmet and determine how long your strips need to be, and how many NeoPixels you'll be using. (Each piece of our NeoPixel strip has twelve pixels, for a total of 48.) Take measurements for the distances between each piece of strip so you know how long your wires will need to be. Also, measure the distance from each component to where the ItsyBitsy M0 will be located. When cutting your wires, add one or two centimetres for insurance.

Next, find a good spot on either side of the helmet for the individual NeoPixels. These will be daisy-chained together, so try to find symmetrical locations that can be connected along the inside of the helmet. You may need to cut holes into your helmet, as we did, for the LEDs to show through.

Below Soldering wires to strip, but try to make sure you label them!

If the inside of your helmet is white or a light colour, it will reflect the internal lighting, making the electronic components and modifications more visible. For a more sleek look, paint the inside of your helmet black, with a few layers of acrylic paint.

MAKE THE INTERNAL LIGHTS

Using your measurements, cut the NeoPixel strips for your helmet. Remove the silicone weather stripping, as you may want to save it for another project. It is important to note the direction of the small arrow printed on the back of the strip. This is the direction that data flows through the strip. All strips connected together must be oriented in the same direction so that the 'data out' of each strip connects to the 'data in' of the next.

It can be difficult to keep track of the direction of the strips once you've cut them down. Mark the direction of the arrow on the back of each strip. A silver marker works well for this.

Using the measurements from your layout, solder wires to your strips, connecting them together, pin by pin: power to power, ground to ground, and data out to data in. Double-check the orientation of each strip before soldering.

Cut wires long enough to reach from the beginning of the strip to the ItsyBitsy, and solder them to the beginning of the strip. Use masking tape to label each wire. There will be a lot of wires going to the ItsyBitsy by the time we're done soldering, and labels can mean the difference between success and hours of troubleshooting.

When finished, set the bundle aside and work on the single NeoPixels.

PREPARE THE EXTERNAL LIGHTS AND MICROPHONE

Again, reference your measurements and prepare wires to connect the two individual NeoPixels together. Remember that the power and ground pins of the first pixel will need two wires each: one wire to reach forward to the other pixel, and one wire to reach backward to the ItsyBitsy. Make sure the wires are long enough to be routed discreetly between each pixel on the inside of your helmet. Also, remember to connect 'data out' on the first pixel to 'data in' on the second pixel.

Add a wire for 'data in' on the first NeoPixel and label the wires that will connect to the ItsyBitsy with masking tape.

Cut three wires for the microphone according to your measurements and solder them to the microphone. Again, label each wire.

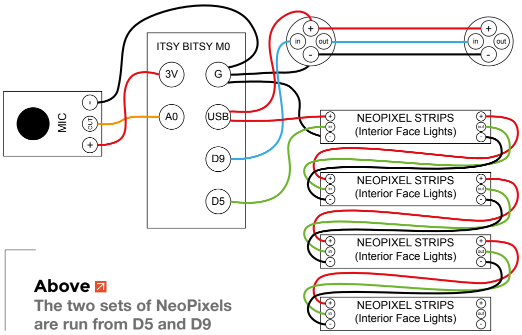
CONNECT TO THE ITSYBITSY

With all the components prepared with wires, it's time to connect everything to the ItsyBitsy, according to the circuit diagram. This is where all those labels pay off! Leave all the labels on while you work. If you need to troubleshoot any connections later, you'll be happy that everything is still labelled.

Because several components will share the power and ground pins, let's start with those. Connect the positive wire from both the NeoPixel strip and the external lights to the USB pin on the ItsyBitsy. You can twist the wires together gently and solder them to the pin at the same time. Do the same to connect the ground wires from the NeoPixel strip, the external lights, and the microphone to the G pin on the ItsyBitsy.

To power the microphone, connect its VCC wire to the 3 V pin on the ItsyBitsy. This will add less noise to the signal than the higher USB pin, and give us a cleaner, more responsive effect. Connect the data pin on the NeoPixel strip to the D5 pin on the ItsyBitsy M0. D5 is an output-only pin specifically designed to work with the 5 V logic of NeoPixels, so it's perfect for our longer strip. Next, connect the data pin on the external lights to the D9 pin on the ItsyBitsy. And finally, connect the OUT pin on the microphone to A0.

Take a moment to gaze upon the lovely mess of wires coming out of the ItsyBitsy M0. Exciting, isn't it? Now, let's program it and make things light up!


ITSY BITSY M0
3V
G
A0
USB
D9
D5
MIC
-
OUT
+
in
out
+
-
NEOPIXEL STRIPS (Interior Face Lights)
NEOPIXEL STRIPS (Interior Face Lights)
NEOPIXEL STRIPS (Interior Face Lights)
NEOPIXEL STRIPS (Interior Face Lights)

Above
The two sets of NeoPixels are run from D5 and D9

PROGRAM THE ITSYBITSY M0

Plug the ItsyBitsy into your computer with a micro USB cable. The ItsyBitsy will appear on your computer as a drive called CIRCUITPY.

As always, start by making sure your microcontroller is up to date before beginning to code. Visit the online guide for ItsyBitsy – at **hsmag.cc/CAbyDw** – and follow the instructions for updating CircuitPython and the libraries on the board. For this program, you'll only need the neopixel.mpy library.

If this is your first time working with CircuitPython, visit **hsmag.cc/JglQCn,** for an introduction and troubleshooting help. Technically, you can use any text editor to write and edit CircuitPython code, but it's highly recommended to use a dedicated code editor. We love Mu Editor, which has a CircuitPython mode and contains helpers and a serial monitor for debugging.

THE CODE

Our code starts with importing the necessary libraries. Then, we set up the external lights as a NeoPixel strip, with two pixels on the D9 pin. Since these lights will react to the microphone input, we'll call this variable `voicePixStrip`.

```
import board
import neopixel

from analogio import AnalogIn

voicePixNum = 2
voicePixPin = board.D9
voicePixStrip = neopixel.NeoPixel(voicePixPin,
voicePixNum, brightness=0.3, auto_write=False)

voiceColor = (0, 250, 90)  #teal
```

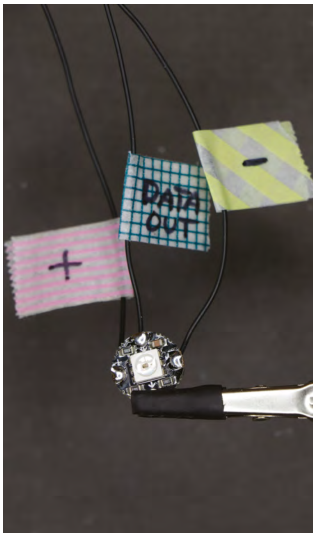
Next, we set up the internal NeoPixels as a strip called `facePixStrip`. Set `facePixNum` to the total number of pixels in your strip. For our helmet, this is 48.

```
facePixNum = 48
facePixPin = board.D5
facePixStrip = neopixel.NeoPixel(facePixPin,
facePixNum, brightness=0.3, auto_write=False)
faceColor = (200, 150, 50)  #warm white
```

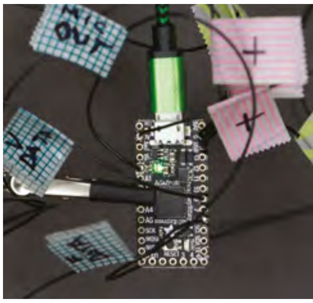
Notice that we have created variables for the colours of each strip: `voiceColor` and `faceColor`. Change the RGB values (red, green, blue) in →

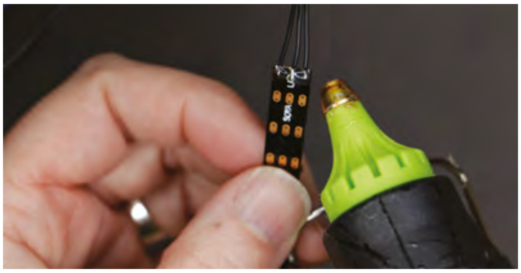
Left
The NeoPixel strips ready for use

Below
Keep your wires labelled for easy identification



Below
Everything connected up, ready to go





Right If in doubt, glue-gun it!

Above Strips and microphone held in place

Below The NeoPixel goes in the chin area

these variables to tune the colours of each strip to your liking. Each RGB value can be set from 0 to 255.

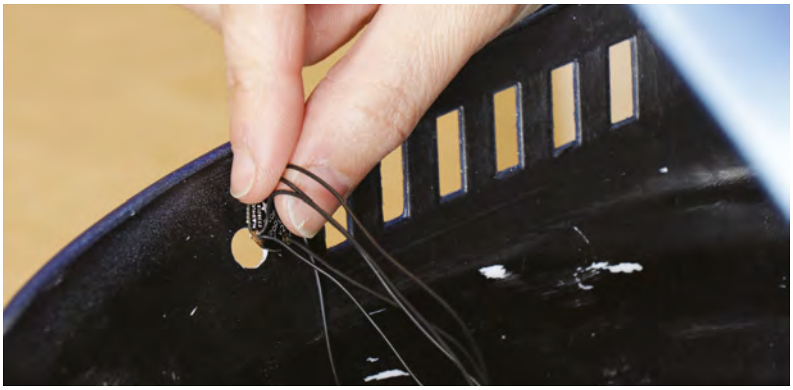
The next few lines set up the microphone on pin A0 and reads the microphone to set a base level. We'll use the **minVol** variable to set the minimum increase in sound needed to trigger the **voicePixStrip**.

```
microphone = AnalogIn(board.A0)
# Take an initial reading of the mic as the base
level
baseLevel = int((microphone.value)/1000)
# Set the minimum increase in volume needed to
trigger active state)

minVol = baseLevel + 4
```

Because we want the internal lights to stay on all the time, we can turn them on before the main loop of our code. This means the ItsyBitsy only has to turn them on once, instead of over and over again while running the main loop.

```
#  Turn on the facePixStrip pixels
facePixStrip.fill(faceColor)

facePixStrip.show()
```

The main loop of our code is simple. We start by reading the microphone and creating a new variable for volume. Next, we compare the current volume with the base level reading. If the incoming volume detected by the microphone increases by more than the value of **minVol**, the **voicePixStrip** will turn on.



Otherwise the pixels will be set to black (0, 0, 0) and will not illuminate.

```
while True:

    # Turn the microphone input value into a
manageable integer to compare with the base level.
    micInput = microphone.value
    volume = int((micInput)/1000)
    print("base: ", baseLevel)
    print("volume:", volume)

    # When the microphone detects sound louder
than the baseline level, fill all voicePixStrip
with color.
    # Otherwise, turn strip off (black).
    if volume >= minVol:
        voicePixStrip.fill(voiceColor)
        voicePixStrip.show()
    else:
        voicePixStrip.fill((0, 0, 0))
        voicePixStrip.show()
```

Saving this code as the **code.py** file on your ItsyBitsy M0 will automatically restart the device and run the code. The NeoPixel strip on D5 should illuminate as soon as the device restarts. See if the two pixels on D9 turn on when you speak into the microphone. If needed, adjust the microphone's sensitivity by increasing or decreasing the **minVol** variable. When your circuit is working as intended, it's finally time to install it!

INSTALL THE CIRCUIT

It's a good idea to reinforce your solder points, especially on the tiny NeoPixel strips, by covering them with a small dab of hot glue. Be very careful when moving your completed circuit around so as not to snag wires or damage anything. Don't remove the labels from your wiring until the circuit is installed and you've confirmed everything is still working.

With the visor removed, place the entire circuit inside your helmet. Start by attaching the face lights. For the helmet shown, it was easiest to work from the end of the strip backwards, gluing each section in place around the inside of the face opening. Try to position the lights so that they will shine onto the face without pointing directly into the wearer's eyes. Attach the strips with hot glue or double-stick foam tape.

Next, affix the microphone to the inside of the helmet, somewhere near where your mouth will be, with double-stick foam tape or hot glue. Orient the microphone so that the wires can be routed easily toward the ItsyBitsy's location.

Now, the external lights. If needed, drill a small hole on both sides of your helmet for the lights to show through. Glue each NeoPixel in place, on the inside of the helmet, making sure the wires between them can be routed around the inside of the helmet.

