

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

January 2019

Issue #14

Touch Sensing

Find the best touch board for your projects

NeoPixel Mask

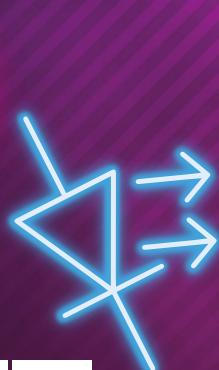
Build an outfit that lights up the room

MAKE MUSIC

*CREATE UNIQUE SOUNDS WITH A
DIY SYNTHESIZER*

Shaping Metal

Get started with milling machines



**DREW
FUSTINI**

Inside OSH Park: How PCBs go from ideas to reality



Jan.2019
Issue #14 £6

ARDUINO PLATES UPCYCLING SEWING

CanaKit Raspberry Pi 3 Ultimate Starter Kit

Model B | 1 GB RAM | 1.2 GHz | Quad-Core CPU



- > Learn to Code
- > Explore Computing
- > Get started with Electronics

KIT INCLUDES RASPBERRY PI 3 AND ...

PREMIUM CASE &
HEAT SINKS



2.5A
POWER ADAPTER



32 GB CLASS 10
MICROSD CARD



PRE-LOADED
WITH OPERATING SYSTEM

USB MICROSD
CARD READER



PREMIUM
HDMI CABLE



QUICK-START GUIDE



GPIO TO BREADBOARD
INTERFACE BOARD



RIBBON CABLE



FULL-SIZE
BREADBOARD



JUMPERS

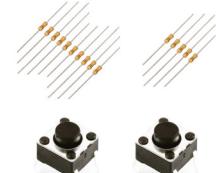


MALE TO MALE &
MALE TO FEMALE

LEDs



RESISTORS &
PUSH-BUTTONS



Available for worldwide shipping at:

WWW.CANAKIT.COM



Raspberry Pi Zero W

Available at CanaKit!





Welcome to HackSpace magazine

Music exists across almost all cultures. It comes in many forms, but some combination of beats and melody are enjoyed by almost everyone on the planet. For many, the creation of music is part of the enjoyment, and there are thousands of different ways of creating tunes.

We're taking a dive into electronic music and looking at how to build a modular synth. This type of instrument is a fantastic opportunity for makers. The basic elements are all simple and easy to understand, but it can get almost endlessly complex as you dive into

the finer points, and you can take your instrument in various directions, depending on your taste in music and circuitry.

Surveying the landscape

Thanks to everyone who took part in our reader survey. It's great to get your input on how to make HackSpace magazine even better. We're still digesting the results, and you may notice a few tweaks as we try to improve. If you've got any further thoughts, please do get in touch with us at hackspace@raspberrypi.org.

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
- [hackspace@raspberrypi.org](https://www.facebook.com/hackspacemag)
- [hackspace@raspberrypi.org](https://www.instagram.com/hackspacemag/)

ONLINE

- hsmag.cc

Available on the
App Store

GET 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

Brian O'Halloran,

Fiacre Muller,

Jun Shena

CONTRIBUTORS

Lucy Rogers, Andrew Huang, Graham Morrison, Cameron Norris, Helen Leigh, Mayank Sharma, Dave Astels, Jo Hinchliffe, Steve Pellard, Michael Jones, Sophy Wong, Les Pounder, Marc de Vinck, Gareth Halfacree, Daniel Hollands, Richard Smedley

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.

HackSpace magazine is published by Raspberry Pi (Trading) Ltd., Station Road, Cambridge, CB1 2JH. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents

06 SPARK

- 06 Top Projects**
You're brilliant and you make gorgeous things
- 16 Objet 3d'art**
3D-printed devices to save time and sanity
- 18 Upcycling**
A new life for old vinyl
- 20 Meet the Maker: Dustyn Roberts**
May her flowers flourish and brassicas never bolt
- 24 Columns**
Learn how to let it go like Elsa in *Frozen*
- 26 Letters**
Tell us what you think of the magazine you're holding!
- 27 Kickstarting**
3D-printed ceramics? Now there's a good idea...
- 28 Hackspace iForge**
Next-level facilities at the University of Sheffield

Cover Feature



34

102

33 LENS

- 34 Build a DIY synth**
Recreate the *Blade Runner* soundtrack with a DIY synth
- 52 How I Made: MINI.MU glove**
What it takes to take an idea from prototype to product
- 58 Safecast**
DIY Geiger counters hard at work in Fukushima
- 64 Interview Drew Fustini**
Inside OSH Park, the home of open hardware
- 72 Improviser's Toolbox Paper plates**
This is now our favourite way to make a drag racer

Tutorial

Get started
with machining



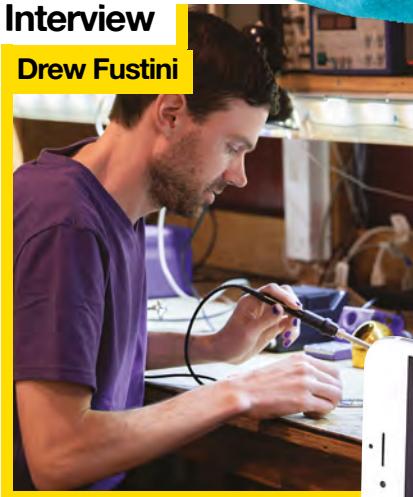
82 How to carve metal
with a milling machine

120



Direct from Shenzhen**Portable sewing****112**

Can you get a new, working sewing machine for under £10?

**18****27****52****Interview****Drew Fustini****64**Open hardware,
in all its purple glory**77****FORGE****78****SoM Electronics 101**

Put transistors to work in a simple oscillator

82**SoM Machining**

Carve metal with power and precision

88**SoM Rotary tools**

Use spinny things to cut just about anything

90**Tutorial Powders**

Alter the state of food to create unique textures

92**Tutorial Remote-controlled lasers**

Control a K40 laser cutter over a network

96**Tutorial Arduino robot**

Add logic and sensors to your homebrew robot

102**Tutorial Wearables**

Use sensors to trigger NeoPixel animations

108**Tutorial Jupyter**

CircuitPython development the interactive way

111**FIELD TEST****16****112 Direct from Shenzhen Sewing machine**
Can you replace a full machine with a hand-held device?**114 Best of Breed**

Touch-friendly boards for interactive circuits

120 Can I Hack It?

Lasers! Sensors! We're having a very hacky Christmas

122 Review GiggleBot

An expandable robotics platform for micro:bit

126 Review Sensything

Just add a sensor and you can monitor anything

128 Review Seeedstudio Air602

A WiFi-enabled development board that costs pennies

129 Book Review Brief Answers to the Big Questions

A collection of posthumous cleverness by Stephen Hawking

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.



ESP Nightlights

By Andy Warburton

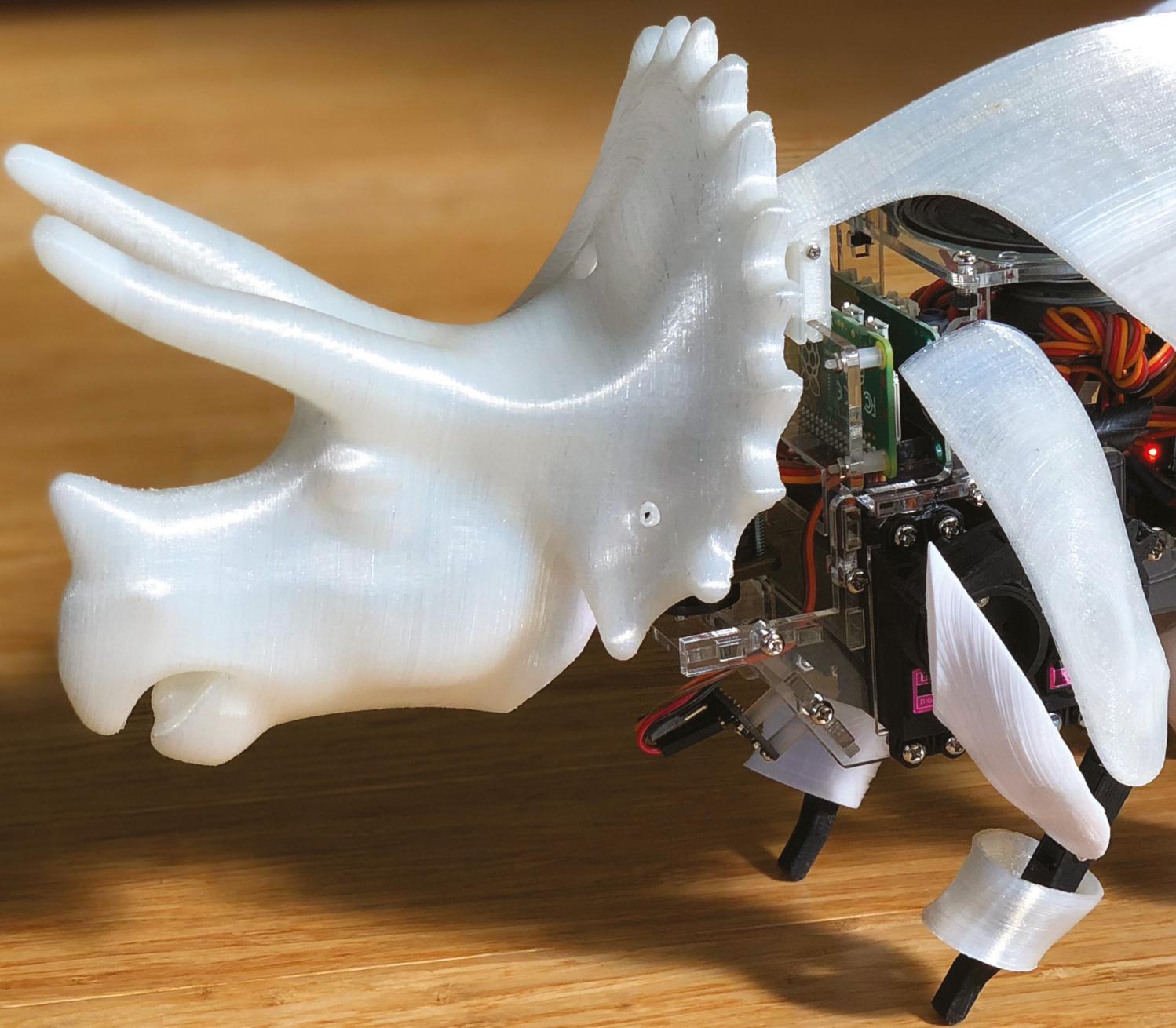
 Andywarburton.co.uk

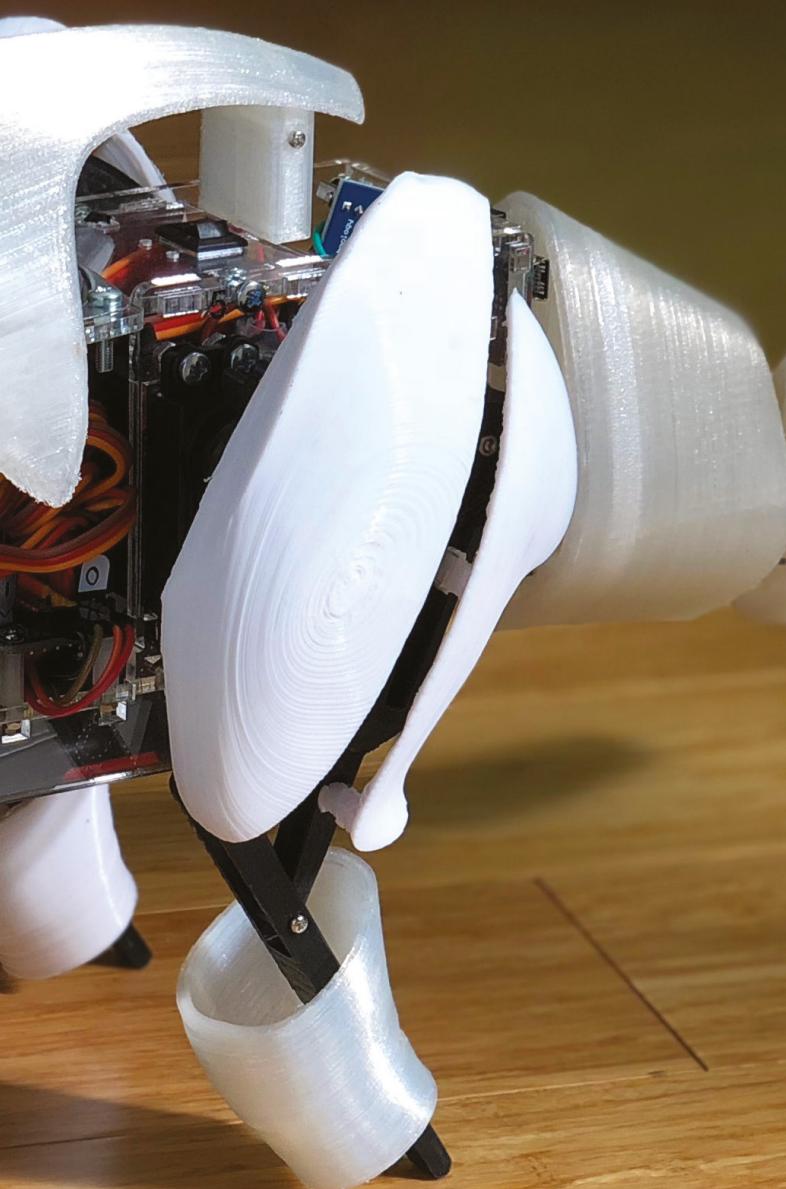
I'm a keen maker and love to solve my daily problems with homemade solutions. If you have kids, you'll know how frustrating it can be to be woken up at 5am when you need to sleep. To combat this, I devised a simple colour system to let them know when they were allowed to get out of bed in the morning. When the lights are red, it's time for bed!

The colours are displayed on this pair of smart nightlights, made using cheap LED strips and ESP8266 WiFi-enabled microcontrollers (parts cost less than £5 per light!). I then used a Python and Flask web server, hosted on a Pi 3B+, to build a quick and simple lightweight API, from which the lights could read the RGB colour values. This enables me to update their light routines without having to re-flash the ESP8266s. Now my kids stay in bed, and I get a couple of hours more precious sleep at the weekend! ☺

Right ☺
We love sleep.
Well played, Andy!







Intellisaurus

By Jacquin Buchanan

@BuchananJacquin

I started this project because I wanted a robot that is easy to assemble, yet looked and behaved a little more like what we see in the movies. I have been a software engineer for 30 years, and a maker since I was child. I am drawn to projects that balance engineering functionality and artistic and design flair. I wanted to make a kit that is simple to assemble, yet has many layers and areas for deep-dive modifications.

This robot has an Arduino to control the direct leg movements of walking, and a Raspberry Pi to give it artificial intelligence, to react to voice input or respond to recognised faces and objects. There is no soldering or programming required to build and play with it. All the software is open source and the white shell 3D-printable parts are available online. You can modify and improve the walking code, give it new AI behaviours, or modify the shell in your favourite CAD program. □

Left □
We know a 35-year-old who misidentified this as a stegosaurus. Don't let the same things happen to your loved ones

Leviathan Axe

By Ty Beringer

 dryadgaming.com

I'm a professional woodworker and I own Dryad Gaming Co., a small workshop that creates hardwood accessories for tabletop games. I have a fully outfitted workshop, but from time to time I like to challenge myself to build something completely different, and with limited tools. This is a sort of creative exercise for me. After playing the fantastic God of War game on PS4, all I could think about was how cool the axe was. I saw a few people build the Leviathan axe on YouTube and Instructables, but they were either out of foam for cosplay, or actual steel axes.

I wanted to do something different and make it out of 100% wood (even the fabric handle wrap is wood). So, pretty much only using a band-saw, belt-sander, Dremel, and pyrography tool (three-quarters of which cost less than \$50), I got started. There were times when using my advanced tools would have been quicker (my laser engraver is a lot faster than my pyrography tool), but I actually find it quite nice to do things slowly and more 'by hand'. Besides, I knew I wanted to show off the end product, and also show that even a very basic woodshop can make cool things! While I do have a professional workshop, this project, like so many others, certainly doesn't need much more than simple tools! □

Right  Everything on this axe, right down to the 'leather' strap, is made of wood





Stargate

By David Jones

 glitch.org.uk

The basic design of the Atlantis gate itself comes from an already existing Milky Way working Stargate model, as the two gates share the same dimensions and overall design aesthetics. With the base of the new model, I wanted to incorporate several recognisable features from Atlantis to make it more interesting than a simple housing for the electronics.

Since there was not enough room to include the full message on the stairs, I have shortened the verse, whilst trying to maintain its meaning:

welcome to those
visiting our home
for the first time
always come in peace
as we strive to go in
peace and you will be
welcome here forever more

Maintaining a wormhole between the Milky Way and Atlantis gates is proving more difficult than first thought. I might need to try it with more distance between them!

When not tinkering with 3D printing or electronic projects, I run an online magazine (kraxon.com) for short stories in the science-fiction and fantasy genres. □

**Right** □

David says he's chuffed with the rear window: "I think the different colours of the RGB LEDs show it just right"



Plywood arcade suitcase

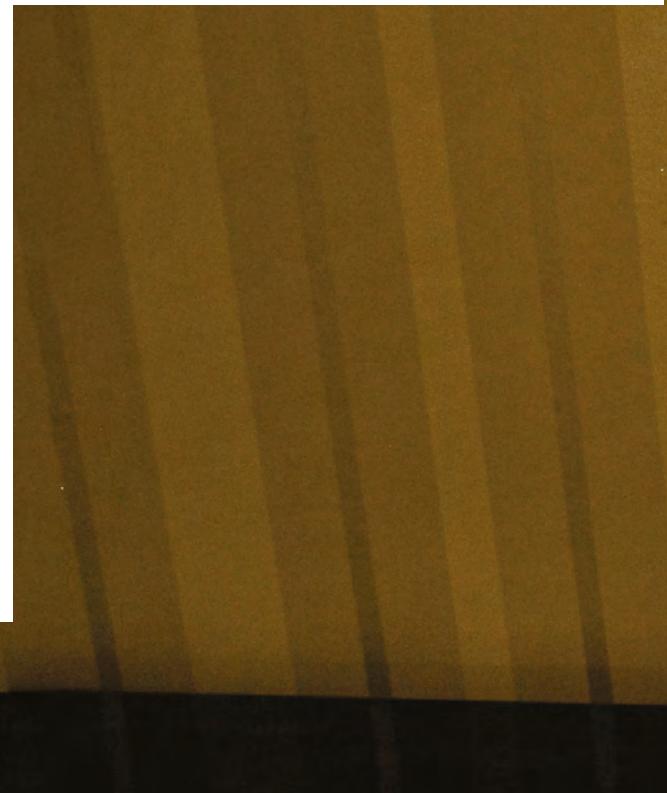
By Henry Lähdeaja

@ hsmag.cc/bZGvTi

I am a 37-year-old (wannabe woodworker) engineer from Finland. I have three daughters (two, three, five years) and a wife.

I've turned our basement into a woodworking and hobbyist place, which makes a good counterbalance place for hectic family life now, when the kids are young. I have previously made two bartop arcades as a 30th birthday present to my twin sisters. It was really fun to do, but I then didn't make one for myself. So, I wanted to build one for myself but smaller. The bartop arcade is quite small, but it's also so big that it's really difficult to hide from my two-year-old daughter who likes all the buttons and joysticks too much.

I also like that suitcase idea: it is really easy to take with you anywhere and the best part is that if you don't know what's inside, it can look just like a normal wooden suitcase. Those amazed faces when opening that suitcase! ☐



Right ☐

Up, Down, Left, Right,
ABC + Start. The
oldies are the besties



Objet 3d'art

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



Head to 3dhubs.com/book
to check out the
#1 3D printing book on Amazon

Kids these days spend a fortune on expensive wireless Apple earbuds that fall out and get lost. Wiser heads still use the older, more reliable technology known as 'wires' to get sound into our ears. Sadly these too have a flaw, turning into tangled spaghetti as soon as you put them in your pocket.

That's not a problem you'll have any more if you print yourself this handy earbud holder designed by Sneakypoo. They've created a simple, robust way to hold your wired earphones in one place, tangle-free, in a case that closes with a satisfying 'clack'. □

↗ hsmag.cc/JKoSJn





Camera gear is expensive, often only working with one specific brand of kit. In contrast, this universal phone holder will keep your camera phone in one place while you're recording, whatever the size of the screen or brand of the device. It's tightened using a gear mechanism to clamp the top jaw down on to the phone, while the bottom jaw stays in place underneath, and the whole thing easily slots into your tripod stand (or on your workbench). Hats off to Jakejake for this design. □

↗ hsmag.cc/BBrSXq

Upcycling: New life to old records

Vinyl records are beautiful, but not all can still be used for music

By Ashley Seager

 seagerstudios.com/art



Inspired by a life-long passion for music, I built a small workshop in the back of my garage in 2013, and have been making music-inspired art and gifts for friends and family ever since.

Exploring new possibilities with vinyl records and vintage sheet music, I specialise in original designs using reclaimed materials that I source locally across Norfolk – metallic fruit bowls, letter racks, guitar pick clocks, and anything that's useful or decorative.

Refining my ideas by connecting with other makers has helped improve the quality of my work in a big way and is something I'd recommend to other upcyclers.

“With meetups, forums, and social media there's no need to go it alone”

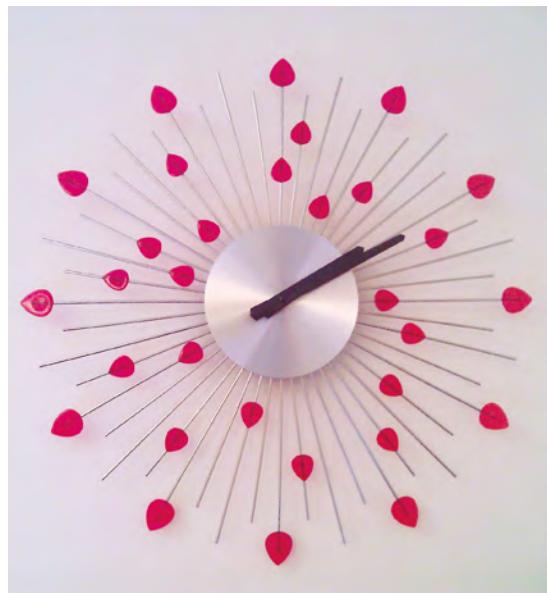
With meetups, forums, and social media, there's no need to go it alone: it's interesting how critics and negative comments can help too. My designs attracted a fair amount of controversy among vinyl enthusiasts before I mentioned that the records are unplayable and destined for landfill. Adding this info into product descriptions has helped vinyl enthusiasts become one of my biggest customer groups.

For 2019, I'm looking forward to experimenting with larger projects like furniture and art installations, growing my Etsy store, and getting in touch with my local makerspace to benefit from their awesome power tools. □

Above You don't need a lot of tools to work with vinyl

Above Right You can do a lot with a record and some heat





Meet The Maker: Dustyn Roberts

Creator of home irrigation system,
Sage Smart Garden



Last summer in Britain was a scorcher, with plants drying up under a red-hot sun. You'll have seen DIY watering systems offering a makerspace answer to this, but even the simplest solutions need a modicum of know-how to put together. Dustyn Roberts is the CEO of Sage Smart Garden, a startup trying to take the smart garden experience to the masses. We spoke to her to find out about how, why, and the importance of not being a jerk when you're running a business (*Apprentice* contestants, take note).

"I just moved to Philly from Delaware and in Delaware I had a system set up in my yard with a drip line for irrigation. But, either I forgot to turn it on or I forgot to turn it off. If you get a timer, you can

“We started talking to landscape designers and green roof companies, and they had the same problems: they wanted to be able to automate irrigation”

set it to come on automatically, but it's still going to water when it's raining, and it's really hard to fine-tune it. You can't give it an extra shot of water at noon on a hot day and things are getting fried because you're at work. So, I started trying to fine-tune a system that would work for me.

"Sage is a system of smart gardening modules that enables you to automate irrigation. The hub [which is a white box] sits inside, next to your router inside your house, and then there are two modules:

the water valve that goes outside fits into a hose or a sprinkler, and the soil moisture sensor sticks into the dirt in your garden and tells you how wet it is.

"So, it allows you to close the loop and, say, 'water my garden if it's dry.' Or, 'water my garden every day at 7am if it's dry, but not if it's going to rain that day.' It can pull in weather data through the API and not water at all if it's going to rain that day."

"At its simplest, it's a remote on-off switch for water; at its most automated, it's doing everything for you. You can pick a tomato on Saturday and not have worried about it while it was growing – which is why I want it, because that's how it came to be."

"We started talking to landscape designers and green roof companies, and they had the same problems. They wanted remote management, they wanted to automate irrigation, and they wanted to use resources efficiently so they didn't have to send people out to monitor the plants, see that they're not dying, and so forth."

"In Philadelphia, there's a makerspace called NextFab. It used to be a machine shop that has a membership model like a gym, so you pay an amount per month, you get to use water-jet cutters and 3D printers, and laser cutters and things, so it's a professional hackerspace. They started a hardware startup accelerator, so we're in their hardware business accelerator programme now, as of two weeks ago, and that's their whole job, to help us be market-ready."

"The plan is in three to six months, you're going to see a Kickstarter or Indiegogo launch, and that's where we're going to get presales from. And, in the meantime, we're going to landscape designer conferences and working with early adopters to help us develop this. →

Right ♦
The original system was a Raspberry Pi hub, Arduinos, and XBee (little radio modules). Now Sage is using a Texas Instruments CC1310



Meet The Maker

REGULAR

Below ♦

From left to right –
Ariel Ramirez,
Chief Plant Officer;
Dustyn Roberts, CEO;
Trevor Stephens, CTO



"We actually launched a Kickstarter about a year and a half ago; we cancelled that after a couple of weeks because, at that time, we weren't ready to go into sales. We were trying to raise some money for prototyping, basically.

"But the rules of Kickstarter were weird: like, you cannot promise the hardware unless you already have a prototype – which makes sense. We want to really do this when we have the thing ready to go.

"If you search with Google 'smart garden' – or go on Instructables or on any website, these DIY content websites, and search for 'Arduino garden' or 'smart garden' – you're going to find everyone who has ever hacked together their own versions of something. But, you still can't buy it anywhere. That's why we decided to turn it from a project into a startup, and hopefully a sustainable model, that we can earn enough money to keep making hardware and not be jerks about it.

"I've been on a lot of business training that talks about 'extracting value from customers', and I don't want that. I don't care what their lifetime value is, or whatever other metric you're using; I want to make enough money to sell cool stuff and use it myself, then enable more people to grow their own foods." □

Above ↑

The Sage Smart Garden system guarantees that your lettuce won't dry out when you go away on holiday



Outsourcing

When should you do it yourself and when should you enlist help?



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

One of the advantages about being a maker is the willingness to try everything. One of the disadvantages of being a maker is the willingness to try everything.

Accounts? Marketing? Fixing the plumbing? Decorating? Changing the oil in the car? I've done all these things, some more successfully than others.

And there's only one of me. Why am I doing the accounts (which I dislike), when I could be doing something I do like – such as making?

In my early career the answer was usually 'money'. I couldn't afford to pay someone to do something I could do. I was time-rich and cash-poor.

As other demands claimed my time, I realised I couldn't do it all – there were two options:

- **Outsourcing** – giving tasks to individuals or companies outside of your business
- **Delegating** – giving tasks to other people within your company

When you're the only person in the company, the choice is easy: I would have to outsource some things.

Richard Branson said: "Everything in your business can be outsourced – if you're not emotionally attached to doing it all."

I was not emotionally attached to doing the accounts – although I did still want to understand them. I outsourced them. When they came back, I found mistakes and things not claimed for that should have been. My accountant's response, when I politely asked him to explain or where I had gone wrong in my calculations? "Wrong degree maybe?"

I took the accounts back in-house and did them myself. I would rather spend days swearing at a spreadsheet, than pay someone to do them AND THEN spend days swearing at their spreadsheets.

**"Fixing the plumbing?
Decorating? Changing
the oil in the car? I've
done all these things"**

This time, I have not been disappointed. They explain everything clearly, and the lack of stress, grumpiness, procrastination, and worry that came at the end of my tax year has gone. I am very happy to pay them to do the job.

So yes, Richard Branson is right – "Everything in your business can be outsourced," but finding the correct person or company requires careful consideration – and that takes time. Personal recommendations, online reviews, availability, ethics, and cost are things I look for.

Engaging the wrong firm can end in more wasted time and expense than doing it yourself. But, when it works well – that's when you and your business can grow. □

Picking an SoC

Find the right silicon for your business



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.

One question I’m often asked is how to pick an embedded system on a chip (SoC) for a given project. It’s a tricky multidimensional optimisation problem, trading off cost, features, availability, and performance. Evaluating an embedded SoC based solely upon its datasheet and price misses one of the most significant hidden costs: firmware. Peripherals are useless without stable drivers and/or good documentation, and a poor compiler can waste significant amounts of memory and performance.

Firmware support can be a significant portion of a chip’s R&D budget. This is why SoCs with good firmware support sometimes seem underwhelming in performance

and price: it can take many man-years of effort to mainline a complex SoC into the Linux kernel. At the other extreme, some SoC vendors basically cram as many features as they can into a chip, hoping to lure a big customer. Only after a significant volume order has been inked does the company start throwing resources at fleshing out the firmware. By economising on firmware developers, these chips can sell at rock-bottom price, and the documentation is strictly NDA. The NDA is important because, as firmware is developed, bugs are eventually identified and the NDA allows

Evaluating an SoC based solely upon its datasheet and price misses significant hidden costs

the chip vendor to hide the existence of the bugs from new customers, investors, and potential legal actions. This practice of taping out silicon and just-in-time developing the firmware, based on a high-volume customer’s needs, explains why, in general, it’s very hard for small-time makers and startups to access some of the more interesting SoCs.

Thus, my usual advice to folks looking to pick an SoC is to first code up a simple demo representative of the workload and I/O footprint, and then port the demo to development boards provided by the vendor. The difficulty of getting the

toolchain up and running, as well as the general maturity of the firmware, will become painfully clear within a couple days of effort. Although repeating this process for a few SoCs will burn

a couple of weeks of time, it can save months of fighting with buggy silicon, fixing drivers, and hand-optimising code.

The other important aspect to check is the availability of the specific part number you’ve chosen. Distribution support will be important, especially if your initial builds are smaller than 10 000 units. A single 300 mm wafer might yield several thousand Cortex-M series SoCs, and a fabrication plant typically processes wafers in lots of 25. So, a few thousand units may be a big deal to a startup, but it’s tiny for a silicon vendor. □

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

HACKADAY PRIZE

It was really inspiring to read about the projects winning the Hackaday prize. Such great projects, and open hardware, so I can see how they're made! I can't wait to see what happens in next year's prize.

Steven

Walsall

Ben says: Glad you enjoyed it!

As well as their main prize, there are also smaller contests that run through the year. At the moment, there's a Circuit Sculpture Contest where people show off their artistic creations: hsmag.cc/RPWNqH



WONDERS OF WOOD

As a lifelong woodworker, I was surprised to read the interview with Wilbur Pan. I guess I've just assumed that everyone did it the same way, but it turns out I was wrong. I've been reading up on Asian techniques, and look forward to giving some of them a go.

Rich

London

Ben says: There's always more than one way to skin a grape, and it's good to take a bit of time to learn about the different options. Even though you might be happy with the technique you're using, learning more about the others may help you understand a little more about the craft.

SUPER SPY

Great magazine guys! Sophy Wong's tutorial on NFC cuff-links was really inspiring. I'm not sure I need to sneak data into an ambassador's reception, but embedding electronics in resin seems like a great idea.

Paul

Minnesota

Ben says: Embedding in resin makes circuits tough and waterproof, as well as aesthetically pleasing. Pouring resin is a great skill to add to your maker arsenal.





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.

CROWDFUNDING NOW

CERAMBOT

Automatic pottery making

From \$69 | hsmag.cc/qglKer | Delivery: April 2019

If you think of a 3D printer, you probably think of a printer head heating up plastic filament and depositing it at a particular place to cool and harden. However, these printers don't have to use plastics, and there are other materials that can also work in a similar way, including ceramics. The CERAMBOT is a 3D printing system that uses clay to build up your models.

There are two types of print head available: the Air version, which needs compressed air to run (\$69 for the early bird offer); and the Pro version (\$129 for the early bird), which just needs electricity. CERAMBOT claims that this should work with any 3D printer, but we've not been able to test one out.

The CERAMBOT printer (\$289 for the Air version, or \$329 for the Pro version) is a delta printer which can print objects 285 mm tall and 170 mm wide.

Printers don't have to use plastics, and there are other materials that can also work in a similar way, including ceramics

One potential issue for homemakers is that most clays have to be fired, and this requires a kiln. You can get air-drying clay (which CERAMBOT say will work with the printer), however this has different physical properties to ceramic clay. Make sure you've got access to the other facilities you need to make the things you want to make before signing up for this.

Opening up consumer-level 3D printing to new types of materials will be a fantastic thing, if CERAMBOT can deliver on its promises. We're looking forward to being able to try this hardware out. □



Above ♦
The print area is perfect for thin, tall objects like vases

Left ♦
The extruder head is open source, and available for download on Thingiverse

Space of the month: iForge



Nick Boone

hsmag.cc/ZRBbMA

S

heffield is a city of makers. It's synonymous with high-tech, high-quality manufacturing, so much so that Siemens, Boeing, McLaren, and hundreds of other technology and manufacturing companies have made it their base (including maker favourites Pimoroni).

The University of Sheffield is therefore the logical place for the first student-led university makerspace in the UK, in the Department of Engineering's brand spanking new Diamond building. We spoke to Nick Boone, a PhD researcher at the university, about what makes the space so special.

"Last year the iForge had a capacity of about 20; we had two lasers, a CNC machine, a couple of pillar drills, a couple of band-saws, and some hand tools. And then, over the summer, we've expanded to the whole other side of the room, which has increased the capacity to 45."

"The expansion is entirely due to student demand: we were turning people away because we didn't have space for them. We're now running two degree modules in here – their design and making bits – that students have to come and make in the iForge. It's all student-led on the making side of it. There have been big queues outside and we're still hitting capacity limits, so we're hoping to grow again maybe next summer."

"That's expanded the equipment we've got, so we're now up to three lasers in here and more



space to do things. One third of the area is our clean zone. That's a no-goggles area. We're soon to have more PCs and a CAD suite going on here, and then we've just got a water jet cutter, which is still in the process of being commissioned.

"We've been donated bits of equipment from the university, grant money, and things that we've been awarded through applications."

"The main funding to start it all up was from the Reece Foundation. We got £50 000 from them last year, and that helped kick-start the space, buying equipment, and that sort of thing. They're a charity that's basically for increasing engineering awareness, getting more people involved in it, especially at school and university level, making it more interesting and accessible."

The expansion is entirely

due to student demand:

we were turning people
away because we didn't
have space for them

We've also had investment from alumni through the university who've come in and said that they want to give money to something for the iForge – that's where the water jet has come from. We've got a suite of CNC tools planned that are also coming out of the budget, because they told us that they wanted the money spending on machines: stuff that people can use, make with. They didn't want it spent on admin or anything, but wanted something tangible.

And then we've got a 3D print lab as well. 3D printing is a big deal for us because it's the cool new technology. We do have to be a bit careful with it, because a lot of people aren't aware of the limitations of what you should and shouldn't 3D-print. There are a lot of people who come wanting to 3D-print boxes, for example, because →



Above iForge reps go out to local schools as part of the university's outreach work to encourage interest in engineering

QUICK

Space of the month

REGULAR



Above ♦

Ex-students can apply to the uni to get an access card to use the library and the iForge within hours

they've heard that that's something you can do. We gently steer them towards the laser cutter for that.

"The process we've got this year is: people can come in and submit prints, a rep will look at them with them, slice it with them, so the student gets to learn about slicers and settings and all those sorts of things. It all comes with practice and time. We don't want ourselves being a print bureau, where we get files thrown at us and clients expect them back the next day; we want users involved in the process, and to understand that it's not just a question of pressing print and having something pop out the other end.