Glue the ItsyBitsy in place at the back of the helmet. Make sure the ItsyBitsy's USB port faces down, so that when plugged in, the USB cable will feed out the bottom of the helmet.

Tidy up the wires inside the helmet (and prevent snagging) by tacking them down with hot glue into neat little bundles. To make the hot glue tacks invisible inside the helmet, cover them with a layer of black acrylic paint. Now is also a good time to touch up any black paint that was scraped off inside the helmet during installation.

Connect the ItsyBitsy M0 to your USB power bank and check to make sure that everything still works. If so, hooray! You can finally remove the labels from your wiring! If not, check all your connections and make repairs.

MAKE THE DIFFUSERS

Now it's time to put the final touches on our helmet. Reinstall the visor and replace any bits and pieces that were previously removed for better access.

LEDs always look better with a little diffusion, but sometimes it's hard to find a translucent 'greeble' that's just right for your project. Here, we made custom diffusers for our helmet using a very familiar material: hot glue! Here's how to do it:

First, make a mould of something that is a good shape and size for your diffuser. In our project, we used a small plastic cap (it came off your author's tripod). Two-part moulding putty is a great choice for a small object like this. Once mixed, the moulding putty cures in about 30 minutes.

Once cured, fill the mould with hot glue and let cool. This amount of hot glue takes a while to cool down. If you're too excited to wait it out, put it in the refrigerator for a few minutes. When cool, your hot glue part should slide right out of the mould! Repeat the process so that you have two identical parts.

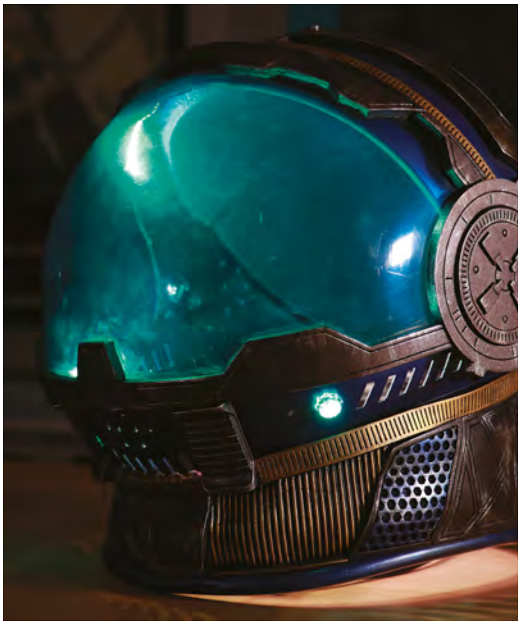
You can leave your diffusers plain, or paint them as desired to fit your helmet design. If painting, leave the top plain for maximum brightness. Have fun playing with this technique, and do experiment with paint to customise your diffusers. Then glue them in place over the NeoPixels on either side of the helmet. Your helmet is done!

BLAST OFF!

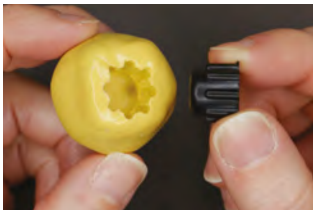
To wear your helmet, plug the ItsyBitsy into the USB power bank and stow the power bank in your pocket (somewhere easy to reach, should you want to turn your helmet off). Turn on the helmet before putting it on. Remember that our code has the microphone take a base-level reading on start, so you may need to press the 'reset' button on the ItsyBitsy M0 to recalibrate the microphone once the USB power bank is on.

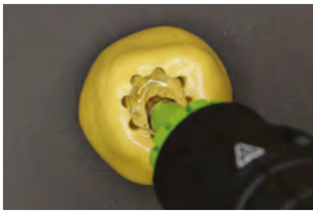
Got a great cosplay project to share? Show us your makes at **hackspace@raspberrypi.org**! □

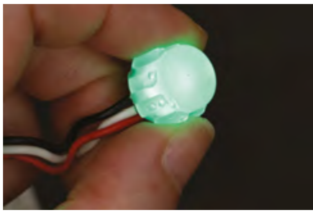
Below
Reassembling everything, once you're ready

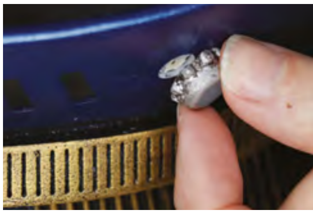


Above
One last touch-up









Above
Moulding your own LED diffusers with a glue gun

Left
The final helmet, ready for use

Automate your desk: Out-of-office box

Shout at visitors without raising your voice

Above
Nobody said it had to be pretty on the outside – take it as motivation to make a nicer job. A bit of papier mâché, and you can have a festive-themed box!

Tired of people dropping by your desk and leaving a Post-it notes cluttering the place up? Maybe you bought some biscuits to celebrate an occasion, but sadly you have back-to-back meetings elsewhere? You could actually be at your cubicle and answer questions without opening your mouth, just by pressing a button on the remote. Fear not! Your own voice can be set in a box to have a welcoming (or non-welcoming) message, which will be triggered by a remote or a distance sensor.

This tutorial hinges on five basic components: the Adafruit Audio FX Sound Board, the HC-SR04 distance sensor, IR remote with matching sensor, a speaker, and an Uno.

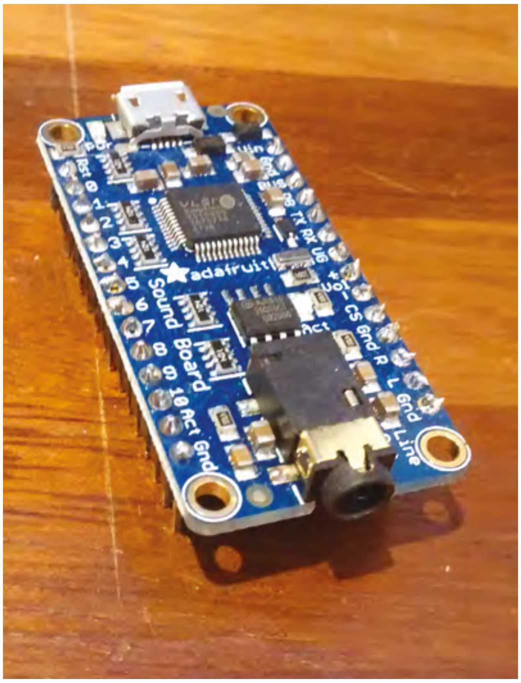
Here, we used a basic set of speakers – that is, non-Bluetooth ones that have a good old-fashioned audio jack. USB-powered speakers are easy to hack, as they just take 5V input. To get to the power lines, cut open the USB cable, then connect the red cable to the 5V supply, and black to ground.

If you'd rather not dismantle, you could use a power supply module with a USB port that you can wire into your circuit, or use battery-powered speakers so you don't have to power it from your Uno.

LEARNING TO SPEAK

We'll make our sounds using the Adafruit Audio FX Sound Board. This little gadget earned a 10/10 in our Best Of Breed article in issue 8, and it is worth all its points and more. We can trigger it from our microcontroller. In order to do this, you'll need to install a few libraries – you'll find details of the relevant software setup at: **hsmag.cc/PvtcUD**.

As well as hardware, we also need some sounds to play. You can record using your phone recorder, or Audacity on your computer. To save space, make sure you save them in Ogg Vorbis format.



Above
You can get audio boards with different amounts of storage space, so you can record as much or as little as you like

You can say whatever you want, but here are a few suggestions:

- Help yourself to some biscuits.
- No Josh, I will not do the minutes for you.
- I am rather busy right now, sorry I can't help you.
- I'm not here, please leave a note!
- Leave the pen.

Alternatively, you can get some Creative Commons sounds from archives such as **freesound.org**.

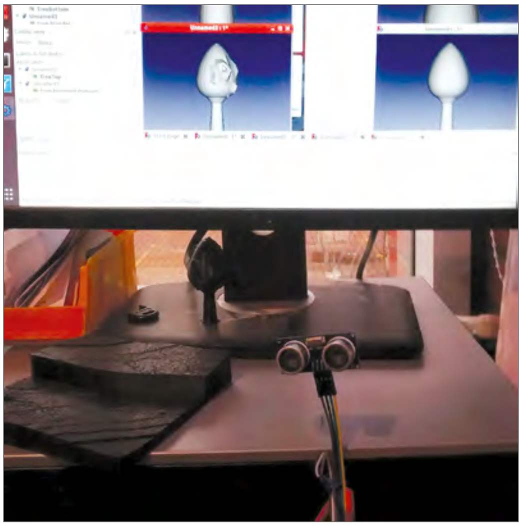
GO LO-FI

Just drag and drop the files into the board. It might help to give the files some descriptive naming or just number them up. If you got the cheapest board, you have 2MB of sounds to use, which is plenty for some short clips.

CHECK YOUR DISTANCE

We will be using the HC-SR04 ultrasonic sensor that is found in the Elegoo kit. The Arduino IDE now offers an easy way to add the library to your setup, but should you want to go manual, you can find it on GitHub: **hsmag.cc/mAJijW**.

Note the angles at which the sensors work: it can be quite wide, so you might want to carefully think of a place to put your box in order to take advantage of this.



Above
The distance sensor could be changed to any other sensor, like a simple light sensor, but the ability to work with different distances makes this HC-SR04 really versatile

REMOTE CONTROL

You can find the remote and receiver in your Elegoo kit. As with the distance sensor, you'll find it in many different kits as well as the one we've used. The Arduino library for it also has very good support for many different boards. You can also download it manually from GitHub: **hsmag.cc/plTjVa**.

> "The Arduino library also has very good support for many different boards"

ADDING BRAINS

In your Elegoo kit, you will have an Uno, but you can use official Arduino boards as well. The full code is available from **hsmag.cc/QyXAQK**, but let's take a look at what it does.

Let's start with the preamble:

```
#include "IRremote.h"
#include <SoftwareSerial.h>
#include "Adafruit_Soundboard.h"
#include <HCSR04.h>

#define SFX_TX 5
```

Andres Muniz-Piniella

c4ad.eu

When not making demonstrators for companies using electronics, CAD and machinery, he does nano-material characterisation using AFM, and runs his Community Interest Company making activities called STEAM punky.

Above
This is the remote control that came with the kit, but other ones should also work

YOU'LL NEED

- **Elegoo starter kit**
- **Adafruit Audio FX Mini Sound Board**
- **Small USB-powered speaker, with phone jack**
- **A small cardboard box within another box** (optional)
- **Some tape**

```
#define SFX_RX 6
#define SFX_RST 4

SoftwareSerial ss = SoftwareSerial(SFX_TX, SFX_
RX);
Adafruit_Soundboard sfx = Adafruit_
Soundboard(&ss, NULL, SFX_RST)
int receiver = 11; // Signal Pin of IR receiver
to Arduino Digital Pin 11
IRrecv irrecv(receiver);     // create instance
of 'irrecv'
decode_results results;      // create instance
of 'decode_results'
HCSR04 hc(10,9);//initialisation class HCSR04

(trig pin , echo pin)
```

In this section of the code, the libraries are loaded, and some variable and functions are referenced and initialised.

Your IR sensor will go to pin 11. The trigger for your HC-SR04 will go to pin 10, while echo will go to pin 9. For your sound board, pin 6 goes to Rx, pin 5 to Tx, and pin 4 to RST.