STUDENT RUN

"We're totally run by students. 12 midday until 8 o'clock every weekday. It took a while to get everyone on side, particularly on the health and safety side. But our accident record has been great. We're very conscious of training on the machines. The users get sick of having to wear goggles, but that's essential: if you're in there gluing something but the person next to you is on the band-saw, you're still in danger of getting hit. People sometimes don't understand that.

"The way we run our training is there's a general induction on health and safety that everybody reads, but after that, training is done on a piece of kit basis, so if you need to come in and use the laser cutter, the only piece of training you need to do is the general training and the laser cutter. That's all done online. When people sign in to the space, the system checks their training; it will only light certain boxes up, so if they, say, want to come in and use

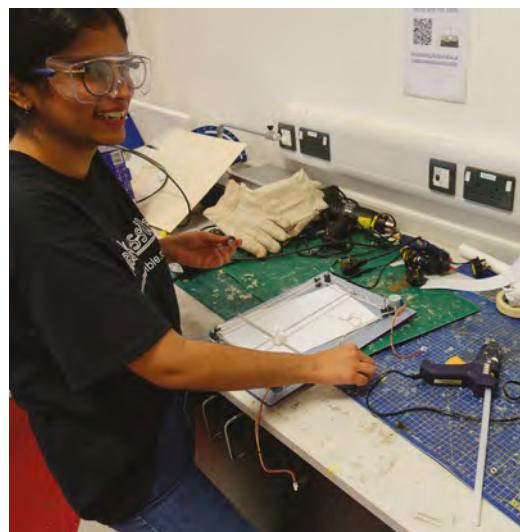
the router, the person behind the desk will be able to see if they've got the training to be able to do so.

"We're planning on implementing that system on the machines in future, so you'll have to scan or swipe a card attached to the machine, and it will power up the machine if you're safe to use it. We didn't find it too bad last year with a capacity of 20, but now we've got a bigger space and more people in here, it's a bit harder to remember what people have got (though most people are pretty honest).

"Over the past week, we've been at capacity from opening to closing, so there have been 45 people in here from 12 pm to 8 pm, with one in, one out. We're almost at 2500 registered users, which is absolutely great as far as we're concerned. People are using the space, people are coming in here wanting to do things.

"We've got our own materials stores. Every user gets £10 free credit to spend on materials with us; the idea is to limit the barriers to making as much as possible. All 3D printing is free anyway, but if you're buying acrylic, sheet metal, plywood from us, we help you get going and then buy from us at small sizes. If someone comes in with an idea for something they want to make, we want the barriers to be as low as possible.

We're still trying to get the makerspace spirit across; even though we're a university, we don't want to be the university workshop. People might be coming in here to do university projects, but that's not just what we're for; the person who's here for a module has just as much right to be here as the person who's making a Christmas decoration for their mum. □



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 ♦
The iForge has two Ultimaker Plus printers, four Prusa MK3s, and three Formlabs resin printers

NEW

Wireframe

Join us as we lift the lid
on video games



Visit wfmag.cc to learn more

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG
52



HOW I MADE: **MUSICAL GLOVE**

Discover what it takes to turn a make into a commercial project

PG
58

RADIATION MAPPING

A group of hackers helping Japan stay safe in the aftermath of disaster

PG
64



HACKSPACE MEET **DREW FUSTINI**

Find out what makes perfect, purple PCBs

PG
34

BUILD A **SYNTH**

GRAB AN ARDUINO AND A SOLDERING IRON. IT'S TIME TO MAKE MUSIC

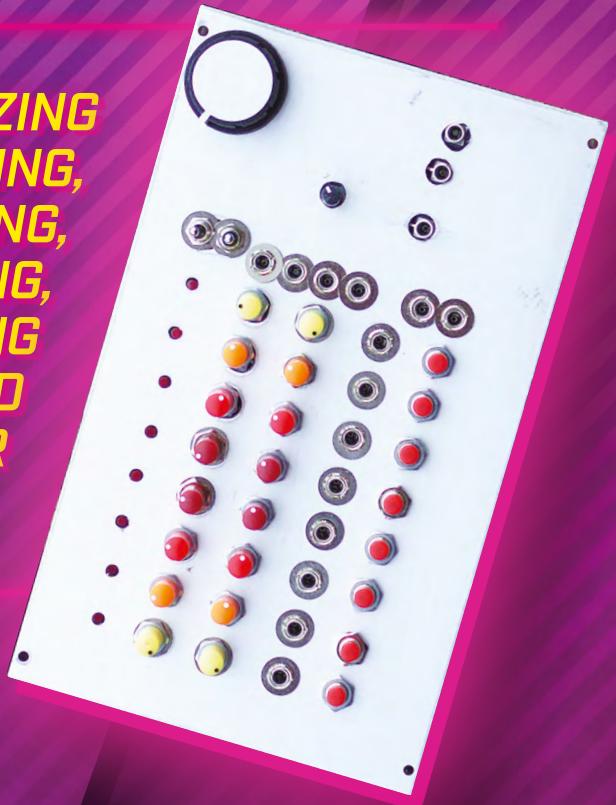
PG
72

PAPER **PLATES**

Turn your next picnic into a makefest

BUILD A SYNTH

**BUILD YOUR OWN AMAZING
SOUND-GENERATING,
VOLTAGE-SEQUENCING,
GATE-TRIGGERING,
KEYBOARD-PLAYING
SYNTHESIZER AND
SEQUENCER**





here is nothing in nature that sounds like a classic synthesizer. From Delia Derbyshire's incredible work at the BBC's Radiophonic Workshop, adding synthesizer sci-fi to

the original *Doctor Who* theme, through to the modern minimalism of synthesisist and composer Kaitlyn Aurelia Smith; it's a synthesizer's harmonic avarice that, quite audibly, sets the tone.

Much like computers, sound-making machines started off as monolithic analogue machinery that turned and crackled and sparked themselves into life. Though not huge, one of these early instruments was the theremin, a wooden box with alien antennas that modulated a tone when a performer moved their hand, like a Jedi conducting a disturbance in the force. And it was Robert Moog, building replicas of the theremin, who helped define what the modern synthesizer was and, most importantly, what it could sound like.

After bland digital synthesizers took over in the late 1980s and 1990s, analogue synthesis with an experimental edge is back, and stronger than ever before. The big manufacturers, like Moog, Korg, and Roland,

are building and selling experimental kit, and there's a growing global community of makers and hackers building and selling their own components, kits, and code, contributing to a new age of DIY sound design and experimentation. And it's this world we're going to visit over the following pages – helping you to build your own sequencer and sound generator, and

SIGNAL PATH

Synth sounds start with an oscillator, because it's the oscillator that generates the raw initial audio. This sound then passes through whatever other stages a synthesizer offers before arriving at the final output.

A waveform, like a mathematical function, has a specific shape, and the most commonly used shape for a synthesizer is a sawtooth.

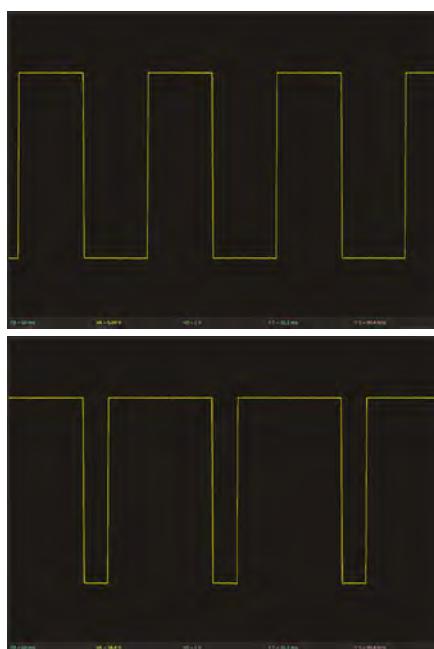
Analogue synthesis with an experimental edge is back, and stronger than ever before

hopefully, causing you to get hooked into this brave new world of sound design, drone, and tonality. Or even just a few Brian Eno-like earworms. But before we get ahead of ourselves, we need to briefly cover exactly what these wonderful machines are, and how they're formed.

A sawtooth, predictably, looks like the jagged teeth on a saw. It's perfect for audio because the hard edges in the waveform result in lots of harmonics. A sawtooth is basically the sound equivalent to a chunk of malleable clay, ready to be moulded and reduced into an infinite number of other sounds. Other common synthesizer waveforms – including square, triangle, and sine – aren't quite so flexible, but they can add to the timbre. But it's the next stage that adds the character: the filter. The filter cuts out some of the frequencies in the harmonically rich oscillator output, and it's the filter that often gives a synthesizer its definitive sound.

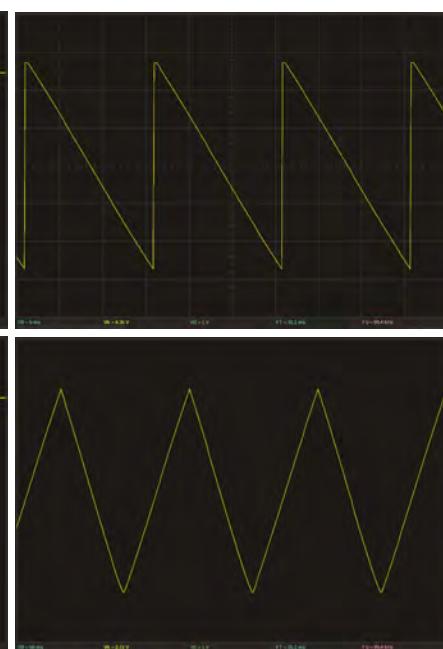
There are only a couple of other elements you can use to make a synthesizer, and they affect how the sound changes over time. These are known as modulators, and there are two common types. The first is an envelope generator, and this is used to specify a level for each stage that a sound is being played, from the initial attack to the point when the sound is released. The second is an LFO, or low-frequency oscillator. This is a slower version of the oscillator used to generate the initial sound. Its frequency will typically be too low to generate audio (not always!) but is used to modulate the amplitude, filter frequency, or pitch.

Having all these separate parts in one setup is known as a modular synth, and that's what we're going to build. Our two modules – a sequencer and a voltage-controlled oscillator – will get our synth started, and you can expand it from there. →



Below ♦

Nearly every synthesizer sound is made up of one or more of these waveforms – clockwise from top left: square, sawtooth, triangle, and pulse (which is an asymmetrical square wave)



BUILD A SEQUENCER

**BUILD YOUR OWN
VINCE CLARKE
TIME MACHINE**



Our synthesizer has two parts: a sequencer and a voltage-controlled oscillator (VCO). The sequencer is used to create a tune. It has

eight steps that will play one after another, before looping back to the start. At each step, there's a button to set it to play or not play, and two knobs (controlling potentiometers) that dial in the notes to play if that step's enabled. This allows us to

program in simple tunes, and edit them as they're playing.

There's quite a lot of outputs from this sequencer and you can integrate it with other sound modules in different ways.

However, the key ones are CV1 and CV2. These are the control voltages that can be used to generate sounds (as we'll see next in our voltage-controlled oscillator). CV1 takes the sequence from one row of potentiometers, while CV2 takes the output from the other. In this way, you can connect this sequencer to two different oscillators to create more complicated sounds. There's no need for it to be exactly this setup (eight steps and two outputs) – this is just the setup we've chosen. If you want more or fewer of either, then that's fine – it's your synth. The only limits are your imagination and your microcontroller's GPIO pins (and you can switch to a microcontroller with more GPIOs if needed).

All the output connectors take audio jacks (as will the inputs on your sound generators). You can connect the two together with jack-to-jack cables. These cables (known as patch cables) are used to set modular synths up to output different sounds. They are, in a sense, the way you program modular synths.

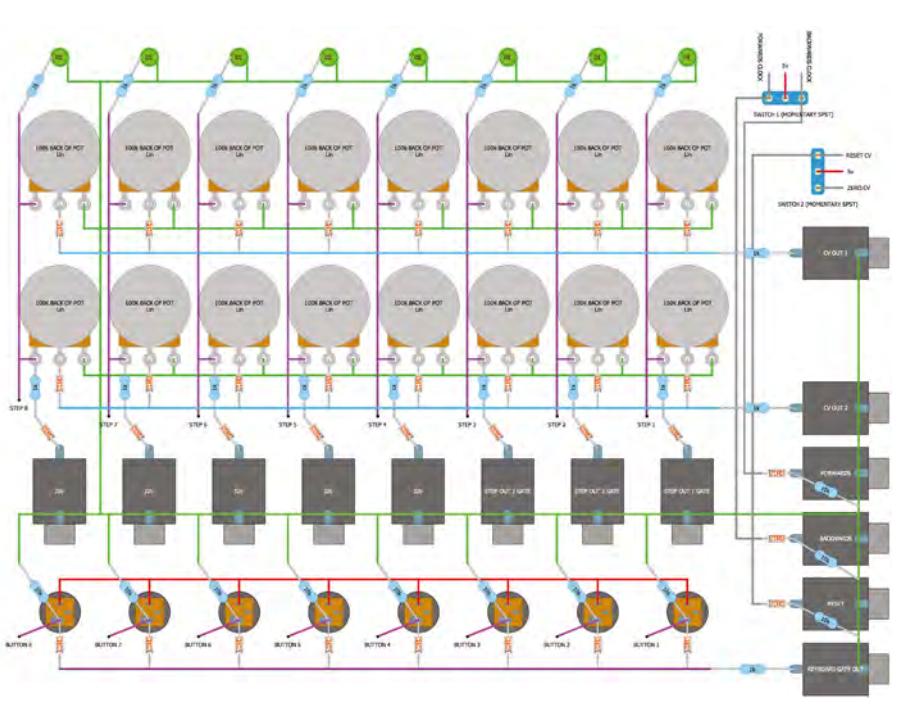


Figure 1 ↗

This is what we're going to build – a sequencer for outputting voltages and an oscillator (VCO) for turning those voltages into notes

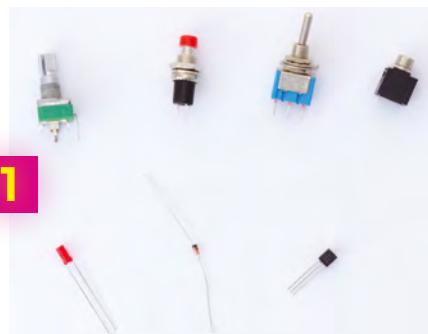
Credit
look mum no computer

1 WHAT YOU'LL NEED

Before we start, we want to mention a couple of things about this specific project. Firstly, it is actually quite straightforward and easy to build, despite what it may look like from the jumble of wires in the end product. We're not going to create complicated circuits with difficult-to-understand components. This sequencer is actually the opposite of a black box, using just a handful of resistors, capacitors, potentiometers, input jacks, and buttons, mostly wired directly to pins on the Arduino, with only simple code to manage them all. But it can also easily become a web of interconnections as you try to wire everything together and make room for everything you want to fit it. We advise you take lots of breaks, and don't do the whole project in one go. Take it a bit at a time, then step away. When you're feeling refreshed, come back and check over the work you did on the previous step. If you've not made any mistakes, carry on! If you have, consider leaving the next step until tomorrow. Take your time and enjoy the process.

The second important thing to note is that you can, and should, change things to suit your needs and imagination. In particular, our project is designed to coexist with other Eurorack modules, and that makes it small. We're soldering most of our components onto a 15x9cm double-sided PCB. If this is your first soldering project, we'd strongly recommend using a larger form factor. Similarly, you may want to avoid the regimented order of the PCB. The PCB format is perfect for prototyping a final circuit, but it needs wires to criss-cross each other to make connections.

Below ↴
See the Ingredients list for all the components you'll need to build the sequencer



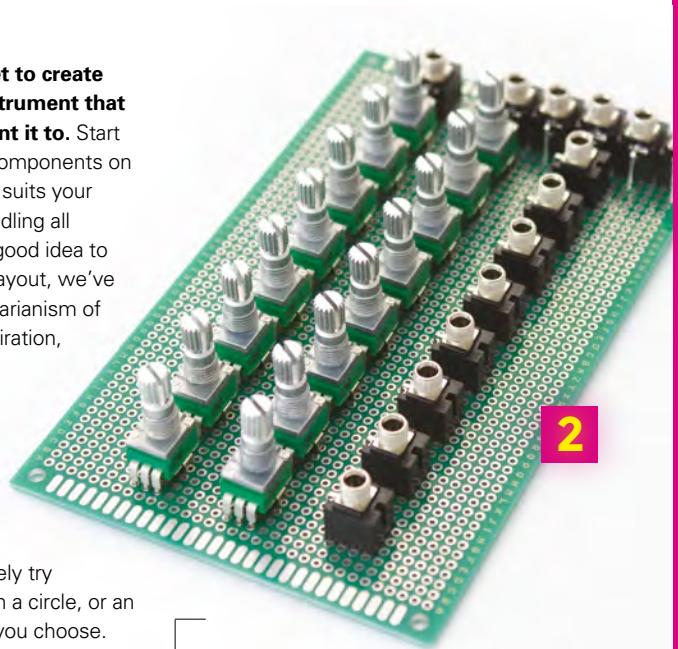
1

2 PCB LAYOUT

This is exciting! You now get to create your own utterly unique instrument that operates in the way you want it to. Start by temporarily arranging the components on a tabletop to find a layout that suits your needs and style. As when handling all electronic components, it's a good idea to ground yourself first. For our layout, we've used minimalism and the utilitarianism of *musique concrète* for our inspiration, so we've gone for a purely functional approach: equally spacing eight LEDs, two rows of eight potentiometers, eight jack outputs, and room on the side for the control outputs, inputs, and switches. You could alternatively try arranging the pots and LEDs in a circle, or an arc, in a 4x4 grid, or any way you choose.

Unlike stripboard, the double-sided PCBs we're going to solder our components to don't bridge any connections, so we'll need to wire or solder everything manually. You need to take this into consideration when you're placing LEDs, pots, buttons, and jacks that are going to be connected together, or to the same input or output on the Arduino. By placing many of our adjacent components horizontally, for example, we can easily create a ground bus to connect them all, and then connect together the same steps in columns and the same devices in rows. But we'll get to that step later.

When you've worked out where everything is going on your PCB, make any adjustments you need, and don't be scared about drilling them. We needed to drill holes through the PCB for the buttons and for the switches, as their legs are too wide for the standard holes. Now slot your potentiometers, jacks, and buttons into their positions. As the pins clip through the holes in the PCB, they should hold their positions, even when the PCB is turned over so that you can solder the legs into position. You can then start by soldering one leg of each component. This allows for some movement in their positions when you make sure everything is aligned, and that the height where the panel will rest on each component is the same. →



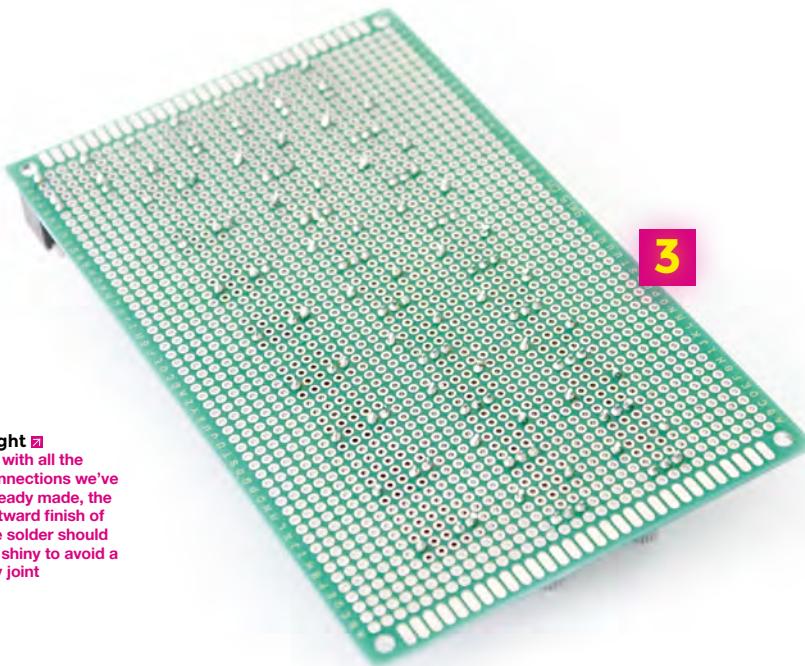
Above ↴
Start by playing with the component layout, without soldering anything

INGREDIENTS

- ◆ Arduino Nano
- ◆ An IC socket or pin headers for the Nano
- ◆ 14 × jack sockets (3.5 mm jacks for Eurorack)
- ◆ 16 × 100 kΩ potentiometers
- ◆ 8 × push-buttons
- ◆ 2 × momentary toggle switches
- ◆ 8 × LEDs
- ◆ 35 × 1N4148 diodes
- ◆ 1 × 78L05 voltage regulator
- ◆ Prototyping PCB or stripboard
- ◆ 18 × 1 kΩ resistors
- ◆ Female socket strips
- ◆ Eurorack power cable
- ◆ Panel or container
- ◆ Mount screws
- ◆ Lots of wire

Build a synth: Sequencer

FEATURE



Right ↗
As with all the connections we've already made, the outward finish of the solder should be shiny to avoid a dry joint

3 SOLDER NEARLY EVERYTHING

At this point, you should really drill the holes in the front panel. If you're sensible, go and do that first, because all of your components are still wobbly enough to wiggle into the holes you make. You can then leave as many components as you are able to in place, especially the LEDs, to make sure their positioning and heights are perfectly aligned with the panel. This is a lot harder to do later when everything is fixed in place. We went ahead and started soldering, however, because we changed a few things while we started to work with the components on the reverse side of the PCB.

We did the soldering in two steps, starting with the pots and jack inputs because these will stay in the PCB when it is turned over. One advantage of doing many components in line like this is that they're quicker to solder, much like a production line. But it's also easier to see when something has gone wrong. One of our input jacks, for example, had a leg folded beneath its plastic, which was easy to spot alongside all the others.

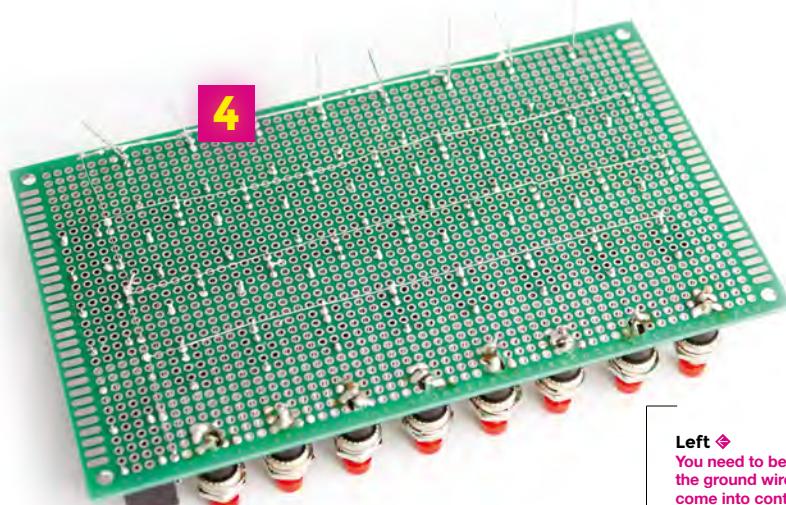
Make sure that all the LEDs have the same orientation. The long leg is positive, while the short leg is negative – if there's no difference in the leg lengths, there should be a flat edge on the plastic of the LED alongside the negative pin. These are also the trickiest pieces to solder without the front panel,

because you not only have to make them exactly the same height – solved by turning the PCB over and placing something of equal height beneath them – you also need to make sure they're the correct height to remain visible when you do fit the panel to the boards. Our solution was to measure carefully, but you're better off poking them through the actual drilled panel and soldering from there. Similarly, our buttons needed to be soldered individually. This is because we'd drilled holes to fit their legs through the PCB, but a blob of solder on either side of the legs on the reverse of the PCB solved the problem.

4 GROUND BUS

You're now at the point where you can start making the circuit. You may want to look at the circuit diagram in figure 1 and tackle this your own way, depending on the layout you've used, but this way worked for us. It's still worth referring to the circuit diagram before each stage to make sure you know what's being connected to where. In particular, you need to pay special attention to the diodes.

We'll start with the ground bus. This snakes its way across many of the components on the PCB. We created five stripped lengths of wire to go the width of the LED row, both potentiometer rows, and the step gate jacks, and across one of the pins for each of the final row of buttons. Connect the negative (short) legs of the LEDs to their wire, the third pin of every potentiometer to theirs, and the outside pin of each gate jack (usually found halfway up the outside of the case) to their wire. Finally, lay down the ground bus for the buttons, and connect each of the rows together with a sixth stripped wire going vertically up either one of the edges. Don't forget, you can anchor any wire down on the PCB if you need to. It also helps if you can leave enough space for other wires you know are going to connect to the other pins on the potentiometer, as well as the jacks and buttons we haven't touched yet. Finally, make sure every ground bus is connected to the others using your multimeter.

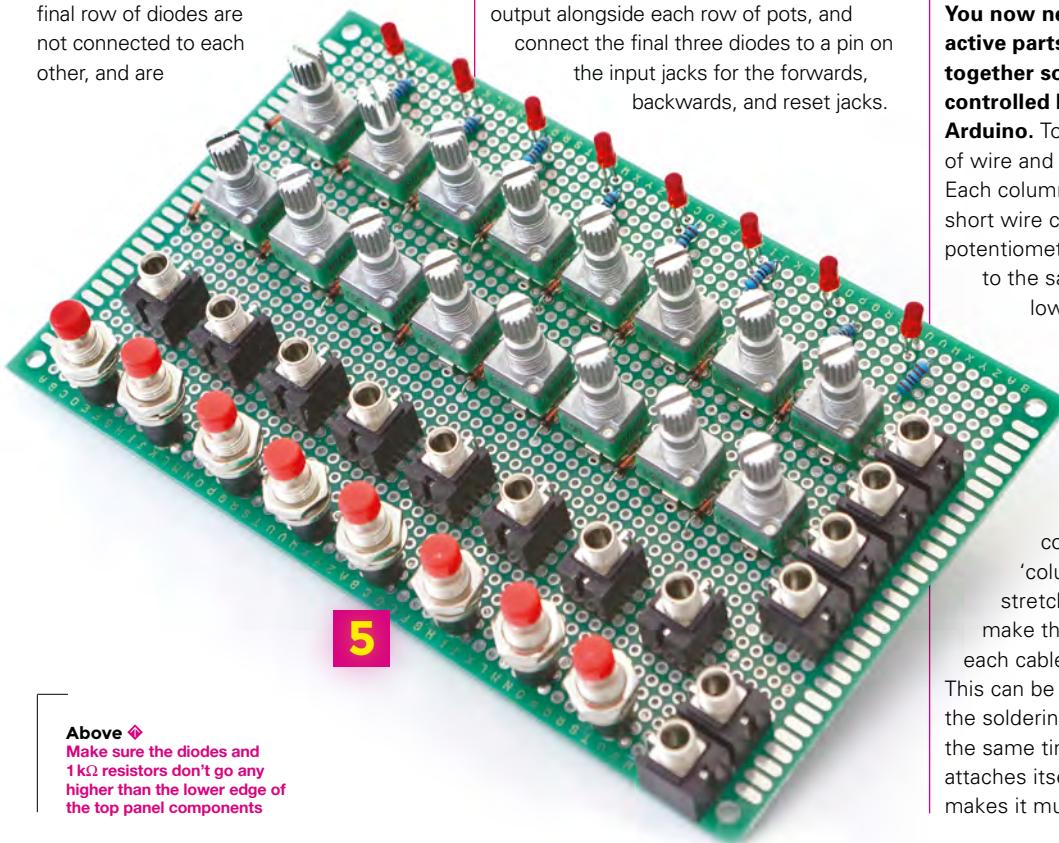


Left ↙
You need to be careful that the ground wire doesn't come into contact with any other component

5 DIODES AND 1K RESISTORS

You're now going to add lots of 1N4148 diodes and resistors, so tread carefully and take your time. Diodes need to be oriented correctly. Start with one of the potentiometer rows, placing eight diodes across the PCB so that the end without the black band is next to the third pin of each potentiometer. If you can, place the other leg somewhere conveniently spaced, so that you can join them up. We did this by pushing the diode through from the top side of the PCB so that both legs were protruding on the side we'd done all our soldering on. We then soldered the leg away from the black band to each leg of the potentiometer, and the diode leg closest to the black band was folded flat and soldered to the next closest diode flat leg. In this way, we were able to solder all the legs closest to the black band to create a bus across the PCB.

Do the same with the other row of potentiometers, creating a bus for their 'black band' pins that stretches across the width of the PCB. The final row of diodes are not connected to each other, and are

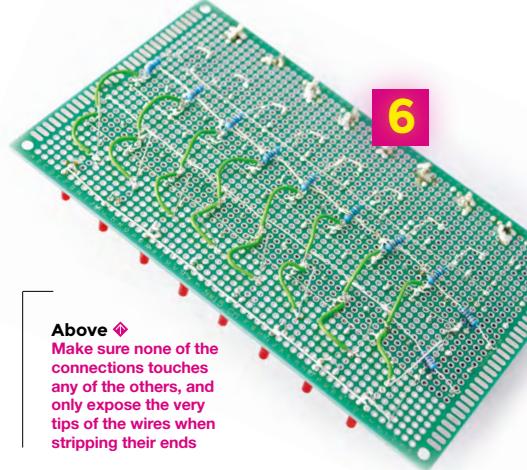


Above Make sure the diodes and 1kΩ resistors don't go any higher than the lower edge of the top panel components

It's still worth referring to the circuit diagram before each stage

oriented the other way around because they're going to connect to the output jacks, where the flow of current is the opposite to the potentiometers (pots). Solder one diode onto each output pin of the jack (not the earth).

It's now time for the resistors – unlike the diodes, you can solder these either way around. Insert them into the PCB next to the positive leg of the LED (the leg that isn't attached to our ground bus), and also next to the jack outputs. This is so that you can solder one leg of the resistor to the other leg of the diodes you just added, while the other leg of the resistor connects to the leg of the closest potentiometer in each column. Finally, add three further 1kΩ resistors to the two CV output jacks and the keyboard gate output alongside each row of pots, and connect the final three diodes to a pin on the input jacks for the forwards, backwards, and reset jacks.



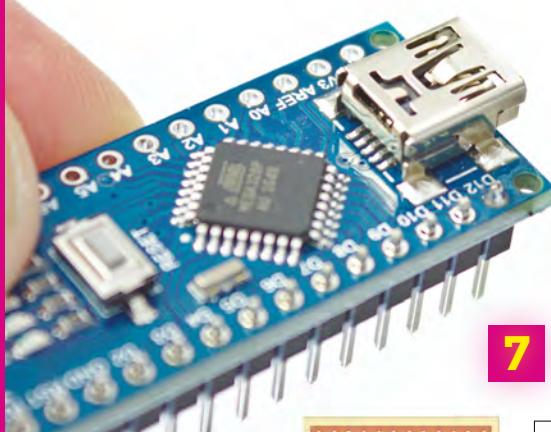
Above Make sure none of the connections touches any of the others, and only expose the very tips of the wires when stripping their ends

6 CONNECTING COLUMNS

You now need to connect each of the active parts of each column step together so that they can ultimately be controlled by the step output from the Arduino. To do this, cut 16 small lengths of wire and eight long lengths of wire. Each column will use three wires. The first short wire connects the third pins of each potentiometer together, connected directly to the same pin as the resistor on the lower potentiometer. The second short wire connects the third pin of the top

potentiometer to the 1kΩ resistor attached to the LED in that column, effectively tying each column together. The long third wire connects to any point on this 'column bus' and will need to stretch to the Arduino board. To make the cables, strip each end of each cable, and 'tin' the end in solder. This can be done by touching the end to the soldering iron and a section of solder at the same time, so that some solder attaches itself to the end of the cable. This makes it much easier to solder the joint. →

FEATURE



7

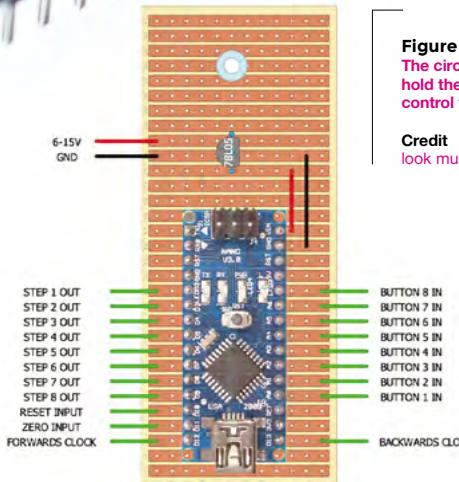
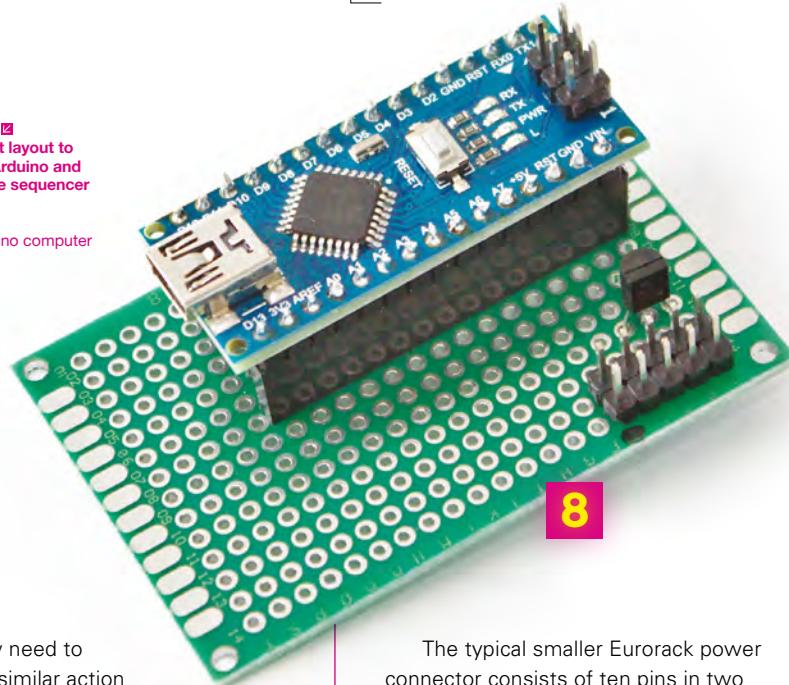


Figure 2 The circuit layout to hold the Arduino and control the sequencer

Credit
look mum no computer



8

7 ARDUINO HEADER

To be able to connect things to the Arduino Nano, you need to start work on the board that's going to hold the Nano and the power supply. We've put the Nano on a separate PCB and, crucially, planned for the Nano to be socketed. Before we could tackle the board, we had to solder strips of pins onto our Nano (although your Nano may have its pins already connected). To do this, heat up one corner pin first, slightly, and then dab on the solder. It should melt and immediately move into the hole and surround the pin, keeping the row of pins in place.

You need to be a little careful when soldering the pins onto an Arduino, as the heat is directly transferred to the electronics. If in doubt, always take iterative steps, gently heating and soldering rather than trying to move too quickly. When soldering long lines of pins, it can be easier to solder a pin at one end and then a pin at the other, so that you can make sure it's straight. This is because you can still manipulate the row into the correct position. When you're happy that all is straight, solder all the pins in between.

You now need to perform a similar action on the second smaller PCB to create a socket for the Arduino. We used two strips of female socket strips that we cut to the correct lengths and soldered onto the PCB. It's easier to do this with the Arduino connected to the two strips, but once again, be careful you don't heat the pins for too long, to protect the Arduino.