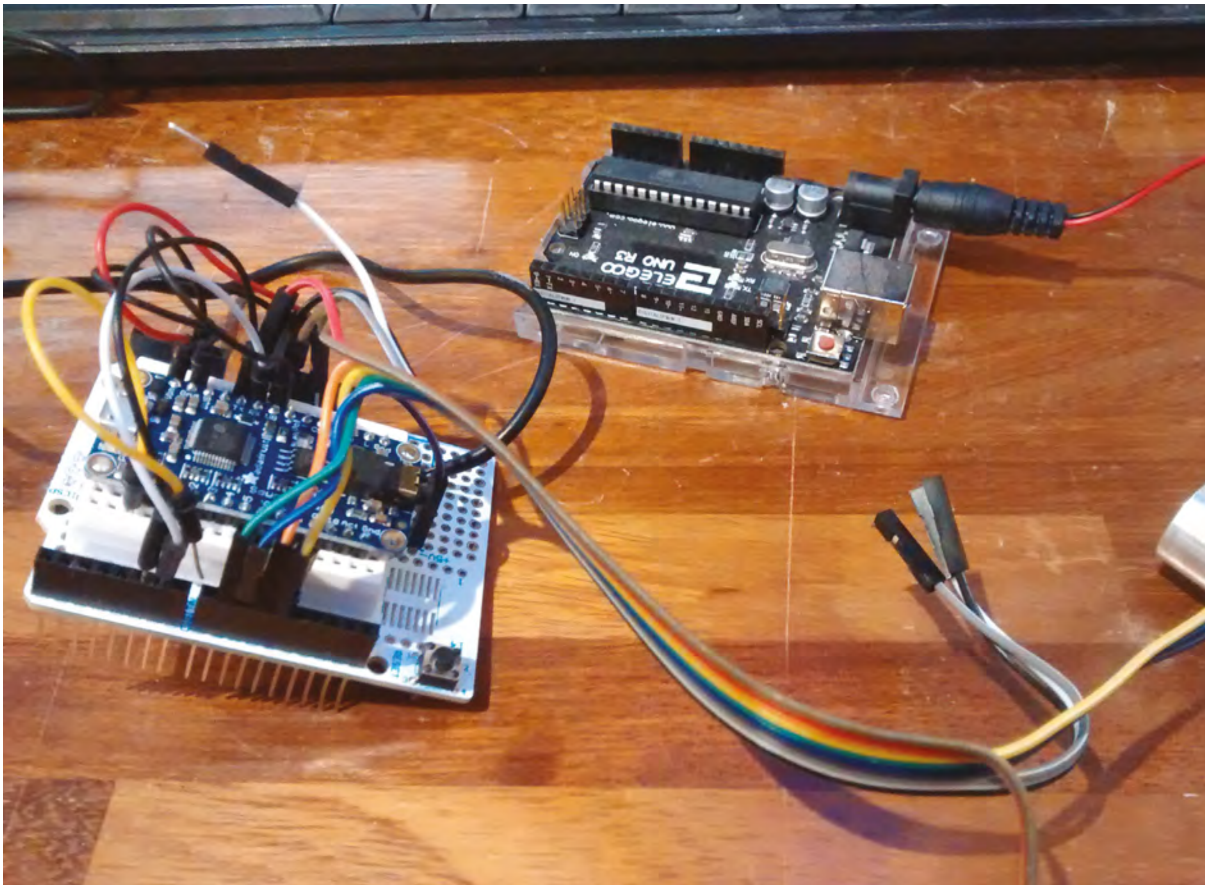
The next important component of the code is the **setup()** function:

```
void setup() {
  Serial.begin(115200);
  Serial.println("Adafruit Sound Board!");
  ss.begin(9600);
  if (!sfx.reset()) {
    Serial.println("Not found");
    while (1);
  }
  Serial.println("SFX board found");
  Serial.println("IR Receiver Button Decode");
  irrecv.enableIRIn(); // Start the receiver

}
```

Here, we set up the communication protocol. Notice that we have two serial connections: one for printing out messages (115200 baud) for us to

QUICK TIP

For a bit of fun, have the distance sensor trigger only when very far away but not when near your desk.



Right For this tutorial, a ProtoShield was used. This enables a semi-permanent and compact solution. Also, kits have many different and colourful cables you can use debug, and one for the communication between the Uno and the sound effect board (9600 board).

Once our program is done, you could remove the **println** statements if you want.

Let’s move to the main loop:

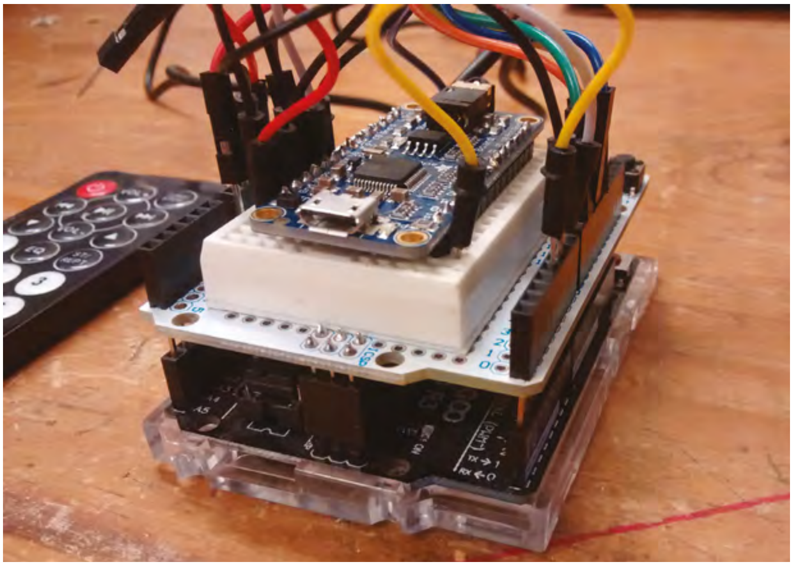
```
void loop() {
   flushInput();
    if  ( hc.dist()< 40)
      {
           if (! sfx.playTrack((uint8_t)7)) {
          Serial.println(“Failed to play
track?”);
           }
      }
        if (  irrecv.decode(&results)) // have we
received an IR signal?
         {
          translateIR();
          irrecv.resume(); // receive the next
value
          Serial.println(“IR online after IF”);
          }
 delay(100);
}
```

The first **if** statement is for our distance sensor. As mentioned, you might want to position your box in a place that is hidden away. Currently, it is set to trigger when anything is closer than 40 cm. Note: it will trigger continuously. So, if someone sits down at your desk, it could be saying “that’s my spot” in a very annoying way in a continuous loop. In this example, it will play the seventh audio track – set in the statement **sfx.playTrack((uint8_t)7)**. You can change this to play whichever track you’d like.

> **We have two serial connections: one for debug** and one for the sound effect board

The next **if** is the part listening in for your remote. This part is trickier, as it needs to translate the IR signals and then reinitialise itself for more commands.

We found a short delay helped keep everything running smoothly, so there’s a pause of 100 ms before restarting the loop.



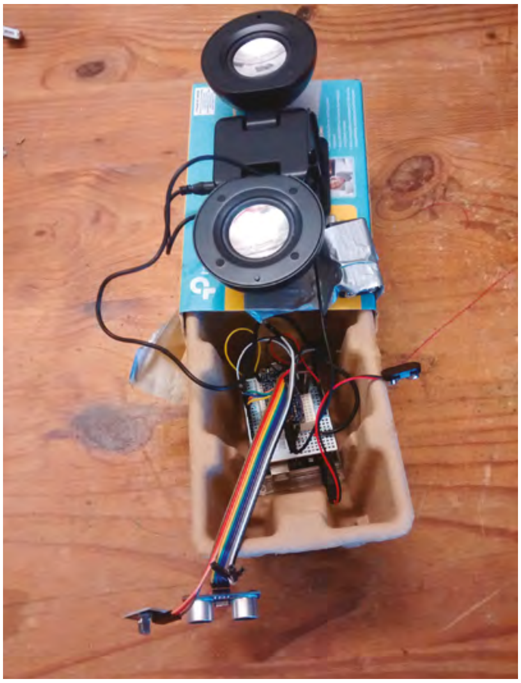
Above It does not need to be pretty on the inside! By using a breadboard shield, it is easy to plug the small sound board in and use jumper cables from the Uno to the modules and speaker. They all share the same 5V and GND supply

FINISHING TOUCHES

You’ve got it working on a separate breadboard, now it’s time to squeeze it into a box. Use the female-to-male DuPont wire provided in the kit; it should be long enough at more than ten centimetres.

Set the battery leads and speaker leads to a similar length.

Use the outer box to feed the cables and modules outside: speaker, battery, and sensors. If you can get hold of a battery holder that has a switch, that would make it more convenient to turn off after a hard working day. Tape it into place, where you find useful, and use the same tape to keep the box closed. □



Left It is bigger on the inside!

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 79 | March 2019 | magpi.cc
WIN! OFFICIAL KEYBOARD & MOUSE KITS
DEVELOP AN ANDROID APP
Build & install your own apps with Raspberry Pi
MAKER SPECIAL!
OF MAKING
projects to create & share
SET UP A SMART

PLUS! FREE CASE, THREE COVERS & CABLES

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

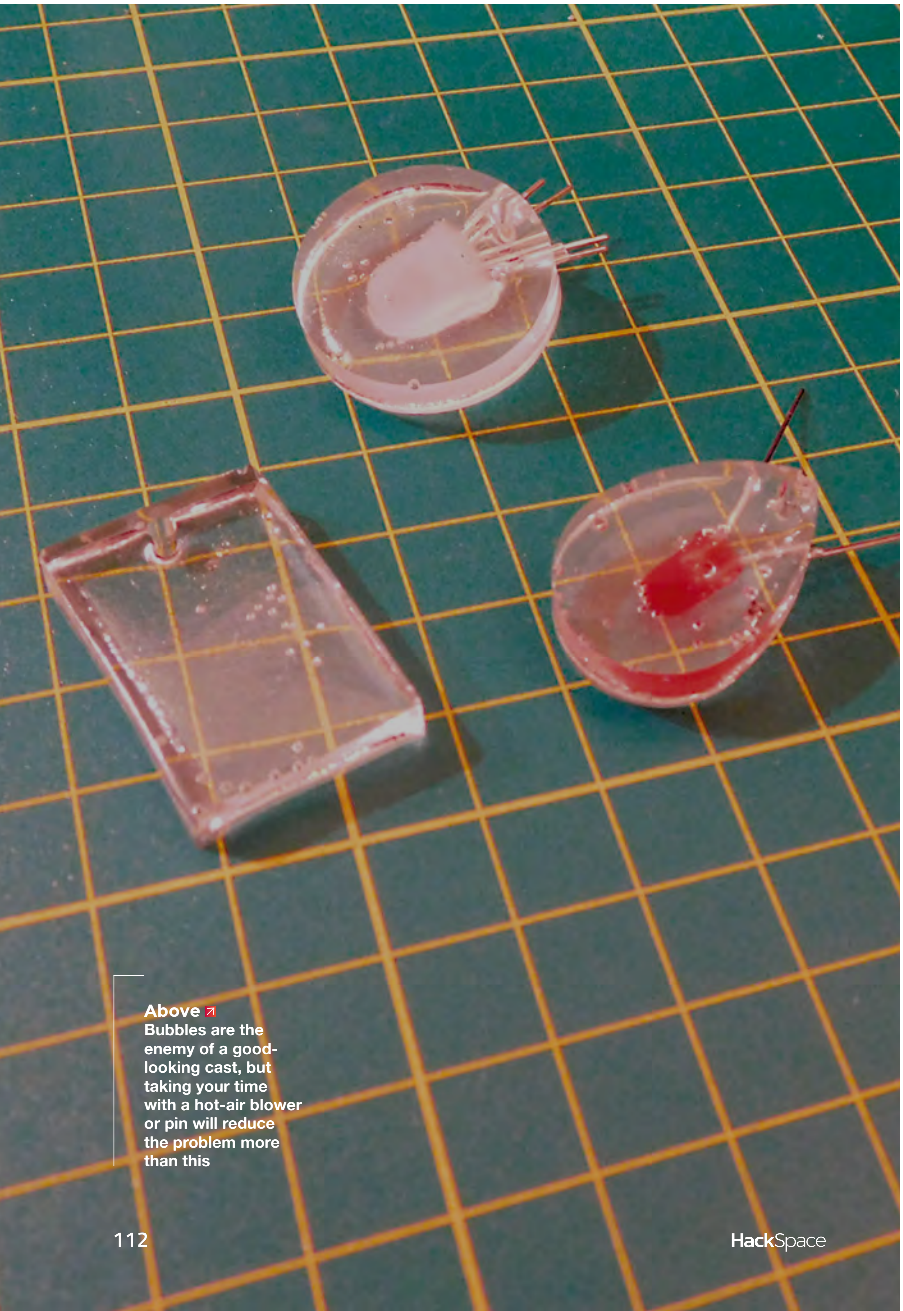
Hacker gear poked, prodded, taken apart, and investigated

DIRECT FROM **SHENZHEN**

Resin kit

A complete kit for getting started with moulds

By **Ben Everard** @ben_everard



Above Bubbles are the enemy of a good-looking cast, but taking your time with a hot-air blower or pin will reduce the problem more than this

Casting things in resin is a fantastic maker skill. You can use it to create unique objects, protect electronics, re-enforce split wood, and much more. It doesn't even have to be expensive. You can make moulds yourself and, in bulk, resins don't cost too much. However, if you just want to have a bit of an experiment, you might find yourself with a high up-front cost before you do your first pour. We took to direct-from-China websites to see if we could find a cost-effective way of getting up and running.