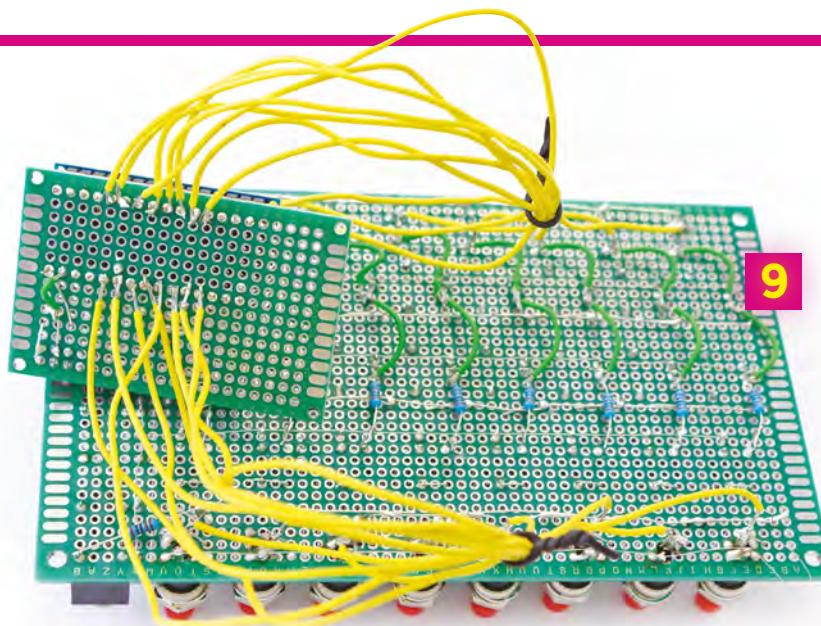
8 ADDING POWER

With the Arduino socketed and ready for fitting onto the smaller PCB, you now need to add power for the entire project. How you do this will obviously depend on how you intend to power the sequencer. The easiest way is actually to do nothing at all, and to use the USB connection on the Arduino to supply the power. This is enough to run the entire sequencer and the Arduino. However, a more permanent solution is to use external power, and we can easily use the power delivered via a Eurorack power bus, which is what other modules will be using.

Below ↴
The ground and power connections to the regulator and Arduino are made with short wires on the underside of the PCB

The typical smaller Eurorack power connector consists of ten pins in two columns of five. Each row is identical, and carries +12V, GND, GND, GND, and -12V respectively. We need to create a pin header for this connector on the PCB and then make a connection from the +12V and one of the grounds to a voltage regulator, which sits between the power coming in and the power and ground we connect to the Arduino. The sequencer power will simply take the 5V and the GND directly from the Arduino.

Start by creating the ten-pin header for the Eurorack power. This should be easy after having just done the same for the Arduino. Place the 78L05 regulator on the PCB too, and connect the power to the bottom pin of the 'D' shape of the regulator and the ground to the middle pin of the regulator. From there, on the underside of the PCB, take the power from the regulator and connect it to where the Arduino's 'VIN' pin is going to connect to the header, and do the same to connect ground from the regulator to the GND adjacent to the VIN on the Arduino. That's all there is to it. ↵



9 BUTTONS, SWITCHES, JACKS, POTS, AND LEDs

We're now going to connect each of the steps, all the buttons, and the various remaining jacks and switches together with the Arduino, so there's going to be a lot of wire. The button bus we created earlier, which is isolated from everything else on the PCB, is going to carry the 5V from the Arduino, as are the middle pins of both momentary switches. As with all our Arduino connections, we're taking one wire from the main PCB to the underside of the PCB with the Arduino. It would be neater if we routed all these wires to a separate header and then used a ribbon cable to connect the two, but we'll leave that as a further exercise. Now that we can see how much space there is on the PCB, we also insert the two switches into the top-right of the PCB. They could equally be left free, attaching only to the faceplate. It's important that these are momentary switches as this means they won't stay in position. Press them once to go backwards and the other way to go forwards.

The ground bus now needs to be connected to the underside of the GND pin on the Arduino. This is a good point to check continuity again by making sure your multimeter emits a sound when one connector is on the Arduino GND and the other is on any one of the ground buses you created on the PCB.

Now prepare twelve cables and solder eight from each button before the diode to pins A0 through to A7 on the Arduino PCB, left to right (see above image). Likewise, solder the eight cables we previously attached to the LEDs, potentiometer, and

Above ♦

If you have the time and space, it would be better to use four headers and two ribbon cables to connect the main PCB to the Arduino

jack columns to D9 through to D2 on the Arduino, from left to right. Now connect the reset jack to one side of one switch, and from the switch to D10 on the Arduino. The other side of the switch needs to be connected to D11. For the final switch, connect one side to the forward input jack and the other side to the backward input jack, also connecting the forward side to D12 and the backward side to D13 on the opposite side of the Arduino.

10 DRILL CASE

We did lots of things wrong in this step.

Firstly, and most importantly, we should have done it earlier, before soldering everything in permanently. We didn't because we were still making up the layout as we went along. Secondly, we drilled holes into an aluminium panel using a hand-held power drill. This works, but won't give the most professional-looking results. If you've got access to a drill press, use this instead.

As we have carefully placed all the components on a PCB, you can easily duplicate your layout on a piece of mathematical paper. We did this and traced the places where we needed holes onto a piece of cardboard, which we then pierced to make holes and mark the aluminium panel for drilling.

11 FLASH ARDUINO AND GO!

You need to download the Arduino IDE for your operating system (hsmag.cc/OYiLpN).

Plug your Nano into your USB port, then open up the code for this project (git.io/fpz1h), select the Arduino Nano as your device, select the USB port it's connected to, and click 'Upload'. You're now ready to insert the Arduino into your sequencer! Plug in the Arduino and then add the power. With a bit of luck, you shouldn't see anything. Try pressing a button. Its LED will light. If you connect the gate output to an oscillator or sound source, it should trigger a sound. Connect CV1 or CV2 outputs to pitch on the oscillator and it will play whatever pitch is dialled into the potentiometer. If you have a clock source, plug this in and the sequencer will start stepping through each column at a time.

If things aren't working, don't be disheartened. Projects rarely work perfectly first time. If an LED is skipped over, make sure all the LEDs are connected properly. If things flash and flicker, look for bridged connections. And sleep on it. With a fresh mind, any mistakes will be obvious and you can start enjoying your new modular sequencer and synthesizer.

Below ♦

If you place your sequencer into a rack, consider running a USB extension to the outside so you can still flash your Arduino

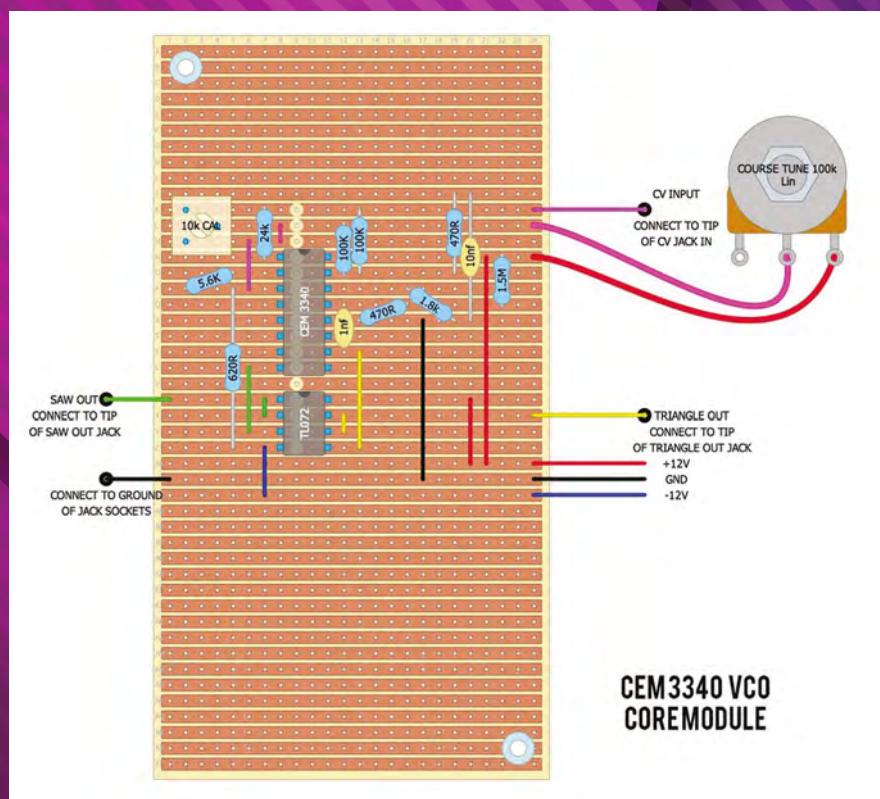


BUILD A VOLTAGE-CONTROLLED OSCILLATOR

YOU'VE BUILT THE SEQUENCER, NOW YOU JUST NEED TO BUILD SOMETHING THAT MAKES A SOUND

Figure 3 →
Here's the circuit layout of the project we're building. Huge thanks to Look Mum No Computer

Credit
look mum no computer



1 WHAT YOU'LL NEED

This project isn't as complex as the sequencer, and is easier to put together.

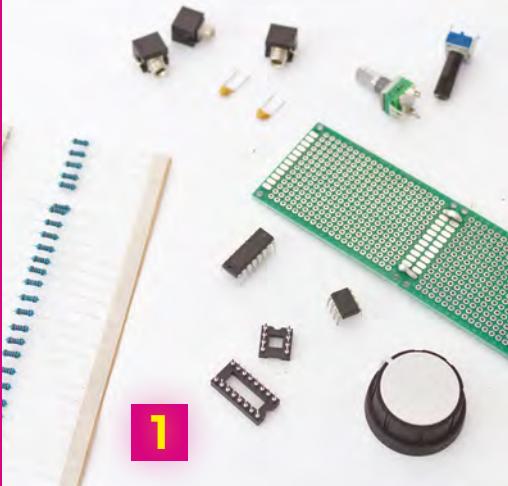
However, with the form factor we're using, it's more fiddly to solder all the connections together.

The most important part of this build, and the most exciting, is the chip that generates the sound. This is the venerable Curtis CEM3340, a chip that helped pave the way for the mass production of analogue synthesizers. It was used in many classics, including the Memorymoog, Oberheim OB-8, Roland SH-101, and Sequential Circuits Prophet 5 (rev 3).

The reason why the chip was so revolutionary then is the same reason why we're using it now. It's a completely self-contained VCO that generates multiple waveforms, and needs very few additional components to work within a circuit. Before the CEM3340, a VCO would need to be constructed from lots of different and difficult-to-source parts, especially when you needed them to sound the same and stay in tune. By using a real CEM3340, we get exactly the same source sound as those old synths, and if the genuine chip is too expensive (typically around £12), there are replicas that perform the same function for around half the price.

Below

Here are most of the things we'll need to build the oscillator. It's fiddly, but there aren't too many parts



By using a real CEM3340, we get exactly the same source sound as those old synths

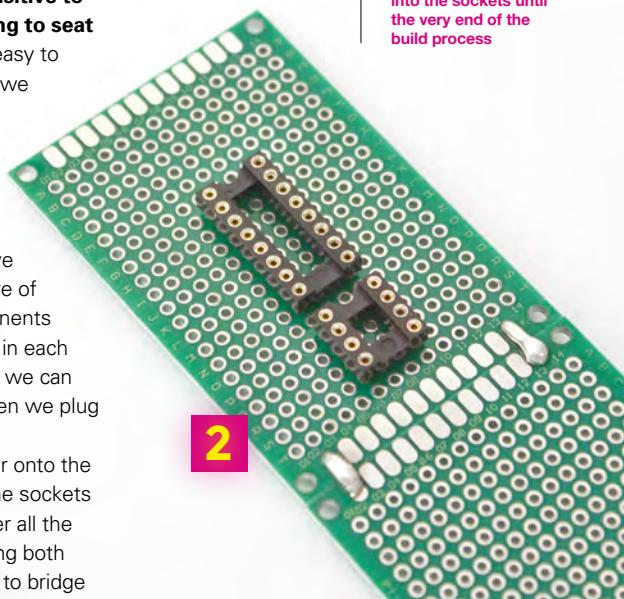
2 SOCKETS

As the VCO is the most valuable part of this build, and the part most sensitive to electrostatic damage, we're going to seat it within a socket. This makes it easy to replace the chips, and also means we can solder the socket, and the socket's pins, without the chip being seated, protecting the chip from the heat. We'll do the same for the TL072 too, which is used to condition the output. We've positioned the sockets in the centre of the PCB as we'll be adding components to each side. Make sure the notch in each socket is facing the top – this is so we can orient the chips into the circuit when we plug them in.

To solder the sockets, dab solder onto the pins on opposite corners to hold the sockets in place, and then proceed to solder all the points in between. As we'll be using both sides of the PCB, we'll often need to bridge adjacent connections from the socket pins, which is worth considering as you start adding components and wires. Take a look at some of the later steps to see how we routed wires around the chips and legs of the pins, and keep this in mind when you create the power header.

This step will be different if you want to power your module from a battery or other power source – unlike the sequencer, which can alternatively be powered via a USB connection on the Arduino. We're using a standard Eurorack power supply. This needs eight pins on the PCB, with +12V delivered to the top pair, -12V to the bottom pair, and ground (GND) in the middle section. We can use these directly with the VCO. The header is created in the same way as the sequencer header, cutting off two rows of eight pins, placing them within a Eurorack power cable, and then soldering them through the PCB, ensuring the connector is on the same side as the PCB.

Below
Don't insert the chips into the sockets until the very end of the build process

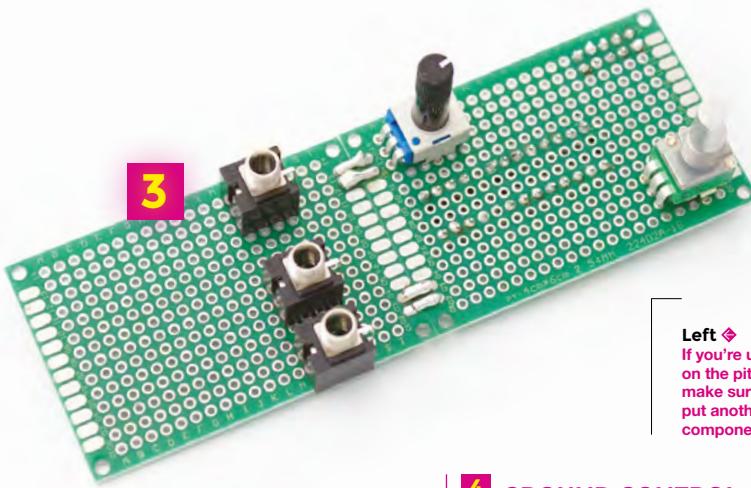


INGREDIENTS

- ◆ 3 x jack sockets (3.5 mm jacks for Eurorack)
- ◆ 10 kΩ trimpot, or potentiometer, for front panel access
- ◆ 100 kΩ potentiometer
- ◆ TL072 amplifier
- ◆ Eurorack power cable
- ◆ VCO: CEM3340, or the copy AS3340
- ◆ IC socket: 1 x 8-leg, 1 x 16-leg
- ◆ 46x24 dotted stripboard or PCB
- ◆ Resistors: 2 x 100 kΩ, 2 x 470 Ω, 1 x 620 Ω, 1 x 1.8 kΩ, 1 x 5.6 kΩ, 1 x 24 kΩ, 1 x 1.5 MΩ
- ◆ Capacitors: 1 x 1 nF, 1 x 10 nF
- ◆ Knob for potentiometer
- ◆ Lots of wire

Build a synth: Voltage-Controlled Oscillator

FEATURE



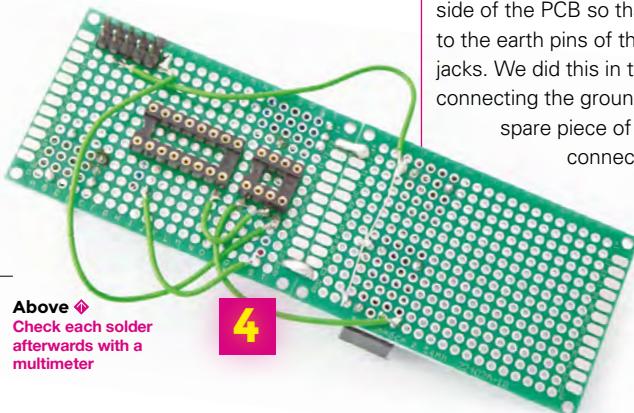
Left ♦
If you're using a large knob on the pitch potentiometer, make sure you don't put another front panel component too close to it

3 POTS AND JACKS

Let's start with the big components first, as we want to make sure these fit onto the PCB before attaching the many wires.

These are two potentiometers, one of which is used to set the pitch of the oscillator, while the other tunes it, and the three jacks. One jack will take an incoming voltage so the pitch can be controlled, while the other two output sawtooth and triangle waveforms from the CEM3340. These go on the reverse side of the PCB from the chips and power header. This is so they're presented to the front panel while the chips and power header remain accessible from the rear. It's worth keeping the jacks on the same row as we'll need to connect their ground pins together, usually by running a wire across all the top pins.

Make sure your pins aren't too close to be soldered and that nothing will be obscured by elements on the front panel. Our 10 kΩ potentiometer also needed to have its anchoring legs trimmed to fit through the PCB. With everything in position, you just need to dab the soldering iron against a little solder and the pins to solder them in place.



Above ♦
Check each solder afterwards with a multimeter

4 GROUND CONTROL

We're going to refer closely to the circuit diagram in figure 3 to make sure we don't miss any connection or component. You may find it useful to tick off each as you make them. Also, don't forget that stripboard has the horizontal rows implicitly connected, which isn't the same on the PCB. This means you need to make sure everything on one row is interconnected, either with a wire or by soldering across adjacent holes. But before we get to that stage, we first need to feed the jacks a connection to ground.

All the GND connections are going to come from any of the middle pins on the power header. The header is on the rear side, facing away from the pots, which is also the side we want to use to solder the connections. This creates a slightly tricky situation where we need to solder a hole adjacent to a pin, bridge the connection to the pin, and make sure the solder runs through the hole to the other side of the PCB, from where we can solder a wire to the destination. It sounds harder than it is. We first need to do this for the ground connection, bringing a GND connection from the power header through to the reverse side of the PCB so that we can solder a wire to the earth pins of the input and output jacks. We did this in two stages, first by connecting the ground pins together with a spare piece of bare wire, and then by connecting this to the ground connection. Though not in the original diagram, we also connected ground to the third pins of each potentiometer.