For £4.56 (including delivery to the UK), from hotselling-3C store on AliExpress, we got a resin casting starter kit, including 100 ml of resin, five jewellery-style moulds, and assorted mixers and measurers. In short, it gave us everything we needed to get started.

The resin, like most resins, came as an A-B mixture. That means that there are two liquids that have to be combined in the correct ratio. Once combined and mixed thoroughly, they can be poured into the moulds where they'll slowly set into a hard, clear, plastic-like material. The kit came with absolutely no information other than the mix ratios (3:1), so we didn't know how long it would take to cure. This isn't a huge problem as there's no harm in leaving it longer than it needs.

We followed these minimal instructions and tried making four pendants. In the spirit of this issue, three of them included LEDs: one NeoPixel, one two-colour flasher, and one red. The other was a plain pendant, to test out the mixture.

GUESS-WORK

There were no cure times included, so we just had to wait and see what happened. After 24 hours, the resin was fairly hard, and it was another three days before it could really be called 'set'. However, despite this time, it remained slightly tacky on the

Left The kit also includes gloves and some scented wipes

surface where it was exposed to the air. The room we made this in wasn't particularly damp or cold, though it's possible that this particular resin is more sensitive than usual to these environmental factors which can leave epoxy resin tacky. We sanded it down and this left us with a clean, solid cast. On subsequent uses, we didn't get this tacky surface, so we can put it down to an environmental factor, rather than a fundamental problem with the resin.

Once it was set, the casts popped out of the moulds easily. Silicon moulds like this can be reused numerous times.

This is undoubtedly an introduction kit. It gives you the bits you need to experiment with resin, using some moulds designed for jewellery. You can try embedding different things, or using different additives to make interesting and unique accessories, or you could use them for experiments as you get used to the general process of resin casting. There's enough resin for some practice runs (four pendants used 12 ml of the 100 ml total, but much of the space was taken up by LEDs), but not enough for any serious pours. Once you've used up this kit, you'll be in a better position to pick your next resin.

> "**You can try embedding different things,** or using different additives to make interesting ... accessories"

The downside of this kit is that there are no instructions, so the user is left on their own. However, we didn't find this too onerous. It would be nice to have some cure times, and some information about the actual type of resin we had but, given the price, we can work around this. If you want more guidance, you can certainly get kits from larger companies that are better documented, but they'll cost more.

The basics of resin casting, fortunately, aren't complicated, and really come down to mixing the two liquids together, adding any additives such as glitter or colouring, then pouring into a mould and waiting for it to cure. Our top tips for getting started with a kit like this are:

- Measure your quantities as accurately as possible. This kit included a pot marked with 1 ml increments and a pipette.
- Mix thoroughly. Keep on mixing, even after you think you're probably done. Pay particular attention to scraping the bottom and sides of the pot you're mixing in.
- Wear gloves. Resin isn't particularly nice stuff to get on your skin.
- Leave your resin to set in a dry, warm room as it's sensitive to damp and cold.
- Experiment with small items before embarking on a large project.

If you're willing to get stuck in and learn by doing, then this kit offers fantastic value and the chance to learn a new skill for a fiver, and still have change for a pack of Chewits. □

ONLY THE BEST

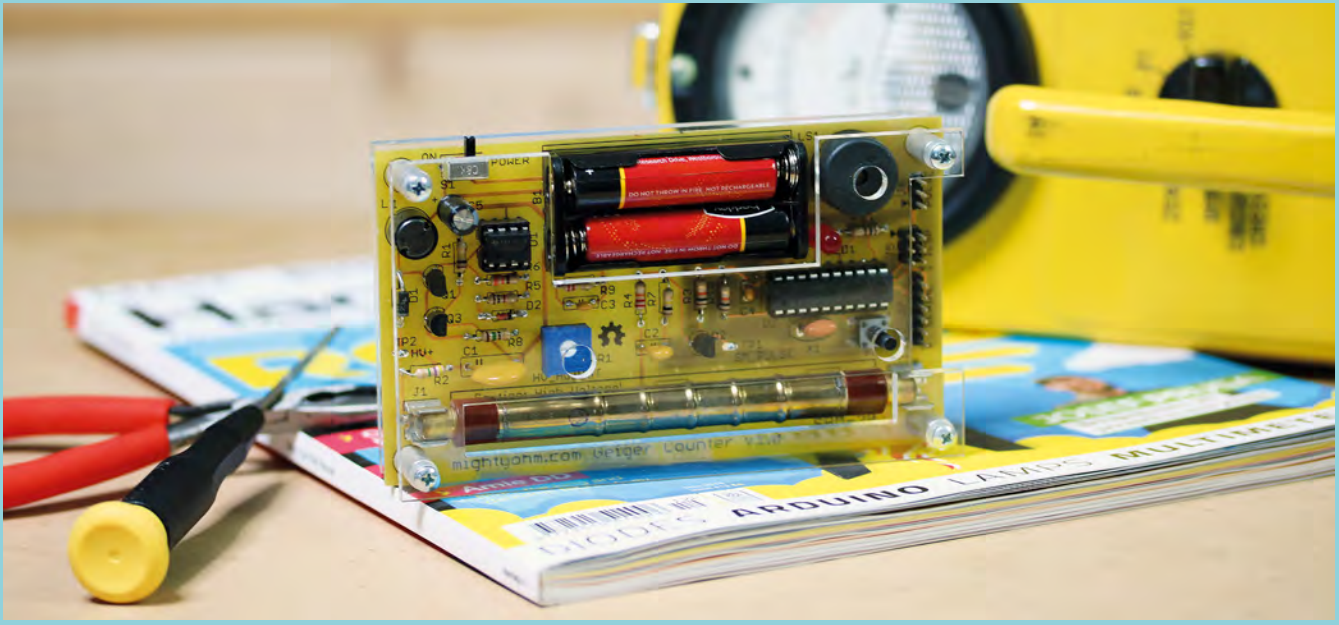
Environmental sensors for the backyard scientist

A roundup of the more unusual DIY-friendly environmental sensors

By Marc de Vinck @devinck

In this Best of Breed article, we're going to take a look at some of our favourite environmental sensors. And not just your typical light or humidity one-trick pony variety of sensors. All the kits in this review are unique or innovative because they either feature multiple sensing capabilities or are unique in some other way, like the MightyOhm Geiger Counter. It's not like you see a kit like that every day! And although a roundup of simple environmental sensors may be a good topic for a future Best of Breed, in this case we wanted to look at some of the more unique and innovative kits available to electronic hobbyists. So, let's dive right in and look at how you can easily add environmental sensing to your next project.

Below We can confirm HackSpace magazine is not radioactive

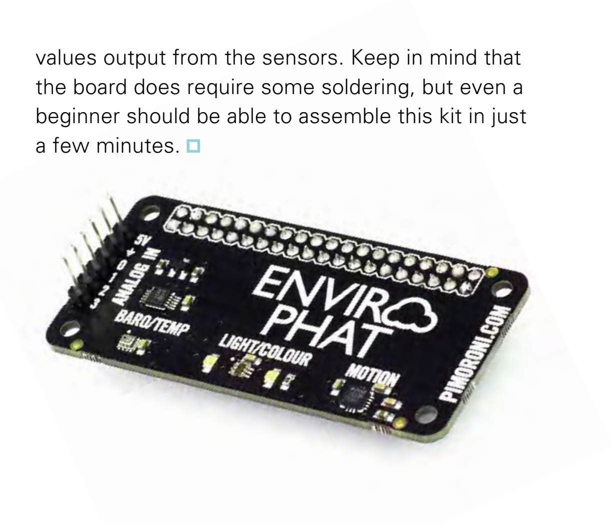


Enviro pHAT vs enviro:bit

Two ways of creating DIY weather sensors

ENVIRO PHAT ◈ $23 | pimoroni.com

The Enviro pHAT features four different sensors allowing you to easily measure temperature, pressure, light level, colour, three-axis motion, compass heading, and additional analogue inputs. And because it's compatible with the Raspberry Pi 3B+, 3, 2, B+, A+, Zero, and Zero W, you can fairly easily set up a remote environmental monitoring station that can be accessed from anywhere in the world. And if all those sensors aren't enough, you can also combine this pHAT with another HAT or pHAT to add additional capabilities. It should be fairly simple to add a display for easy visual feedback, or graph of values output from the sensors. Keep in mind that the board does require some soldering, but even a beginner should be able to assemble this kit in just a few minutes. □



Left ◈
A nice suite of sensors for your Pi

Below
More than just a few sensors, this kit includes a beautiful laser-cut stand

ENVIRO:BIT ◈ $54 | pimoroni.com

Most environmental sensors are some variant of a breakout board that attaches to your microcontroller or computer. And although that's fine for many projects, sometimes you want a more refined kit, and that's where the enviro:bit micro:bit Kit comes into play. Not only does this kit feature a breakout board with multiple sensors, it also has a fun little laser-cut stand to make your own unique weather station. The kit even includes a micro:bit, a Pimoroni enviro:bit board, along with everything you need to get up and running.

All the electronics come fully assembled and ready to use. The Pimoroni enviro:bit board includes sensors to read temperature, pressure, humidity, light, colour, and sound. You can program the board using MakeCode, and no soldering is required. If you want an adorably fun and programmable little weather station, this kit is for you. □

> "All the electronics come fully assembled and ready to use"

VERDICT

Enviro pHAT

An easy way to add a few environmental sensors to your Pi.

9/10

enviro:bit micro:bit Kit

Easily add a few sensors, along with a way to hold them in place.

10/10

BEST OF BREED

Adafruit BME680

Temperature, humidity, pressure, and gas sensor

ADAFRUIT $22.50 | adafruit.com

The Adafruit BME680 includes a lot of environmental sensing on a tiny package. This board is able to measure temperature, humidity, barometric pressure, and VOC gases, and all over SPI or I^{2}C, making it simple to use in almost any project.

The sensor featured on the board is a BME680 from Bosch. It's capable of measuring humidity with ±3% accuracy, barometric pressure with ±1 hPa absolute accuracy, and temperature with ±1.0°C accuracy. Not too bad for a board that measures 18 mm × 16.5 mm × 2.8 mm. And keep in mind, since air pressure changes with altitude, as an added bonus, you can also use this sensor as an altimeter with ±1 metre of accuracy.

In addition to all those capabilities, what really makes this little board stand out is the MOX sensor and its ability to measure volatile organic compounds in the air. This means it can be used to detect gasses or alcohols like ethanol, alcohol, and even carbon monoxide. Just remember, it only gives you a value, not a concentration amount. It should never be used to measure dangerous environments. Leave that to dedicated, tested, and certified, commercial devices. □

Above
So many sensors are packed into this little breakout board

VERDICT

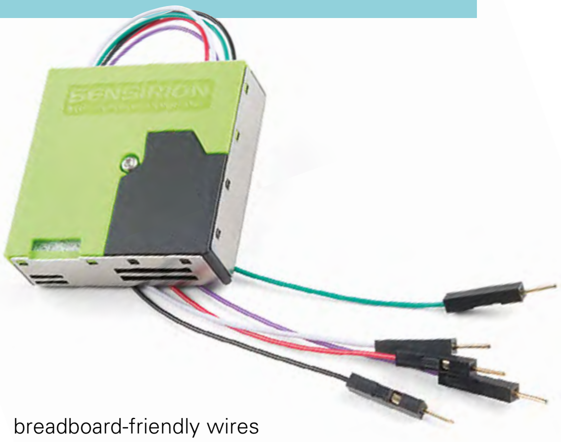
A small form factor and great documentation.

9/10

Particulate Matter Sensor

Checking air quality

SPARKFUN $46.95 | sparkfun.com



The Sensirion Particulate Matter Sensor, from SparkFun, is a high-quality and compact sensor that uses laser scattering and Sensirion's contamination resistance technology to accurately output particle measurements. You can measure mass concentration and the number of particles at 1 µg/m3, 2.5 µg/m3, 4 µg/m3, and 10 µg/m3. All of that technology in a package that's only 41 mm × 41 mm × 12 mm. Communicating with the sensor is made easy by the included 5-pin cable, which has five breadboard-friendly wires that are colour coded for simple hook-ups. Sensirion also provides drivers for both the UART protocol and I^{2}C. Just keep in mind that if you plan to use this with your Arduino, it works best with UART and not I^{2}C. As always, check out the datasheet for more information. □

Left
A high-quality sensor that's fairly easy to use

VERDICT

An innovative solution to measuring particulate.

9/10

Adafruit STEMMA Soil Sensor

For the green-fingered maker

ADAFRUIT ◈ $7.50 | adafruit.com

There are a lot of different soil moisture sensors out there, but most have the same issues, like long-term reliability and accuracy. That's because they typically use a resistive measurement to gauge the conductivity of the soil. This method works fine at first but, over time, the measurements become more and more inaccurate due to the corrosion.

That's where the Adafruit STEMMA Soil Sensor comes into play. It uses capacitive sensing to measure the moisture level in the soil. The advantages of this technique are long-term reliability and accuracy, due to the fact that the actual metal sensor does not need to be in direct contact with the soil, which in turn virtually eliminates corrosion and long-term failures. It's a smart solution, made even smarter by integrating I²C communications, allowing for a simple and reliable hook-up to your microcontroller. □

> "The advantages of this technique are **long-term reliability and accuracy**"

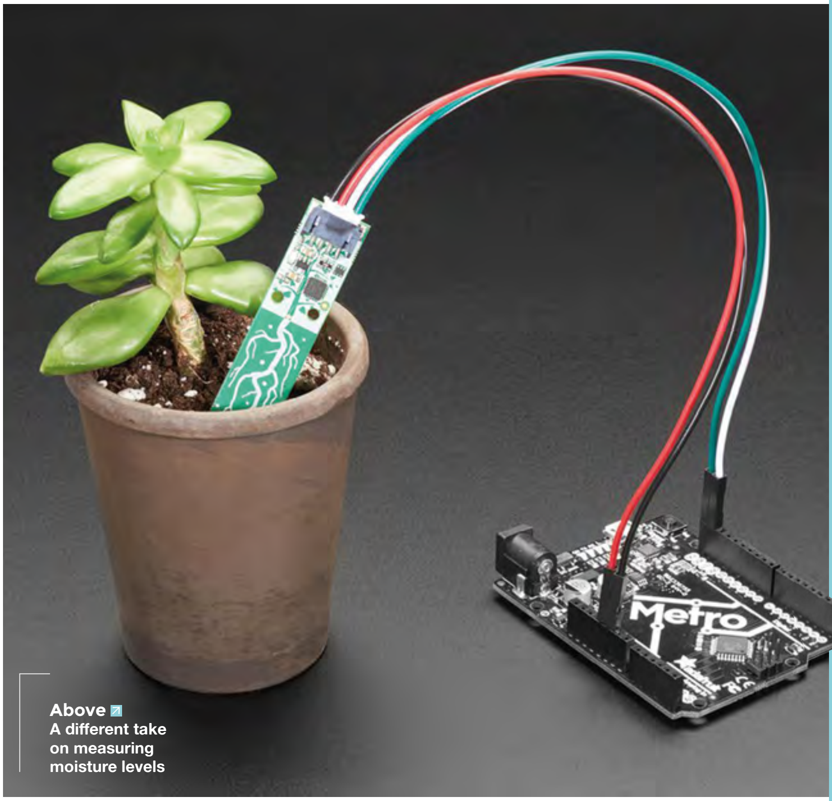
ADAFRUIT.IO

Once you've got your sensor readings, what do you do with them? If you want to track them over time, and get them to interact with other devices, it can be useful to save them in some form of cloud service, and Adafruit.io is a great option for this. The free tier lets you save 30 data points a minute and store it for 30 days. You can then view this data on the site. You can interact with other systems using IFTTT or the API. This works with any boards that have an internet connection (regardless of manufacturer).

VERDICT

This isn't your typical moisture sensor: this one is designed to last.

9 /10



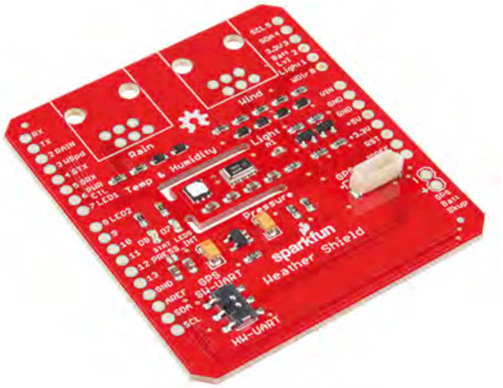
Above
A different take on measuring moisture levels

SparkFun Weather Shield

Keeping an eye out for rain

SPARKFUN ◈ $39.95 | sparkfun.com

The SparkFun Weather Shield is a simple-to-use Arduino shield that features barometric pressure, relative humidity, luminosity, and temperature. It also has the option to expand and include more sensors like wind speed and direction, rain gauge, and even GPS for accurate locational data and time. It's a nice shield that includes the basics to get you started, but also allows for future expansion if needed. Keep in mind, the shield doesn't come with any header pins, which is a bit surprising, so be sure to grab some if you decide to purchase this shield for your next project. □



Left ◈
An expandable weather shield for Arduino

VERDICT

Easily expand this shield to fit your needs.

9/10

Sensor Puck by Silicon Labs

Get data on your phone

SILABS ◈ $29.99 | silabs.com

The Silicon Labs Sensor Puck provides you with a really convenient way to gather your environmental data. With its integrated EFM32 MCU, which features Bluetooth Low Energy, you can easily connect the coin cell-sized sensor puck to your smartphone and read the sensor's data in real time. That's the real value with this product.

The sensor puck can measure the UV index, ambient light, relative humidity, and temperature. It can also detect your heart rate if you place your finger over the Si1147 sensor. For optimal power management, which is really important in such a small form factor, the board features a Touchstone TS3310 boost DC-DC converter. Another interesting feature is the ability to use it in broadcast mode, which allows the user to gather data from multiple modules at the same time. Check out the product page for more information, including a link to download the iOS or Android app. □

Left ◈
A convenient form factor for a variety of sensors

VERDICT

Easily access sensor data with your smartphone.

8/10

MightyOhm Geiger Counter kit

There's radiation everywhere!

MIGHTYOHM ◈ $99.95 | mightyohm.com

Left ◈ A unique sensor that's lots of fun

I built the MightyOhm Geiger Counter kit many years ago, and it's still one of my favourite kits. You might think that a Geiger counter is only useful if you are worried about a nuclear meltdown or explosion, but the truth is, beta and gamma radiation is all around us, all of the time. I have used it to detect the various amount of radiation in different products around my house, including my granite kitchen counters. I have to admit, when the Geiger counter starts clicking, it is a little unsettling, but it's also a fun thing to show guests at your next party. Just don't freak them out too much! There are several versions of the kit, from a bare PCB, to a complete kit which includes the Geiger-Muller tube and an enclosure. You'll need only basic soldering skills for assembly, and it doesn't take too long, depending on your skill level. Just remember that this kit, along with all the other kits in this article, is designed for educational use only. □

VERDICT

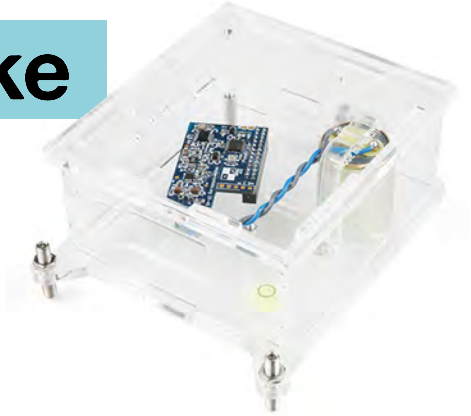
Not an inexpensive kit, but it's well worth the money.

9/10

Raspberry Shake

Did the earth just move?

SPARKFUN ◈ $215 | sparkfun.com



Left ◈ A highly accurate seismograph for experimenting

The Raspberry Shake is an easy to use earthquake monitor, or seismograph. You can use it to monitor your home, office, or bring into the classroom for experimenting. The kit contains an extremely sensitive vertical geophone as its sensor, which is used in various industries for accurately detecting even the slightest of vibrations. The kit also includes a protective case for the control circuitry for the geophone, but does not include the required Raspberry Pi and power supply. After some fairly simple assembly, you'll be able to monitor earthquakes, avalanches, eruptions, and more. If you don't want all the components in this kit, SparkFun also sells the geophone on its own for about $60. □

> "**The kit contains an extremely sensitive** vertical geophone as its sensor"

VERDICT

A highly accurate seismograph but, with high accuracy, comes a high price.

8/10

Right
It may be bright pink and look like fabric, but this plastic purse holds an exceptional 512 LED matrix

Can I Hack It?

A smart pixel purse?

Does a pink plastic handbag contain parts for an RGB LED scrolling sign?

Les Pounder

@biglesp

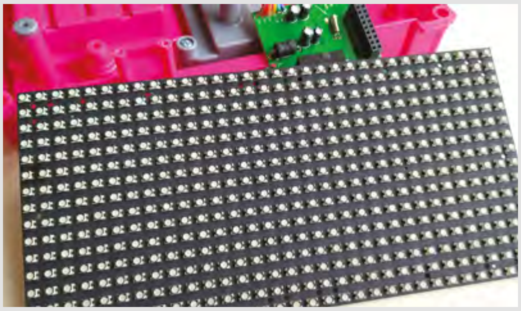
Les Pounder loves taking things to pieces and seeing how they work. He teaches others as part of the Raspberry Pi Foundation's Picademy event. He blogs over at **bigl.es**

This Project Mc² Smart Pixel Purse, looks like a handbag (known as a purse in American parlance) but, in reality, it houses lots of very hackable electronics. The purse was made as part of the *Project Mc²* Netflix series, which involves a team of girls working for government organisation NOV8 as spies, using their STEM skills to protect the world. The smart pixel purse is a vital piece of equipment, but can we hack it? Let's find out!

The purse is held together with a mixture of friction-fit plastic and six cross-head screws, and so taking the purse apart is relatively simple. The LED matrix screen has a clear plastic screen to protect it, and to separate the pixels, so that our image/text is clear.

ELECTRONICS

There are a lot of electronics in this purse that can be repurposed. Firstly, we have a 4 × AA battery compartment moulded into the purse. While this cannot be hacked and reused outside of the purse, it can be used as a power source for projects that reuse the purse. Moving on, we have a chunky switch, broken out into its own module. The switch module connects to another module which contains the 'brains', courtesy of a microcontroller hidden under an epoxy blob. But, we do have access to the board via an I²C interface which can be used for debug. All we need to do is solder on some header pins and we can communicate using I²C from a Raspberry Pi or Arduino. This board also boasts a connector that attaches to the LED matrix: more on the matrix later. This connector is a 2×8 IDC

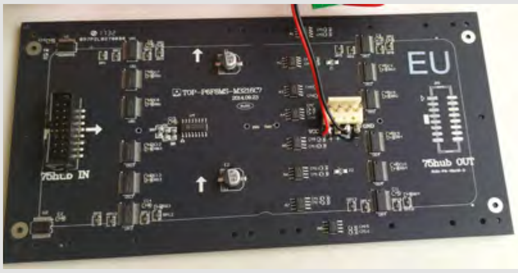


Above
With 512 LEDs in a 192mm by 99mm frame, this little board can show any colour at different levels of brightness. Scrolling text and images are no problem!

connector, and it has all of the pins needed for the microcontroller to communicate with the LED matrix, and handily it has the names of the pins broken out for reference. But the most important and brightest part of the purse is the 16×32 RGB LED matrix. 512 RGB pixels, which can be controlled to show any colour and brightness. The LED matrix also has separate power inputs so that we can supply 5V in addition. The pins used to connect our 5V supply are not polarity protected, so take care before plugging in.