HackSpace

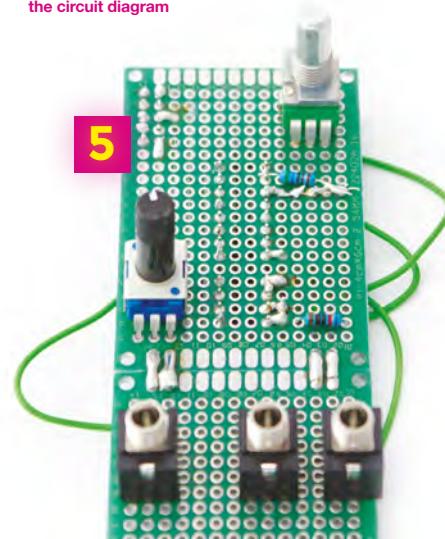
5 CHIPS AND PINS

We now need to work through all the connections on the board. You will need around 20 sections of wire in total, but they'll all be slightly different lengths. Don't forget to 'tin' each stripped end of wire first with a little bit of solder.

For the overall strategy, we found it easier to start with the left side of the chips on the rear side of the PCB, working upwards from the CV input jack. Take time to see where the connections need to be made and take it step-by-step, working up from the input jack.

The -12V connection from the power supply is connected to the lower left pin of the TL072 socket. This is pin 4. Pin numbers on chips go from the top left, which is always pin 1, down the left side and then continue from the bottom right to the top right, which is pin 8 on the TL072 and pin 16 on the CEM3340. You also need to bridge pins 1 and 2 on the TL072 socket, which you can do directly from the pins on the front-facing side of the board. You'll also need to bridge and flow the solder through for the jack connection and for the first resistor, the 620Ω, which connects this pin to 3 on the CEM3340. Continue like this for the three resistors on the left side, plus the trimmer pot and the sawtooth output, which will need to connect to the pin opposite the ground on the jack you're using to output the sawtooth.

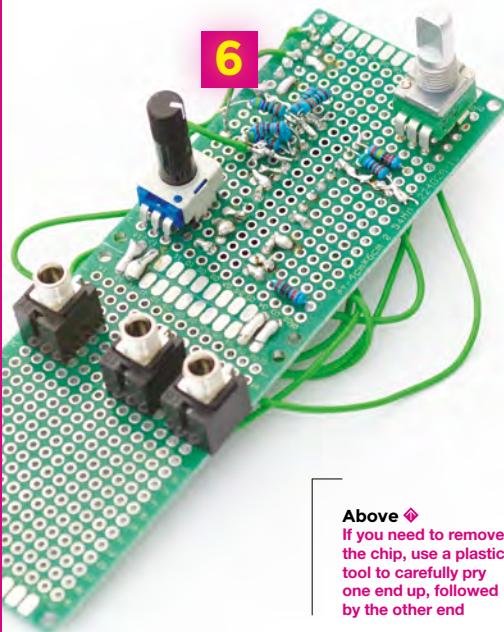
Below ♦
It would be tedious to cover every connection in this tutorial. It's easier to methodically step through each wire and resistor in the circuit diagram



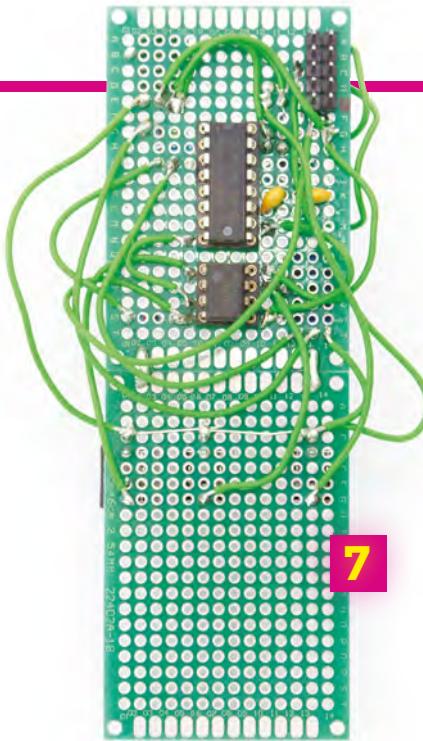
6 POWER AND CAPACITORS

The left side of the PCB is completed when pin 3 of the CEM3340 is connected to one leg of the trimmer, and pin 1 (via the 24 k Ω resistor) is connected to the other leg. As mentioned earlier, our trimmer had three legs, and we connected this third leg to ground. It's now time to tackle the right-hand side of the circuit, and this is just more of the same, albeit with a higher component and wire density. Pins 6 and 7 on the TL072 are bridged, and pin 7 connects to the tip of the triangle output jack. Pin 5 then connects to pin 10 of the CEM3340. The tightest soldering is off pin 15 of the CEM3340, near the top right of the chip. This pin needs to connect to two 100 k Ω resistors, the 470 Ω resistor, a connection from the +12V power connector, and the output to pin 3 of the coarse tune potentiometer. Getting these soldered in an orderly way was almost impossible with our small PCB, but each pin of the resistor can be soldered together or used to bridge a horizontal stretch of the PCB for the other connections.

Just take your time and make each connection in turn. There are far fewer to make than the sequencer, and there's usually plenty of room when you start soldering legs together.



Above ♦
If you need to remove the chip, use a plastic tool to carefully pry one end up, followed by the other end



Above ♦
We found it easier to push the capacitors through from the side with the sockets and chips

7 CHIPS AND POWER

Once everything is soldered together, you can insert the two chips into their sockets. New chips will have their legs slightly too far apart to fit into the socket. This is normal and you need to use a blade, or something with a straight edge, to fold both sides in slightly. Regarding pressing the chips in, ensure the notch in the chip is aligned with the notch in the socket. If either chip doesn't have a notch, look for a circle next to one of the corner pins; this is used to mark pin number 1, and should be oriented into the top left position of the socket.

You're now ready to see whether your oscillator works. This is the most exciting step – if it doesn't work, disconnect the power and look at your circuitry. To test the oscillator, connect the power, making sure the red stripe on the Eurorack power cable is facing down where the -12V needs to be.

Connect the sawtooth output to a mixer or a PC audio input, or something you can listen to. Now turn it on. There's a chance you won't hear anything. You need to first use the tuning potentiometer to bring the pitch within range. Try to sweep across this until you hear anything, even if it's a loud occasional thump. As soon as you get a sound, use the pitch knob to dial in a tone. Congratulations: you've just built your own classic oscillator!

8 TESTING AND FRONT PANEL

The oscillator will generate a constant tone, with the pitch being set by either the 'pitch' potentiometer or an incoming control voltage to the pitch input jack. The CV input is used to play notes, and the pitch potentiometer can then be used to control the root notes from which the CV input will diverge. Standard voltage tuning for almost every Eurorack module and keyboard is 1V per octave, meaning the twelve semitones in an octave are divided across a single increase or decrease in one volt. And the great thing about the CEM3340, and why we didn't need any more complicated circuitry, is that it also tracks pitch at 1V per octave. Connecting one of the CV outputs from the sequencer to the CV input on the oscillator means you can now create your own sequences.

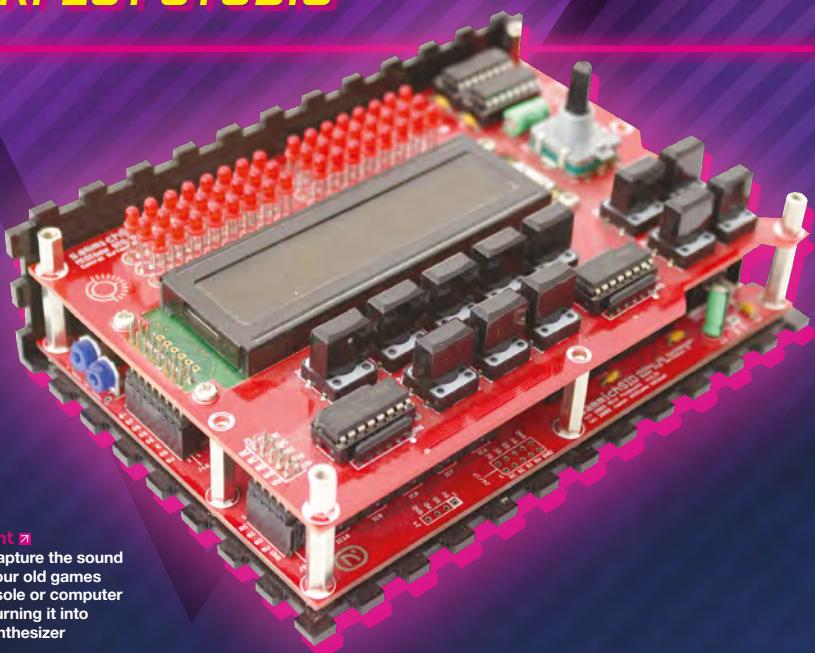
All that's now left is to create the front panel. This is going to depend entirely on how you're going to use your VCO. We sneakily kept space next to the sequencer front panel so that both modules can be mounted into the same unit, so all we needed to do was drill five new holes for the potentiometers and the jacks. As with both of these projects, our aim has been to create a great-sounding and useful sound engine. When you strap in the VCO alongside the sequencer and connect them together, you'll have a powerful and capable proto-synthesizer from which you can expand in almost every direction, and yet it already sounds absolutely fantastic.

Below ♦
Here's the completed module fitted alongside the sequencer. The only job remaining is to spray-paint the panel so you don't see the scratches!



TAKING SYNTHS FURTHER

NOW YOU'RE HOOKED, SPEND THE NEXT TEN YEARS BUILDING YOUR PERFECT STUDIO



Right ↗
Recapture the sound of your old games console or computer by turning it into a synthesizer



ith a sequencer and VCO, you now have something that can be used to make awesome music, even just by feeding the pitch

from the sequencer into the oscillator. This is what Kraftwerk did, and Wendy Carlos had only a few more oscillators when performing *Switched-On Bach*. But this is just the beginning, and you've hopefully got the itch to take both of these projects further, and to look at extending your new 'studio' into something with more possibilities.

POWER-UP YOUR RIG

If you're going to stay with the Eurorack, and you should, you're going to need a Eurorack power supply. You can obviously buy these along with the racks to hold your modules, but you may also want to build your own – and for just £10, you can. The Frequency Central Power DIY kit, (hsmag.cc/oNEViK), as recommended by Look Mum No Computer, is perfect for creating +/-12V, ~100mA at 5V from a 12V AC power supply. You'll then be able to connect your modules

directly to the same source, and add new modules with ease.

We've barely tapped the potential of the Curtis CEM3340 chip. In particular, it can also generate pulse and square waveforms without too much extra circuitry. From a technical perspective, both of these waveforms are actually generated from the same source, as the 'width' of the square shape can be modulated using another voltage control input. With a 95% width, for example, the waveform is a sharp pulse, whereas at 50%, it's half-way up and half-way down, which is the shape of the square wave. Changing this percentage is called pulse-width modulation (PWM) and is another classic synthesizer sound source

changes like these, share them with the community. You never know where that might take you.

ADDING HARDWARE

While the sequencer and oscillator are capable of generating some excellent sounds, they still don't quite fulfil the role of an entire synthesizer. For that, you're going to need a few more modules, as we explained at the beginning, and these are a great place to start if you want to expand your module collection. Your first addition should be a filter, as this will add much needed character and harmonic control to your synthesizer sound. There are as many different filter designs as there are

It's built around an Arduino, and that of course means you can change its functionality

that's amazing for bass as a square, and amazing with sawtooth when it's closer to a pulse, and amazing when you adjust the PWM amount with an incoming voltage.

If you don't feel like doing any extra soldering, the great thing about the sequencer project is that it's built around an Arduino, and that of course means you can change its functionality through the code without having to change any of the hardware. In this way you can make it truly unique and specific to your own needs. You could add a random step mode, for example, or add swing timing so that the clock isn't so regimented. You can also change the functions of the switches – especially as the backwards and forwards switches may be unnecessary if you're using both clock inputs. The switches could even be used to change between various playing modes for the sequencer, and you could show which mode or preset you're using by briefly hijacking the LEDs to show a patch number. And, if you're only using one clock input, change the code to use the other input for something else, such as changing direction, or doubling the speed. This really is the best thing about building your own modules. If you do make

synthesizers, and this being modular, you can (and should) aim to have more than one.

Your second and third additions should be a VCA, a voltage-controlled amplifier, and an EG, an envelope generator. This is because there's currently no way to attenuate the output from the oscillator, so the sound is always on. By plugging the audio output from the oscillator into a VCA, you can control the VCA levels over time with the EG, and you can trigger when the EG starts using the gate outputs from the sequencer or button keyboard. If you want the gate to match the pitch change, use a 'multiple' to split the clock into two, with one end going to the sequencer and the other to the EG. This is exactly how modern synthesizers respond to input from a keyboard. Many synths will have two or even three EGs, because they can also be used to change the amount of filter over time, or adjust the pitch of the oscillator over time, although you could equally use one and patch the control outputs to multiple destinations. A multiple is a module that takes a single source and provides multiple outputs; certain modules can take 'banana' cables that allow you to connect more than one cable to a single jack output. ↗

VCV: VIRTUAL MODULAR SYNTHESIZER

If you're still a little intimidated about the various elements that need to come together to make a synthesizer, of even the nascent beginnings of a Eurorack system, the answer is to experiment with software first. And there's an amazing piece of open-source software that not only teaches you about how all these various synthesizer modules fit together, and what they sound like, it teaches you about the exact modules you can build, buy, and install on your own system. VCV (vcvrack.com) is a virtual rack for virtual recreations of real hardware. The software is free and open source, and accurately models everything about a module, from its panel design and interface, through to loading the actual firmware that digital modules run, and the emulations of all the components in the circuitry.

You can even recreate our humble project without soldering a single component. Just install and run VCV on your chosen operating system (Linux, Windows, and macOS are supported). The main view is an empty rack for you to fill with modules, and you don't have to worry about power. Just right-click and select 'Fundamental' to open the menu of basic modules and select 'SEQ-3' to add a sequencer almost identical to the one we built. Similarly, select 'VCO-2' to add a simple oscillator. We also need to get the audio out from the virtual rack into your headphones or speaker, and you do this by adding 'Audio' from the 'Core' module. Now connect the CV output from one of the rows on the sequencer to the FM input on the VCO and turn up the 'FM CV' knob, and connect the output from the VCO to an input on the audio module. If you select an audio device with a right click, you'll immediately hear the pitch of the VCO being modulated by the potentiometer on the sequencer, just like our real hardware. You can now experiment with additions and new configurations without having to build the real hardware.

Below

If you don't want to play with a soldering iron, you can create a perfect virtual Eurorack using the open-source VCV software



OTHER PROJECTS TO TRY

IF YOU'RE NOT YET READY TO PUT DOWN THE SOLDERING IRON, THESE ARE THE PROJECTS WORTH TAKING A LOOK AT FOR INSPIRATION



Right →
Small kits, like Radio Music, are a brilliant way to start with modular synth building, as they're cheap and can be expanded, upgraded, and subverted easily



here's a considerable community that's grown, and continues to grow, around build-your-own synthesizers, catering for all kinds of different styles of music and all kinds of

engineering capability. These projects range from simple circuits that will take an hour or two to complete, to fully fledged re-creations of old classic synthesizers that could take a year of work, and salary, to put together. Many people who started off building their own creations will now sell their own PCBs and faceplates, and buying one of these kits is a great way to grow your own collection without having to design everything from scratch or drill your own holes through aluminium plates.

RADIO MUSIC

One of the simplest projects you can start with, and one of the most creative, is called 'Radio Music'. Radio Music was designed by Tom Whitwell and inspired by early musique concrète experimentation by the likes of John Cage, Karlheinz Stockhausen, and Don Buchla, as they each played with loops of randomly recorded lo-fi radio. It takes an SD card crammed full of raw audio files you

make yourself or source. Radio Music then lets you control how those files are played back, from which point, and from which 'channel'. It's a wonderful sound source that can operate as a VCO with a completely non-traditional sound. But the hardware is also completely open, and that means the schematics, along with the designs, the bill of materials, and the code that runs on the Teensy microcontroller are completely open source (CC-BY-SA). It's also suitable for all levels, because not only is it an easy project for beginners to put together, you can even buy it fully completed if you'd prefer. You can build it yourself from the information on its GitHub page, you can buy the PCBs and the front panel, and you can buy kits that include everything you need, bar the soldering iron and the solder.

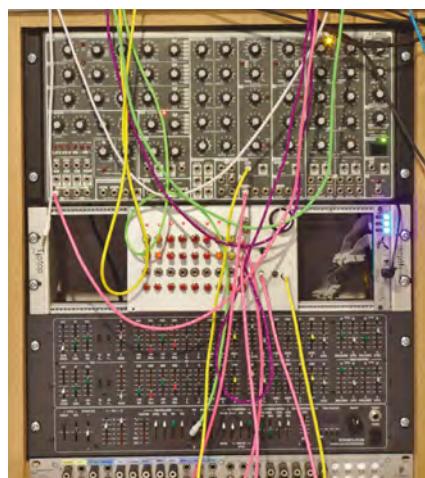
Another great thing about Radio Music is that it can also be something else completely different – a chord organ and rather gritty VCO. With exactly the same hardware, only with a different Teensy firmware, it switches from being a digital playback device to something that can play different chords. The 'station' knob now switches between chords, for example, while the 'start' knob adjusts the root note and octave for the chord. Finally, the 'reset' button now selects

between sine, square, sawtooth, and pulse width. It's brilliant for generating complex chord progressions from a single module and a simple kit. For more details, check out Tom's GitHub page: hsmag.cc/UPiAJO.

There are hundreds of other Eurorack modules that you can build yourself from designs shared online, or from kits put together for usually modest sums of money. In the UK, there's even an annual meet-up of these home-grown modular synth moguls in Brighton. At the 2018 event, you could rub shoulders with these makers and their hardware, listen to music made with the equipment, and even attend workshops to help you build your own modules – from beginners to experts.

ADVANCED PROJECTS

Another great source of projects that require varying amounts of commitment is MIDIbox. Rather than fitting into the Eurorack format, many of MIDIbox kits and circuit designs create self-contained units that can be used to control other devices, or turn old and esoteric sound hardware into a



Above Our sequencer in its natural home, alongside other synth modules

reverse-engineer the original PCBs of some old equipment and design new circuits to include more easily available components, putting the whole thing together over multiple revisions and kits until a final stable version becomes available. At this point, the PCBs are usually manufactured in small runs, and a BoM for the build is created for other people to follow.