The LED matrix has one IDC connector which connects to the microcontroller board, but the output IDC connector is unpopulated, meaning that we cannot daisy-chain further LED matrices without soldering our own connector. This can be done, and the connectors are quite cheap online. The power inputs for the matrix have some beefy solder joints, and our TS100 iron was unable to desolder them. But the four-pin interface used for power is another easily sourced component. Or, you can fashion your own.

Below
The rear of the matrix sees an IDC input used to connect to a microcontroller that controls the matrix. There is also a dedicated power connector with beefy solder joints



CODING YOUR MATRIX

Writing code for your RGB LED matrix comes in a few forms. For our test, we opted to use the Adafruit RGB Matrix Shield for Arduino. Coming as a kit, the shield requires a little soldering. Matching the Arduino Uno form factor, the board fits easily atop the Uno for a neat finish. The shield then has an IDC connection and a 5V output again for a neat project. This means that we can directly connect to the matrix. So, with that done, we visited the Adafruit learning guide for this board and followed the instructions to download the libraries necessary to interact with it. This took mere minutes and is a testament to the quality of Adafruit's guides. The libraries also brought with them example files to test our board. Loading the demo script PanelGFXDemo_16x32 and flashing it onto the Arduino is simplicity and in a few seconds we see a colour test / bouncing ball / scrolling text demo, proving that our board is working! So, what next? Well, some scrolling text is always fun. We opened scrolltext_16x32 and changed line 39 from "Adafruit 16x32 RGB LED Matrix" to "@biglesp woz ere" then flashed it to the Arduino. And then we saw our name in lights! With a little tinkering to the example code, we can easily make something cool!

HACKABILITY

The main area of hackability is the 16×32 RGB LED matrix. It can be easily disconnected from the purse and built into another project. So, we first attempted to connect the matrix to a Raspberry Pi using a very helpful Adafruit guide, which also gave us a pinout reference to use jumper wires to directly connect the Pi and matrix together, and supply an external 5V power source. Sadly this did not work, most likely due to the lack of a level-shifter chip such as a 74AHCT125. If you wish to use the matrix with a Pi, then purchase an Adafruit RGB Matrix Bonnet for Raspberry Pi, or the Adafruit RGB Matrix HAT – these both come with on-board level shifters and an IDC cable, so no messy jumper wires! We tested our board using another Adafruit product, an RGB Matrix Shield for Arduino. Coming as a kit, we soldered everything up and installed the libraries necessary. Then a quick demo script was tested, and everything worked.

> " **The main area of hackability** is the 16×32 RGB LED matrix "

CONCLUSION

The number of raw components that we can salvage from this toy are few; however, the largest component, the RGB LED matrix, makes it worthwhile. A typical 16×32 matrix can retail for £25 to £30 online, and if you are savvy and shop around, you can get this bag for £10.

If you are going to use the purse, then the internals can be easily hacked to accommodate an Arduino or Pi Zero and still be powered from the internal batteries or, with a little Dremel action, we can install a LiPo battery solution. □

YOU'LL NEED

Project Mc² Pixel Purse

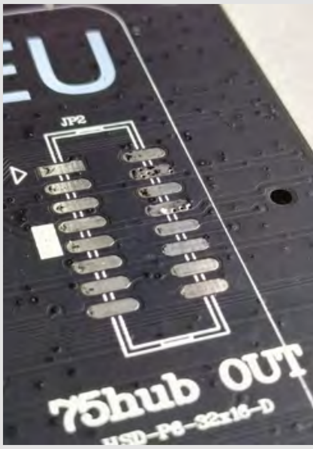
COST

£25 approx.

WHERE

hsmag.cc/NwIWCp

Below
You'll need to solder your own output IDC connector if you want to daisy-chain LED matrices



REVIEW

Grand Central M4 Express

One board, so many inputs

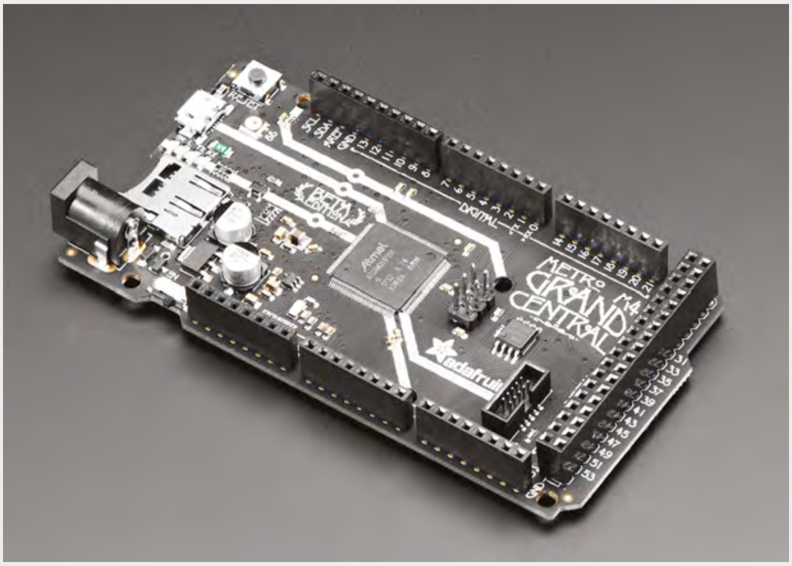
ADAFRUIT ◈ **$37.50** | **adafruit.com**

By **Ben Everard** @ben_everard

There's no shortage of microcontrollers built in the Arduino form factor. However, almost all of these are built in the style of the Uno. The Mega form factor (with its vastly expanded range of IO pins) has seen only one significant insurgent in the last nine years – the Arduino Due, which, despite some advantages, never became popular. However, this has now changed with a new board supporting a large number of IO pins: Adafruit Grand Central M4 Express.

This board houses an impressive 54 digital IO pins and 16 analogue inputs (two of which can be used as analogue out via a 12-bit DAC).

Below ◈ If you Charlieplex all the GPIO pins, you can drive 3782 LEDs. Let's get blinking!

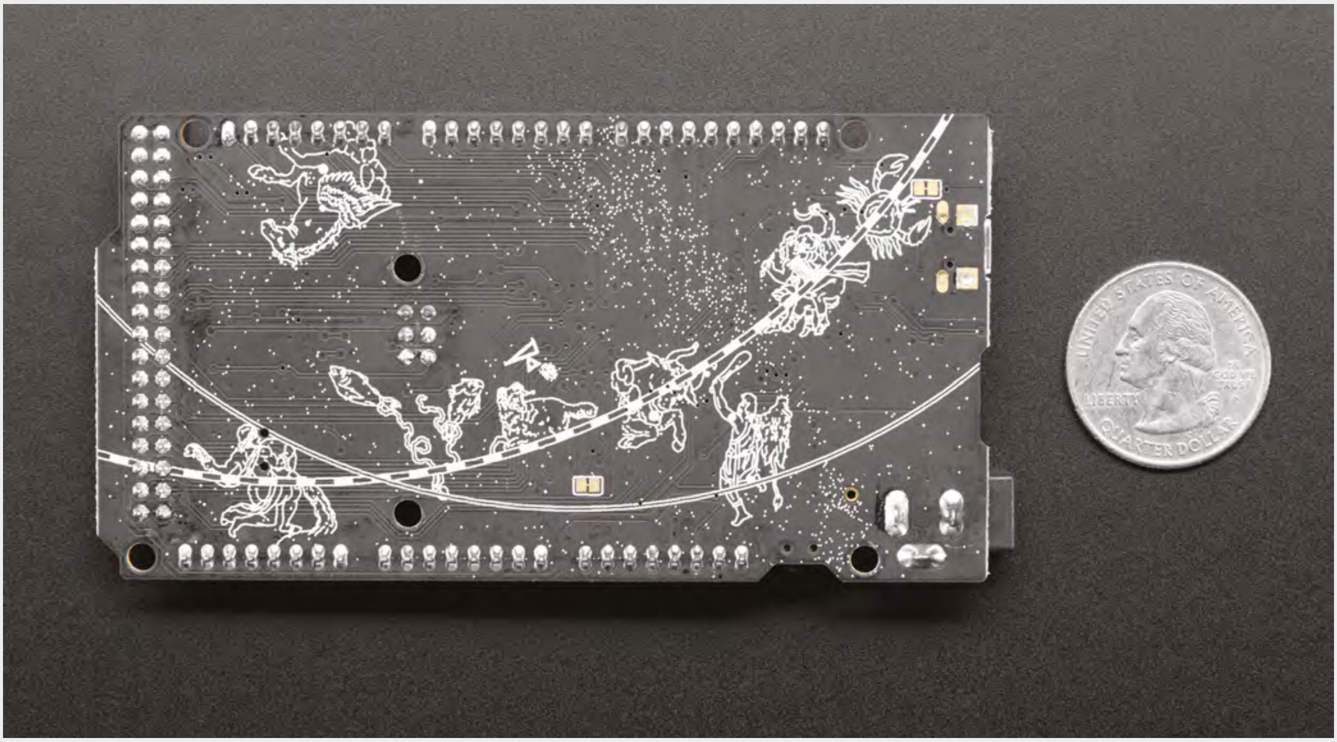


There are a few clues about the processing power of this board in its name. M4 refers to the version of the ARM core on the board, while Express – in Adafruit terminology – means that there's more than 2MB flash space (there's actually 8MB). You can fit a lot in 8MB, but if that's not enough, there's also a microSD card slot, so you can pile (almost) as much data as you like into storage.

The M4 core runs as 120MHz, and has both hardware DSP (digital signal processing) and floating point support. It's a little hard to compare the speed of different microcontrollers because there are a lot of differences in the underlying silicon, as well as the speed it runs at. Floating point can be really slow on some microcontrollers, so the speed-up may be much faster than the numbers alone may suggest.

To test how much faster, we compared this board to a Circuit Playground Express, which has an M0 core running at 48MHz (this is a fairly quick microcontroller by many standards) without a floating point unit. Using the Arduino IDE, we programmed this to perform a million integer multiplications and a million floating point multiplications. On the CPX, this took 189 milliseconds for integer, and 8308 milliseconds for floating point.

On the Grand Central, the integer operations took 67 milliseconds – which is about in line with the expected speed-up, as the core is 2.5 times faster and slightly more powerful. The floating point operations took 75 milliseconds – only slightly slower than the integer operations. As well as floating point, the M4 cores have hardware support for integer divide, with a similar speed-up of about 40 times. This speed means that the Grand Central



Above The silkscreen image is taken from the ceiling of Grand Central Station in New York

can be pushed into areas that many microcontrollers just can't cope with – such as audio manipulation, and calculating complex LED patterns.

The DSP can provide a huge speed boost, however, using it isn't easy. Unless you're interested in diving down into the minutiae of compiler optimisation, it's probably only going to be relevant if you're using libraries that support it. The most popular example of this is the Audio Library originally designed for the Teensy (the Teensy 3.x also use an M4 processor). As M4 processors become more widely used, more libraries supporting the DSP are likely to be created.

> **The M4 core runs as 120MHz,** and has both hardware DSP and floating point support

The form factor – as we mentioned at the start – is based on the classic Arduino Mega, which is an extension to the Arduino Uno, and this means that there are a range of shields already available. Like most modern microcontrollers, the Grand Central is a 3V board, so you need to make sure that any shields are compatible with this voltage.

The Grand Central has a huge amount packed into it, but all microcontrollers are about compromise – you simply can't make a board that has it all, especially when 'all' often includes small size and low price. The most obvious compromise on the Grand Central is the lack of any wireless connectivity – there's neither Bluetooth nor WiFi. This doesn't mean you can't use it wirelessly, but you will need extra hardware, which means extra cost and complexity.

A SOFT TOUCH

On the software side of things, both Arduino and CircuitPython are supported, but Adafruit has this to say: "We have a working Arduino board support package, with lots of stuff working, but our primary target for this board is CircuitPython".

While this does sounds a little pessimistic on the Arduino front, it does need to be taken into account compared to Adafruit's usual excellent support. For most uses, the Arduino environment should work as expected, just don't expect loads of libraries and examples targeting the more esoteric features of the board, such as the PCC camera interface.

CircuitPython using Mu requires version 1.0.2 or later to detect the serial connection, and works as expected. There are already official guides for creating a soundboard and a MIDI interface.

The Grand Central M4 Express packs a huge amount onto a microcontroller board. There's enough IO to control almost anything, and the processing power to crunch through the massive amount of data it's capable of bringing in. As the name suggests, it's not the smallest board, but if you've got the room, this is a great brain for IO-hungry projects. □

VERDICT

Inputs, outputs, and processing power galore. This is a great board for complex controls and interfaces.

9/10

EleksDraw

EleksMaker's pen plotter put to the test

ELEKSMAKER ◈ **£115–£140** | **eleksmaker.com**

By **Jo Hinchliffe** @concreted0g

If you're ever wondering if you need a plotter, head to Twitter and take a look at the hashtag #plottertwitter. You'll discover a community creating amazing images on both vintage pen plotters and modern machines.

The EleksDraw, by EleksMaker, is a kit which sits firmly at the budget end of pen plotters (around £115–£140 currently online) and consists of two sets of rails, to form an X and Y axis, with a belt system to move around a pen affixed to a lift mechanism.

The EleksDraw arrives extremely well packaged, and with all parts to complete present. A few extra of each type of bolt and nut are included, which is welcome for the 'I've dropped one and it has disappeared' scenario. Inside the box there is a small printed card, with a link to the EleksMaker Wiki, where the build instructions are to be found. The instructions consist of a series of pictures of each build step, and each picture is annotated with how many nuts and bolts are used in that step, and which size they are. A little tip is to have a ruler present to double-check you have the right length component. For example, there are some 8 mm standoffs and some 10 mm ones, and they are easy to confuse without measuring.

Below ◈
Once assembled and a few tweaks are ironed out, the EleksDraw is capable of very fine and accurate work

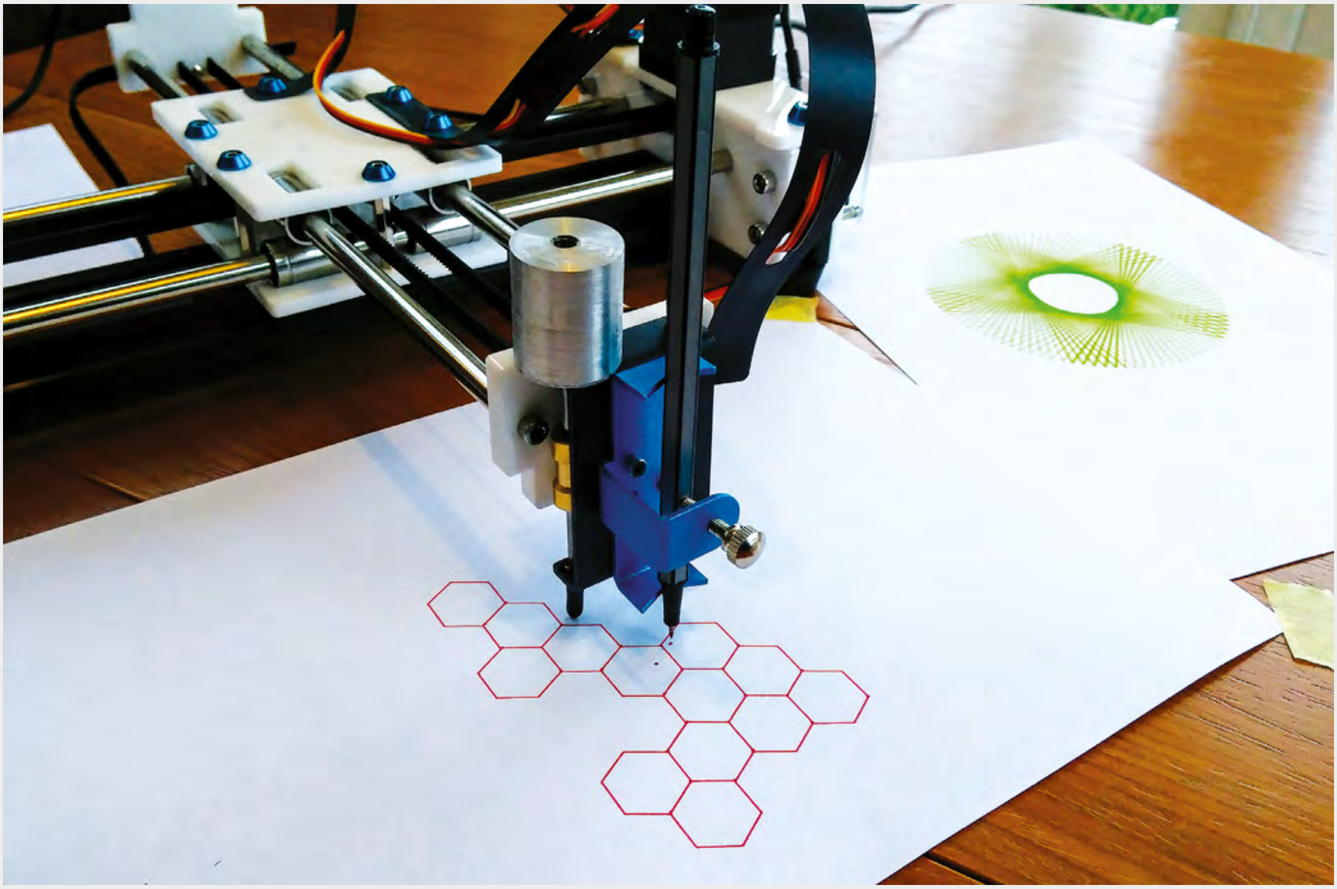
We found we had to go back a few times as we put things together incorrectly. It took us around three hours to build, and there were a few areas we found more tricky, and a couple of problems we had to research. The EleksDraw forum (**hsmag.cc/hdtDcK**) has advice from other people who have been through the same process.

> "Connecting the electronics and motors was an easy plug-and-play affair"

Getting the belt tension on this machine takes a little time. As the belt is a toothed GT2 type, it's easy for it to leave some slack in just one section of the belt, and you only realise when you move the axis by hand. The tension needed to be tighter than we first imagined but, once it was correct, the clamp that traps the belt on the rear of the machine locks it well, and it has required no further adjustment in use.

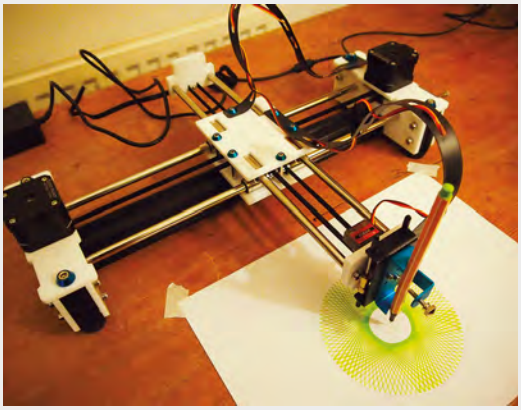
MOTORING ALONG

Connecting the electronics and motors was an easy plug-and-play affair. The next step was to download and install the software and drivers. The link to a folder of Windows software is on the instruction Wiki page. After installing the driver, instructions are given to select which of the EleksDraw machines you have, as the 'EleksCam' software is built to control



the whole range of EleksMaker products. We then turn on the machine and are nearly ready to draw something. The software is reasonably intuitive. There is a 'jog' control, allowing you to move the pen plotting head and a set home button, so you can zero the machine when it's in the desired start position.

One thing to note is that this machine has no end stop switches, so it is possible to crash the machine into its endpoints. As it is belt-driven, the belt will eventually slip, meaning that it doesn't damage itself much in a crash, but it's worth being aware of as it will ruin the image it is working on. To check the area the plotter will operate in, there is a preview button which moves the head, with the pen lifted, to show the area it will cover for the design currently loaded. Pressing the 'Laser ON/OFF' button in the software should lift and release the pen-lifting assembly, but this revealed that the assembly was far too tight to actually fall under its own weight. We stripped it down and rebuilt the assembly many times, trying to make it as free as possible. In the end, we used some fine wire wool to polish the metal slide rods, using some fine oil to eventually get them free enough. It's also possible to hang a couple of nuts or washers onto the assembly to add a little weight to help push it back down to the page when the servo releases it.



The EleksDraw will deal with a range of file formats, including SVG, which is a great vector format widely used by Inkscape users. We've thrown a range of files and vectors at this machine and it is really good and accurate and capable of brilliant work. We have found, however, that it erroneously scales vector images – we're exploring workarounds and solutions in the community and discussion forums. It's an addictive machine: there is something brilliant and compelling about watching a pen draw an image under robot control! □

Above
The EleksMaker EleksDraw, having plotted some generative vector images with a fine-tipped felt pen

VERDICT

Pen assembly frustrating to get correct, software and instructions a little vague, but a great machine capable of producing really high-quality work.

8/10

Digirule2

A computer embedded in a ruler

DIGIRULE2 ◆ From $15.95 | bradsprojects.com

By Gareth Halfacree @ghalfacree

In the earliest days of computing, ‘programming’ was a very physical job, involving the connecting and disconnecting of thousands of individual wires. Later, these wires were replaced with toggle switches, and later still by offline storage devices like punch-cards, paper tape, magnetic tapes, and eventually disks.

In the years since the launch of the Altair 8800, an imposing desktop computer which was programmed exclusively through front-mounted toggle switches, the focus of computing has been to distance the user from the lowest level of how a computer works – the zeroes and ones of binary code. Without that distance, modern computing – where memory is measured in the billions of bits and storage well into the trillions – would simply be too much to manage.

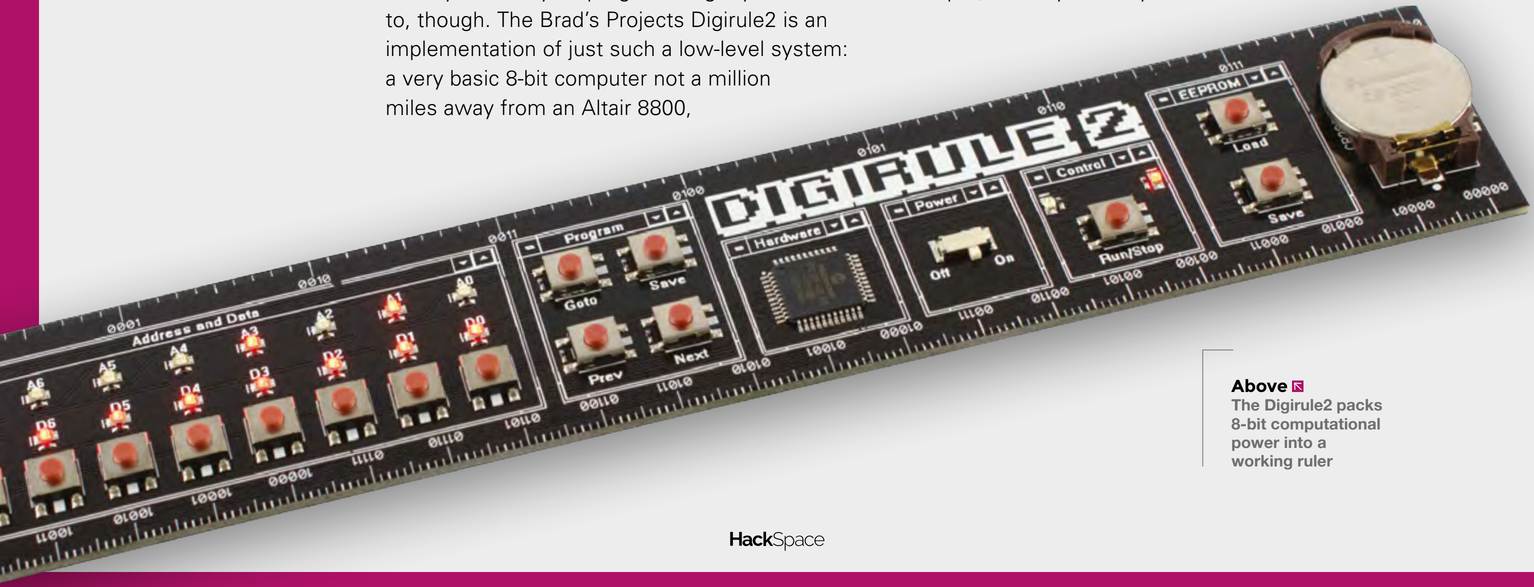
That’s not to say you can’t go back to the good old days of bit-by-bit programming if you want to, though. The Brad’s Projects Digirule2 is an implementation of just such a low-level system: a very basic 8-bit computer not a million miles away from an Altair 8800, but instead of taking up a chunk of desk space, it slips neatly into a pocket.