MIDIbox kits create self-contained units that can be used to control other devices

synthesizer. There's a PCB for generating sound out of a Commodore 64's SID chip, for example, and another for creating a bank of faders that can control various synthesizers over MIDI. Many of these projects also have points that can be used to add control voltages, and many builders bend the designs to suit their own requirements and form factors.

But if you've mastered the soldering and you're looking for a real challenge, there are plenty of hundred-hour-plus projects that synthesizer affectionados pour their time and money into. Many of these projects involve recreating classic and unobtainable synthesizers from the 1970s, and they are a huge challenge for a number of reasons. They usually centre on a few individuals who

One such project that followed this path is TTS - an acronym for 'two thousand six hundred', which just happens to be the number for a very classic and now very expensive 1971 synth, the ARP 2600. TTS is a complex project that has itself been refactored into another clone, the STP 2600, which promises to be much easier to build without compromising the sound. Take a look at diysynth.de if this sounds like your kind of endeavour.

We have to give huge credit to Sam Battle's 'Look Mum No Computer' for the circuit designs we've used as the basis for both the oscillator and the sequencer. It's his website, lookmumnocomputer.com, and in particular his Patreon page, patreon.com/lookmumnocomputer, we'd recommend looking at for your next steps. His site includes the extra circuitry and components needed to tap into the missing elements on the oscillator, and he also has schematics for extending the sequencer to use an Arduino Mega and add another eight steps, if you can handle the extra wiring.

For the ultimate in synth sound nostalgia, many people consider the Yamaha CS-80 to be the definitive synth of the 1970s, and even this has succumbed to DIY enthusiasts.

The CS-80 was famously used by Vangelis in his late 1970s and early 1980s golden era, on soundtracks such as *Antarctica*, *Blade Runner*, and *Chariots of Fire*. The sound of the CS-80 is what many people consider the sound of synthesizers, with sweeping pads and strings dripping in eight seconds of reverb, and yet its signal path is rather unusual, consisting of two parallel voices and eight notes of polyphony. That makes 16 voices in total, alongside a strange parallel filter control and polyphonic aftertouch. And like the ARP 2600, you can now embark on a DIY project to build a synth with the same character if you're prepared to spend hundreds of hours and pounds on the components, PCBs, and cases. This DIY recreation is called Deckard's Dream (deckardsdream.com) and may well be the sonic equivalent of a unicorn running through a forest clearing. □

Below A Yamaha CS-80 synthesizer can now cost tens of thousands of dollars, and you need a full-time engineer to keep it tuned. If you have the patience and the time to solder thousands of components, you can get close to the CS-80 sound with a DIY Deckard's Dream synthesizer



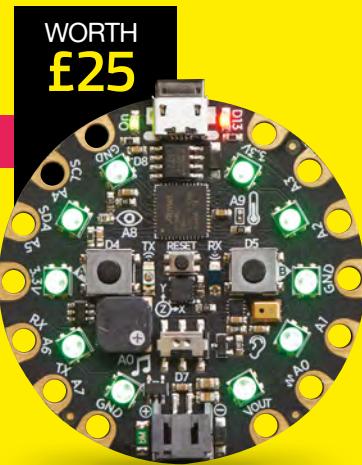


CIRCUIT PLAYGROUND EXPRESS

WITH 12-MONTH PRINT SUBSCRIPTION



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

SUBSCRIBER BENEFITS

**SAVE UP TO 35% ON THE PRICE
FREE DELIVERY TO YOUR DOOR
EXCLUSIVE OFFERS AND GIFTS**

OTHER WAYS TO SUBSCRIBE

Quarterly subscription
Get your first three issues from £5:

- Use the code HS-SAVE at the checkout
- Spread the cost of your subscription
- Try out HackSpace magazine with no commitment

Rolling subscription from just £5 a month:

- Quick and easy to set up
- Cancel any time
- No long-term commitment
- No large up-front cost

DIGITAL SUBSCRIPTIONS ALSO AVAILABLE



Visit: hsmag.cc/subscribe

How I Made

MINI.MU GLOVE

Turning an idea into a product

By Helen Leigh

The MINI.MU is a make-it-yourself, gesture-sensing musical glove for children. It is a collaboration between international recording and performing artist Imogen Heap, the MI.MU glove team, the super-friendly people at Pimoroni, and myself.

My name is Helen Leigh. Things I am: a children's author, an electronics nerd, and a maker of creative technology. Things I am not: a product designer, a marketing guru, or an engineer. I made the first version of this glove in a shared makerspace in Hackney, and six months later it's about to become a real product that people all around the world can buy. It's been an eye-opening journey for me as a maker, and I want to share the story of how the MINI.MU went from make to manufacture.

HOW IT STARTED

I'm a hardware hacker and a musician. All the best things I've done combine these



Above The Mini Mu glove: bringing technology and music to a school near you

two things. I've made breadboard synths live on air on *BBC Click*, I've made DIY electric guitars in classrooms all over the country, and I've made a neon pink wearable capacitive touch synth in a field with my friend Phoenix Perry.

As I became more involved in music tech hacking, I became part of the Hackoustic and Music Tech Fest communities, where I met people making extremely odd musical instruments. At our awesome gatherings, I saw music made with marble runs (by Andrew Hockey), terrifying pipe organs made out of hacked Furbies (by Sam from Look Mum No Computer), a gong made out of a massive rusty saw blade (by Tim Yates), and interactive games with adorable singing capacitive touch robots (by Phoenix Perry).

I started to use these ideas as inspiration in the classrooms I visited, to teach subjects as diverse as music theory and physics, and saw the effect it had on the imagination and understanding of what can often be seen as very dry subjects. I also saw how naturally young people take to making noises, and



Right ◇
Users can make the
glove in different
sizes to fit their hand

how music can be used as a joyful vehicle for knowledge.

At around this time I saw a demonstration of a prototype MI.MU glove, invented by Grammy award-winning recording artist Imogen Heap and her team, and I immediately started thinking about how I could bring this magical instrument into the classrooms I worked in.

WHAT IS THE MI.MU?

The MI.MU glove shows that there is a better way to make music than with sliders and buttons – through the complex movement of the human body. The MI.MU glove combines textiles and electronics, including flex sensors, gesture detection using IMU, haptic feedback, and state-of-the-art low-latency wireless communication. It uses special software to translate movement into MIDI or OSC messages, and has been used around the globe by world-leading musicians, including on Ariana Grande's world tour.

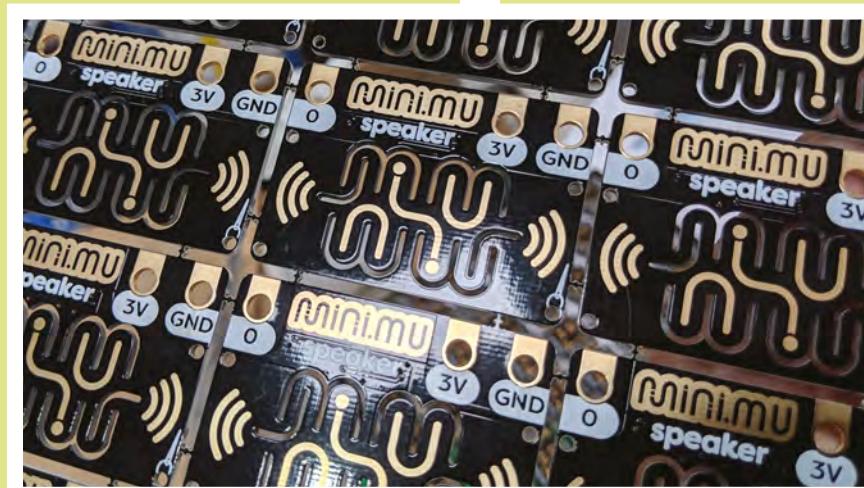
I couldn't stop thinking about this idea, so I contacted Imogen to ask if she'd mind if I designed a hack for kids, based on their technology. To my surprise, Imogen's response was to ask if I could make a

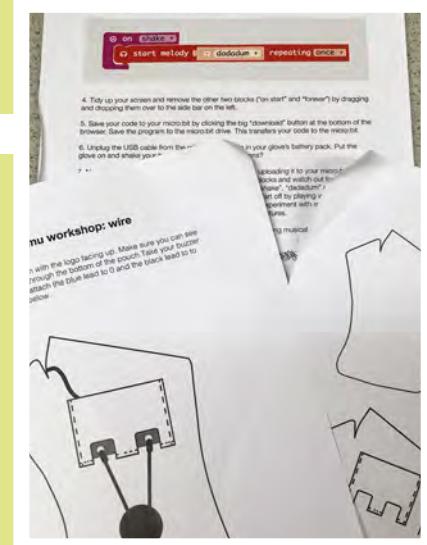
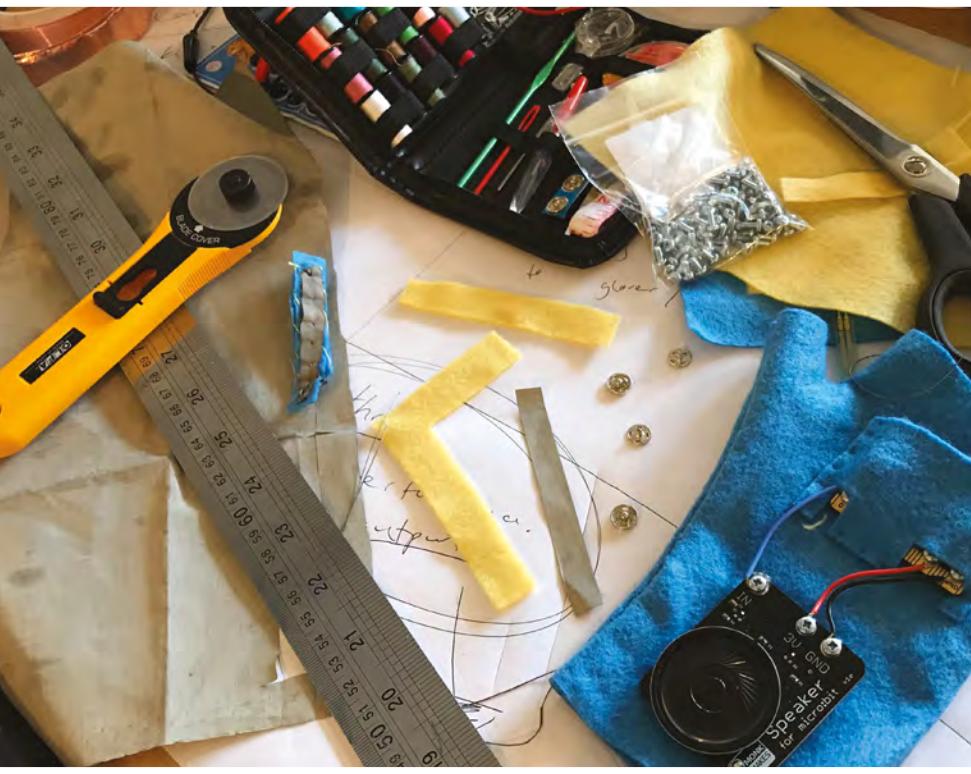
prototype to show for their big upcoming investor event, which happened to be in ten days. That's ten days to design, prototype, classroom test, and make a new product. Naturally, I said yes: who needs sleep when you can make musical instruments with lasers and sewing needles?

I needed to work fast and be very precise about what I wanted to make. The glove needed to be as cheap as possible, easy to make, and extremely fun. I needed to figure

out how to strip out a lot of the sensors and simplify the functionality so we could find a way to make the idea of a gesture-controlled musical instrument accessible to a lot of people. The most obvious sensor for gesture-based anything is an →

Below ◇
The PCB speakers are
easy to mount on fabric





Above Good documentation is a key part of a good product

Left Early prototypes used a more conventional speaker

accelerometer, which can sense where it is in space on the x-axis, y-axis, and z-axis, which are also known as roll, pitch, and yaw.

I'd been using the accelerometer-based gesture control functionality built into the micro:bit for a while, and I knew it was cheap, fairly indestructible, and came with a child-friendly way of coding it. The board does have limitations with the sounds it can produce, but the more I played around with it the more I came to love the retro gaming chiptune-style square wave bleeps and bloopes the micro:bit can make. To finish it off, I added a laser-cut felt glove pattern, a battery pack, and a simple speaker connected by just three wires.

All of the prototyping was done in Machines Room, my makerspace in London, with my friend Rehana Al-Soltane. Having access to a laser cutter, an electronics bench, and a decent sewing kit was invaluable for making rapid changes to the first prototype.

The first version of the MINI.MU glove was born in less than ten days, but straight

away we knew it was an idea that had potential. This was where my expertise started to falter: I'm a maker, not a product designer! I'd made and sold small batches of DIY kits out of my makerspace before, but nothing that required proper manufacturing, packaging, or distribution.

The level of interest starting to build around this project started to feel

This was where my expertise started to falter: I'm a maker, not a product designer!

overwhelming, so I reached out to someone who had done this before: the awesome Paul from Pimoroni. I met Paul first at a party I threw during an Education Technology trade show called Bett. I invited teachers, technologists, artists, and makers to my makerspace and we all played games, shared learning, and had epic fun thinking about how education should be.

As Paul remembers it, "A month or so later she got in touch to casually say 'Had

lunch with Imogen Heap. I have a great idea. Could I talk to Pimoroni about making it happen?'. We were keen, and super-happy that our work had not gone unnoticed."

After our initial chats, he invited me and Adam Stark, who heads up the MI.MU team, up to their factory in Sheffield to talk things over. The visit opened my eyes to the realities of how much work and organisation is involved in manufacturing and distributing maker products.

There was a whole massive wall of laser cutters constantly buzzing away, giant shelves stuffed with well-organised electronics parts, packaging materials, and cool trinkets. Even better, there was a team of nerdy, friendly makers with tea, biscuits, and jokes. A maker's dream, essentially.

Adam said, "we were blown away by their mixture of positivity and professionalism. Their facilities are amazing, and they have a team who go about their work with a sense of joy and optimism. Pimoroni gave us the vital link we needed to help us take our ideas and prototypes and make them a



reality that can be shared with thousands of people.” Pimoroni guided us through the issues of manufacturing the prototype I’d made at scale. As Paul explained, “there’s a mountain of difficulty you have to climb in making a product. Making one or ten of something is tough. Making 100 or 1000 is that much harder. Making more than that, and making the kit lovable, affordable, and interesting to shops, rather than just selling yourself, is really, really hard to get right.”

LEVELLING UP

“This was a glove that kids would use all over the world, so that meant that everything had to be quality. We had to make sure the felt had to have a proper Material Safety Data Sheet which says how safe and fireproof the material is, that the whole project had good documentation, and that all the boxes were ticked to allowed us to get crucial safety certification.”

The choices seemed endless at this point, from the dozens of types of cardboard boxes we looked at to the style

Right ◆
Pay attention to packaging, as it can be more complex than you might think



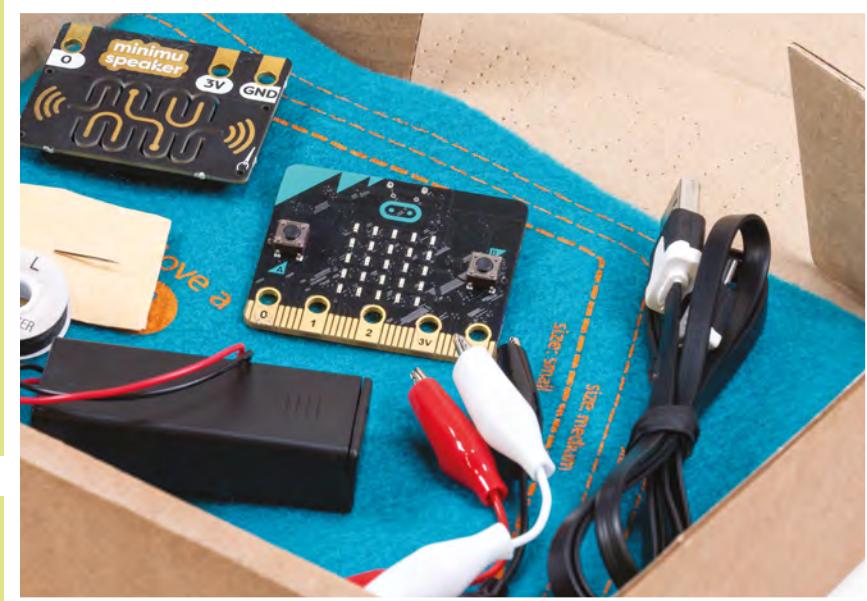
Left ◆
Choosing the right fabric for the glove

Above ◆
The author ready to lay down some grooves

of packaging, or the exact weight of felt and the size and shape of the needles supplied. We had to strike a balance between cost, practicality, durability, and design. Rachel Freire, the e-textiles expert and designer at MI.MU, explains that, “The MI.MU glove has to balance beauty, comfort, functionality, and robustness. This is a lot for a product to do! We had to make sure that the MINI.MU’s design aesthetics fitted with our vision

for MI.MU. Seeing the final design for the MINI.MU was super-inspiring and having it as part of our world is brilliant. I love that each glove is handmade by the user, so they really understand how it works. It brings us back to the core of why our gloves exist: to be accessible and inspiring.”

Before making the MINI.MU glove, when I thought about the cost of a product, I used to think about the cost of each of the things inside the box. However, much of the cost of a product can come from labour and machine use. For example, the original glove was laser-cut: perfect for prototyping but on a larger scale it would cost too ➤



Right ◆
The gloves can be personalised by the users to match their own style

much employee and machine time. Instead we decided to die-cut rectangles of felt that we then screen-printed with the pattern for kids to cut out themselves. This process could be automated far more simply and therefore turned out to be much more cost-effective for a bigger run of products. In any case, we realised that the screen print idea turned out to be a better design choice, as we were able to easily allow for different sized hands.

Solving the problems of making this kit, and getting an agreement, took a few months of researching suppliers and prototyping the look and feel of the kit. Alongside making sure it looked the way we wanted, we had to consider the price implications of every decision. It's not as simple as adding up how much everything costs and putting it on the market for a little more than that: you have to look at what people are willing to pay for the type of the thing you're making and make sure your product fits in the right bit of the marketplace.