DRAWING A LINE

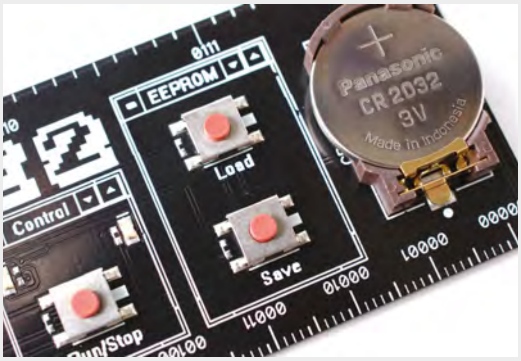
The Digirule2, in fact, is presented in the shape of a ruler, complete with inches and millimetres marked in binary on its longest edges – though the fact that these markings start proud of the edge of the device means it’s not the clearest ruler to use.

Measuring isn’t the Digirule2’s main function, though. Despite its surprising size, it’s a fully-functional microcomputer – complete with its own built-in display, taking the form of a series of LEDs to the device’s left-hand side.

Inserting a button-cell battery – which only clips into place, so it’s not something to give to young children unsupervised – and flicking the power switch springs the Digirule2 into life. Its LEDs will light to show that it’s ready to receive input, which you do by



Above The Digirule2 packs 8-bit computational power into a working ruler



inserting binary bits – the zeroes and ones that are the only things a binary computer understands – into memory using the push-button switches. Each switch corresponds to one bit of an 8-byte memory address; a program can take up 256 bytes of memory in total, and there are eight slots for you to save programs into for later recall.

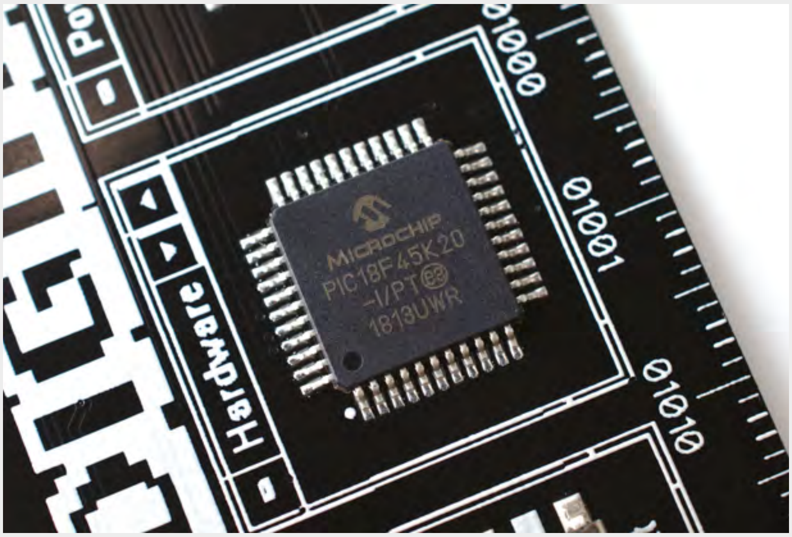
That's not a lot of memory: Minecraft, for instance, takes up a minimum of 180MB – 188,743,680 bytes – which gives you an idea of just how limited the resources of early personal computers were.

With limited resources, you need unlimited ingenuity. Using the 33 instructions printed on the back of the Digirule2, and repeated with full explanations and usage examples in the downloadable manual, it's possible to write some surprisingly functional programs. The Digirule2 comes with a selection preloaded, including a reaction-testing game called Kill the Bit, and a persistence of vision (POV) trick which draws smiley faces as you wave it around the air in a darkened room.

The Digirule2 isn't likely to replace a desktop, laptop, or Raspberry Pi anytime soon, but it's not trying to. The device exists as a means of easily playing around with low-level computing concepts, deliberately using limited specifications to make the user think about how something can be achieved in as little space as possible.

GETTING STARTED

It's here that the Digirule2 excels. The downloadable documentation is extremely detailed, and an update released earlier this year has corrected some minor typographical errors and other small mistakes. Better still, everything is open source: the downloads included on the Brad's Projects website include not only the manual, a spreadsheet which allows you to convert your programs into the binary code you need to enter on the device, and other tools and utilities, but the design files and source code for the Digirule2. With these, it's possible to build your own – though soldering the small surface-mount components is tricky without practice.



With its low cost, accessibility, and quality documentation, it's easy to see the Digirule2 finding a place in classrooms throughout the world. Its use for education comes with a caveat, however: while the Digirule2 teaches the core concepts well, its custom instruction set doesn't directly translate to anything in the realm of modern computing – meaning that anyone wanting to move on to a device with a bit more power, up to and including a modern PC, will need to learn a whole new instruction set or abandon low-level programming in favour of a high-level platform-agnostic language like Python, or C.

Left
Built-in memory stores up to eight individual programs for later recall

Above
The heart of the Digirule2 is a small Microchip PIC microcontroller

> " **Using the 33 instructions printed on the back of the Digirule2,** it's possible to write some surprisingly functional programs "

Outside education, the Digirule2 still has value. Its small size, and ability to function as a ruler for drawing straight lines or measuring them, means it's the perfect desk toy, hiding away until you feel the need to impress someone with the preprogrammed Larson scanner example, Police Flasher animation, Target Practice game, or software of your own devising.

At just shy of $16 (around £12, excluding VAT), the Digirule2 is worth the cost of ownership just as a curiosity, but if you use it for education, either to teach yourself or to teach others the concepts behind low-level computing, it becomes an absolute bargain. □

VERDICT

The Digirule2 is a great device for anyone looking to learn low-level computing.

9/10

Collectable cards

Games Consoles and Home Computers collectable cards

COMPUTING HISTORY ◆ £10 PER SET | **computinghistory.org.uk**

By **Gareth Halfacree** @ghalfacree

The Centre for Computing History, the educational charity behind the eponymous hands-on computing museum in Cambridge, wants to be in your pocket. Specifically, it wants you to pick up a series of collectable cards it has developed, showcasing some of the devices you'll find in its halls.

Originally developed by 8bitkick and launched via the Kickstarter crowdfunding site, the Home Computers 1974–1990 collectable card pack took the classic statistics-comparison game, popularised by the Top Trumps brand and found in schoolyards across the world since the late 1970s, and applied it to vintage computing.

Each card features a single home computer, from the Acorn A3000 to the Texas Instruments TI-99/4A – with a bonus, 'joker' card based on the rather more modern Raspberry Pi – with a full-colour picture and a series of statistics ranging from CPU type and speed to memory quantity, display resolution, and colour palette, as well as the year of release. Any two cards can be compared and, when shuffled and dealt, you've got a game on your hands.

8bitkick's card set soon became an official product of the Centre for Computing History, and the two have continued their collaboration with the launch of a second set: Games Consoles. Like their predecessors, the cards pull in high-quality imagery and statistics for classic consoles from Atari's Home Pong to the less well-known Watara Supervision.

It's here, though, that a trick has been missed: while some of the statistics, such as CPU speed, launch year, and memory quantity, are shared between the two decks, others are not. The Games Consoles deck introduces new statistics including units sold, game library size, launch price – which switches currencies throughout, keeping players on their toes – and 'cool factor', a divisive and entirely subjective rating sure to be the cause of more than a few arguments.

Not being able to shuffle the two decks together to build a larger deck is a missed opportunity, but one that doesn't take too much from what 8bitkick and the Centre for Computing History have built. The imagery – all Creative Commons licensed, and in the Games Consoles deck the exclusive work of *The Game Console* author Evan Amos – is of high quality, and simply rifling through each deck is sure to bring back memories.

> **Simply rifling through each deck** is sure to bring back memories

The packaging, too, is enough to trigger a near-fatal overdose of nostalgia: each deck comes in a plastic clamshell designed to mimic that of a cassette tape, though without the pins that held the tape in place, while the liner includes instructions on play.

At £10 a deck, the cards make official Top Trumps sets seem cheap – but knowing the money's going to charity makes it a lot easier to swallow. □

Above ◆
Each card picks out comparable statistics, alongside high-quality photography

Above ◪
The cassette-style packaging is a perfect choice for these collectable cards

VERDICT

It's a high price for a simple 'trumps' game, but all for a good cause.

8/10

Biophilia

Edward O. Wilson ◆ £20.95 | hup.harvard.edu

By **Richard Smedley** @RichardSmedley

"Go to the ant, thou sluggard; consider her ways, and be wise," runs the proverb, and whilst T. H. White reacted in horror to their rigid division of labour, biologist E. O. Wilson found a world of Darwinian wonder as evolved specialisations allow the ant to succeed across the globe.

Wilson speaks with passion and insight on the connections the biologist makes to analyse life at every level: geological time, DNA, the programmed behaviour of individuals, and how artistic synthesis balances out the reductionist accusations levelled at science since Tennyson. Indeed, scientists seek out theories which are elegant, beautiful even.

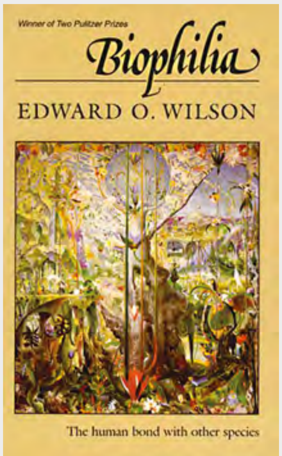
But *Biophilia*, the book, is more than musings on science, life, and the relationship between science and the humanities; it's also the first work in which Wilson began to put forward the theory of biophilia. First coined by Erich Fromm in *The Anatomy of Human Destructiveness* (1973), the biophilia hypothesis describes the connections that human beings subconsciously seek with the rest of life, biologically rooted affiliations with all living things.

"We are in the fullest sense a biological species and will find little meaning apart from the remainder of life," says Wilson. An idea which struggled to find a wide audience when this book was first published 35 years ago, when the mainstream of science still regarded human beings as something very much apart from nature, and the idea that people like looking at nature was seen as a simple matter of aesthetics.

Wilson lays out the cumulative evidence for why the landscapes every culture seeks out are so remarkably similar to those of our ancestral savannah home. All other mobile animals, from colon bacteria to newborn kangaroos, will navigate to the right environment. Recent research has backed up the idea not just that the natural world is good for us, but that it is essential for our mental health.

But this affiliation with life needs better harnessing towards conservation of other species. Ethics just touches the surface of reasons why we should conserve habitats and stop extinctions.

Well written, poetic even, and infused with a love of life – all life. Few science books written half a lifetime ago come over as still fresh and relevant, yet *Biophilia* is a joy to read. □


Winner of Two Pulitzer Prizes
Biophilia
EDWARD O. WILSON
The human bond with other species

VERDICT

A ground-breaking synthesis of science and humanities that is now supported by research.

9/10

next month

issue **#17** ON SALE **21 MARCH**

FEATURING

FIGHT FOR YOUR RIGHT TO REPAIR

ALSO

- → MACHINING
- → VINEGAR
- → ARDUINO
- → MUSIC
- → AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

Protect your LEDs from too much current

$$\text{Ideal resistor} = \frac{\text{supply voltage} - \text{forward voltage}}{\text{intended current}}$$