We decided that we wanted to make the full kit available for under £40, including a micro:bit – not an easy task, especially when you take into account that a retailer

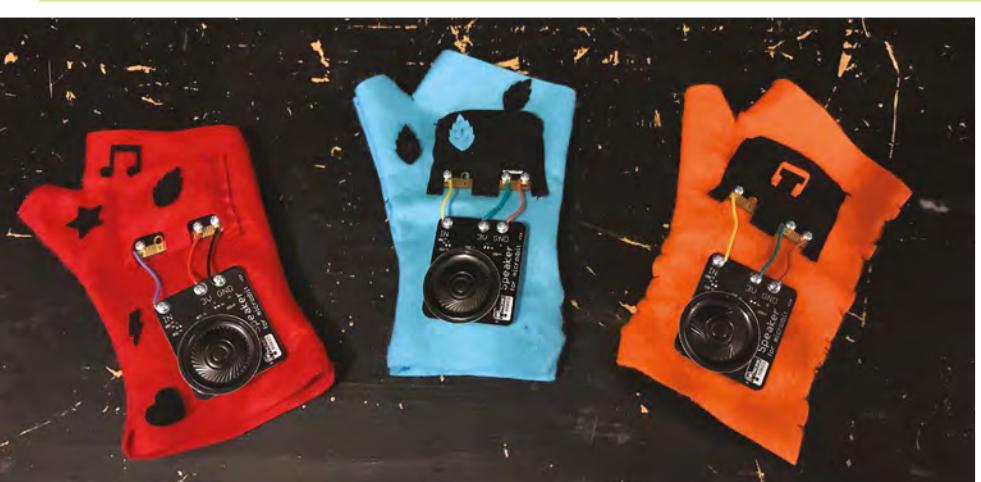


You're going to make it way better
if you ask the opinions of the
people it is for

will want to make a margin of 40% on that price. At every stage of the physical product development, I was working with young people, teachers, and families to make sure that what we were doing made sense. When you're designing something, it can be really easy to get blinkered and

keep going with your own ideas, but that is never going to lead to a great product. If you're designing anything, whether it's a mass-manufactured product or a service like an educational club, you're going to make it way better if you ask the opinions of the people it is for. It sounds obvious but you'd be surprised at how many people don't do it!

Something that helps me when I'm designing, especially for education and families, is to split the types of people I'm testing it with into two groups: users and clients. Users are the people who use the product – in this case, the children making the glove – and clients are the people who actually part with the cash to buy it – in this case, usually parents or a Head of



Left ◆
Testing the glove with larger speakers



Department in a school. To give your ideas the best chance of survival, you should make sure that you are making both groups happy. Another thing that's worth mentioning is that people – especially children – love to be asked their opinion. In schools, I ran workshops and explained that they were part of a special group of prototypers and testers that would help make this glove into a real product on a shelf. Every classroom and family I worked with took their 'job' very seriously, and many felt a great sense of empowerment and accomplishment because of the advice they gave us.

I tested the MINI.MU glove with over 250 different people in five sets of tests, and watched very carefully each time. How did people interact with it? What did they get wrong? What did they find difficult? How can we give the moments of joy we saw the best chance of being repeated for every child who makes the glove? Each time we tested the glove we changed something, and the end product is so much better for it.

IS IT TIME TO PRESS 'GO' YET?

By this point in the process, we were all very happy with the MINI.MU. I spent a joyful week in Sheffield finalising designs, pressing 'go' on the pick-and-place machine (#dreamjob), and working on the teacher resources with our education writer dream team including myself, Lorraine Underwood, and Tanya Fish.

Probably the funniest part of this week was an afternoon spent lying on the floor with my arms at strange angles, while acting

as the hand model for the packaging! We were almost ready to go, so we decided to finalise the paperwork. However, getting a contract in place turned out to be by far the slowest part of the process – taking almost as long as the entire rest of the prototyping and production process put together.

Adam, who heads up the MI.MU team, shared this advice about contracts, "Do not underestimate the time and complexity involved with getting a legal framework in place. We were 100% convinced Pimoroni were the right partner and just needed a simple, friendly, clarifying document on what we had agreed informally. So our task was easy, or so we thought! However, we still found that when it came to lawyers drafting contracts, there were a lot of hypotheticals to pick through. We also found that insurance companies place their own restrictions on what should be in a contract – so this should be taken into account."

After the contract went back and forth several times, we managed to get it signed and approved with just one day to spare before our planned pre-launch date on Ada Lovelace Day. It felt wonderful to finally set the MINI.MU free into the world, and the pre-launch itself was spectacular, with 80 gloves made by 80 kids in two days in London, and a simultaneous workshop happening on Imogen's tour date in Iceland.

Left ◇
Short crocodile leads are easy to connect and stay in place during use

GO, GO, GO!

As this article goes to press, we're taking delivery of the final bits and pieces to go in the box before it ships in the next few weeks. The sewable speaker boards have been delivered, the felt has been cut, and the packaging has gone to print. The biggest job we have left to do now is to start telling people about it! We've had a little bit of a head start with it going on Imogen's 40-date world tour, but getting a product off the shelves and into the hands of children is no mean feat.

We are so proud of what we've done so far. At MI.MU, we are all interdisciplinary artists and makers – we all work at the junction between art and technology. We hope that the MINI.MU can bring to children the idea that all creation is fun and they don't need to be put into the boxes that society creates for us. They can be both an artist and work with technology, that they can be creative with code and expressive with electronics. Above all, we hope they learn that they can achieve anything if they put their minds to it. □

Below ◇
The final kit boxed up and ready to ship



SAFecast: DIY RADIATION MAPPING

How citizen scientists at Tokyo Hackerspace took on the Fukushima nuclear disaster



Cameron Norris

[@cameronsnorris](#)

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

All images by Safecast
CC-BY-NC 4.0 unless otherwise noted

Right ↗
A bGeigie Nano removed from its Pelican hardshell

Safecast is an independent citizen science project that emerged in the wake of the Fukushima nuclear disaster to provide accurate, unbiased, and credible data on radiation exposure in Japan.

On 11 March 2011, an undersea earthquake off the Pacific coast of Thoku, Japan caused the second-worst nuclear accident in the history of nuclear power generation, releasing almost 30% more radiation than the Chernobyl disaster in 1986.

The magnitude 9.0–9.1 earthquake resulted in a series of devastating tsunami waves that damaged the backup generator of Fukushima Daiichi Nuclear Power Plant. Without functioning cooling systems, the temperature of the plant's many nuclear reactors steadily began to rise, eventually leading to a partial meltdown and several hydrogen gas explosions, launching nuclear fallout into the air and sea. Due to concerns over possible radiation exposure, the Japanese government established an 18-mile no-fly zone around the Fukushima plant, and approximately 232 square miles of land was evacuated.

INVENTION

The Geiger-Müller tube was named after Hans Geiger, who invented the principle in 1908, and Walther Müller, who collaborated with Geiger in developing the technique further in 1928, to produce a practical tube that could detect multiple types of radiation.

However, citizens of Fukushima Prefecture living outside of the exclusion zone were faced with a serious problem: radiation exposure data wasn't available to the public until almost two months after the meltdown occurred. Many residents felt they had been left to guess if dangerous levels of ionising radiation had contaminated their communities or not.

Alarmed by the situation, Dutch electrical engineer and computer scientist Pieter Franken, who was living in Tokyo with his family at the time, felt compelled to act. "After the massive wall of water, we had this invisible wall of radiation that was between myself and my family-in-law in the north of Japan, so that kind of triggered the start of Safecast," says Pieter.

Pieter picked up an idea from Ray Ozzie, the former CTO of Microsoft, who suggested quickly gathering data by attaching Geiger counters – used for measuring radioactivity – to the outside of cars





Above Pieter Franken, a Dutchman living in Japan, who helped start Safecast

Images
Joi Ito - CC BY 2.0

before driving around Fukushima. The only problem was that Geiger counters sold out almost globally in a matter of hours after the tsunami hit, making it even more difficult for Pieter and others on the ground to figure out exactly what was going on. The discussion between Pieter and his friends quickly changed from buying devices to instead building and distributing them to the people of Fukushima.

At Tokyo Hackerspace, Pieter – along with several others, including Joi Ito, the director of the MIT Media Lab, and Sean Bonner, an activist and journalist from Los Angeles – built a series of open-source tools for radiation mapping, to enable anyone to build their own pocket Geiger counter and easily share the data they collect. “Six days after having the idea, we had a working system. The next day we were off to Fukushima,” recalls Sean.

A successful Kickstarter campaign raised \$36,900 to provide the funding necessary to distribute hundreds of Geiger counters to the people of Japan, while training volunteers on how to use them. Today, Safecast has collected over 100 million data points and is home to the world’s largest open dataset about environmental radiation in the world. All of the data is collected via the Safecast API and published free of charge in the public domain to an interactive map developed by Safecast and MIT Media Lab.

EMBRACING OPEN DATA

“As we’ve grown, we’ve achieved global reach with crowdsourced measurements from over 100 countries, representing every continent. About 90% of our data has been collected by the most active 10% of our volunteers,” explains Sean. “The Safecast dataset also now includes data from far corners such as Sudan, Iraq, Antarctica, and the Marshall Islands, and sites of interest such as Chernobyl.”

The vast majority of this massive amount of data is collected using Safecast’s bGeigie series of pocket Geiger counters, designed entirely at Tokyo Hackerspace. All bGeigie devices use a standardised LND 7317 ‘pancake’ Geiger-Müller tube to accurately detect ionising radiation. Each data contributor is free to measure areas they consider worthwhile. “By measuring things, your perspective changes dramatically from doubt to certainty,” explains Pieter. “Our goal is to establish independently measured data about the environment, including radiation so that ordinary citizens can get an independent check, and regain some trust.”

In addition to the bGeigie mobile radiation sensors, the team has also developed Pointcast, a fixed real-time radiation monitoring system used to monitor how radiation levels change over time. Pointcast installations in Fukushima have been placed on homes in evacuated areas so that residents can keep track of radiation levels while they are away. It also provides a way to confirm government statements about how quickly radiation levels are declining.

“Researchers are now able to study radiation backgrounds in ways that were not possible before, and the public now has a vetted, trustworthy, independent source of information,” says Sean. “Nothing like it existed in 2011.” →



REMOTE ACCESS

It wasn’t until 2017 that engineers were able to pilot small remote-controlled robots into Fukushima Daiichi’s flooded reactor buildings. Six previous robots had already succumbed to obstacles and the intense radiation levels. Levels at one spot inside the reactor were estimated to be 650 sieverts per hour, which is less than one-tenth of the radiation levels inside a running reactor, but still enough to kill a person almost instantly.



Above Installing the pancake Geiger-Müller sensor into the bGeigie Nano

Below Safecast volunteers, including Pieter Franken and Toshikatsu Watanabe

BGEIGIE NANO MOBILE RADIATION SENSOR

Keep track of radiation levels wherever you are



Above The main board of the bGeigie Nano

Right The first bGeigie hit the road in April 2011. Since then, over 1500 devices have been assembled by volunteers

The Safecast bGeigie Nano, developed in collaboration with International Medcom, is a mobile GPS-enabled logging radiation sensor intended for cyclists and pedestrians. It is small enough to fit in the palm of your hand and can be assembled in four to five hours by someone with basic soldering skills.

Parts used include an Arduino Fio, GPS Bee, OpenLog, Inspector Alert Geiger counter, GPS receiver, Bluetooth module, and LND 7317 'pancake' Geiger-Müller tube detector. You can buy the parts online from the Safecast site as a kit, or source them yourself using the freely available bill of materials and assembly instructions found on GitHub.

"What we did was not unique per se. Geiger counters are not new, GPS is not new, Arduino is not new. But putting it together and making it accessible to local communities and doing it in such a way that it can be shared easily and openly, that was the breakthrough – making it super-simple so that even kids can do it, that is the big difference with what other people were doing," explains Pieter.

When gathering data, a toggle enables users to quickly switch between two modes of operation:

ALPHA AND BETA

Alpha radiation occurs when an atom releases alpha particles due to radioactive decay. Alpha particles are unable to penetrate our outer layer of dead skin cells but can cause severe cell damage if ingested. Beta radiation involves either an electron or a positron being emitted from an atom, and this type of radiation can penetrate the skin a few centimetres. However, the main risk still comes from accidental ingestion.



'recording mode' saves radiation measurements to a microSD card every five seconds along with the time and GPS coordinates, enabling researchers to track how ionising radiation concentrations change over time. 'Display mode' provides additional live radiation readings without saving the data to the microSD card. When fully charged, the bGeigie Nano runs for an impressive 35–40 hours in recording mode.

STRINGENT TESTS

bGeigie Nano units have been regularly put through stringent calibration tests at QualTek in the US, at the Jülich Research Centre in Germany, and the IAEA testing laboratory in Seibersdorf, Austria. In all cases, the measurement accuracy was shown to be +/- 10% typical, +/- 15% maximum – an excellent performance. The main unit can also be taken out of its protective case and used as a surface contamination spot meter, further increasing the device's ability to detect alpha and beta radiation.

Data collected by the Nano can be submitted to the official Safecast database through the API using XLoader or the Safecast Drive app, once the Nano has been connected to a smartphone via Bluetooth.

"The educational side is central. We want to build a culture of people who know about radiation and know how to measure it and know how to use that data somehow in their daily life," says Azby Brown, Safecast's outreach coordinator.

POINTCAST

REAL-TIME RADIATION MONITOR

Monitor places long-term

In March 2015, Safecast launched a fixed near real-time sensor network called 'Pointcast.' Pointcast sensors detect radiation levels and publish the updates online. Currently, Pointcast is the only independent near-real-time radiation monitoring system in Japan designed to detect radioactive particle releases from the Daiichi plant.

Pointcast sensor stations include a separate sensor and a communication module. The sensor module contains two Geiger-Müller tubes – one that measures the radiation dose rate equivalent in sieverts per hour, and one 'pancake' tube to measure the combined alpha, beta, and gamma activity in counts per minute. The Pointcast sensor's 'pancake' tube and i-Rover high-voltage supply are the same as the bGeigie Nano, which provides good equivalency between the fixed and mobile sensors.

RADIATION DATA FROM DAIICHI

Pointcast sensor modules are installed outdoors, while communication modules are generally placed indoors and connected by a cable. The indoor module has a large OLED display, which shows the outside radiation levels, and an alarm function provides a warning if radiation levels suddenly increase. Data is measured and transmitted every five minutes to dedicated data transmission and visualisation systems, which store the information and ensure it is publicly available.

The data communication unit was designed by Safecast volunteers to fit inside a rugged weatherproof case. The CPU board uses a Teensy 3.1 board for the main processor, and an integrated SD card stores configuration and measurement data to prevent data loss, should communication failure occur.

Right
The Safecast team visits Fukushima Daiichi for the first time

Below
A fully assembled bGeigie Nano kit, housed inside a Pelican hardshell



LONG ROAD AHEAD

"TEPCO, the power company that operates Daiichi plant, estimates it will be 30 years before it will be completely decommissioned, and during that time the balance between removing the contamination and keeping it from spreading will continue to be precarious."

Sean Bonner, Safecast Co-founder



MODULAR AND MOBILE!

"Initially we expected that most deployment sites would use a fixed Ethernet connection. In Japan, however, many people have abandoned fixed phone lines and fixed internet access altogether, and exclusively rely on mobile phones for internet access. Few locations available to us in the Fukushima exclusion zone have fixed internet access either," the team explained in the Safecast 2017 Report.

With this in mind, they decided to focus their attention on providing more 3G-capable Pointcast systems, and the majority of systems deployed since the summer of 2016 have a modular design that allows fixed Ethernet, cellular 3G, WiFi, Bluetooth, or other point-to-point or mesh communication hardware to be used. →

MEET SAFecast's MOST RECENT DEVICE: **SOLARCAST**

Radiation monitoring powered by radiation



SAFecast

Below A Solarcast Nano prototype gathers radiation data in Washington state



With the deployment of the Pointcast network, Safecast realised that requiring an experienced team member to be physically involved with each installation slowed down

deployment too much. Their collective experience with the bGeigie proved that the simpler the process of getting data from devices to their database was made, the more data would be collected. This sparked a discussion about what design parameters should be adopted to overcome such problems.

CONTINUOUS IMPROVEMENT

Ray Ozzie, who took the lead of Solarcast's development, concluded that a completely wireless, self-powered, auto-configuring device that could be "dropped anywhere and forgotten" was needed.

Because the idea was to create the simplest system possible, the project was initially called 'Simplecast' but as it became clear that the independent power aspect was the key technical challenge, it was renamed 'Solarcast'.

Similar to the Pointcast system, Solarcast contains dual LND 7317 'pancake' Geiger-Müller tubes and dual air particulate sensors: the Alphasense OPC-N2, and a Plantower 5003. These sensors work to validate each other and provide useful real-world data for future devices and deployments.

The device can run for over ten days on a full charge with no

sunlight, and run indefinitely with only a few hours of direct sunlight each day, thanks to dynamic power-adaptive sensor-sampling scheduling algorithms.

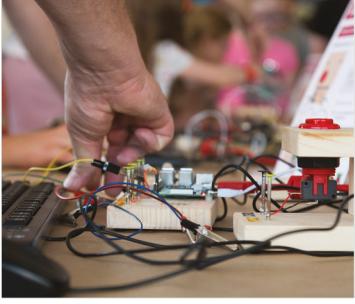
The Solarcast is also LoRa-capable, and can use any existing LoRa or LoRaWAN gateway it detects. LoRa is a digital data communication technology that enables very long-range transmissions with low power consumption. LoRa utilises licence-free sub-gigahertz radio frequency bands, and has a range of up to 20 km outdoors; indoors, the signal can penetrate concrete walls at the expense of distance, making it particularly useful in situations where a cellular connection is weak or non-existent.

An initial run of 30 Solarcast units was produced in Slovenia in 2017, and these are gradually being brought online in test deployments in the US, Europe, and Japan.

"As we look ahead a year, five, or ten years down the line, we know we'll need to make some bigger investments to ensure these systems continue to function as needed, when needed. And we continue to look for partners to join with us on this mission," says Pieter. Find out more about the Safecast project, and access their interactive map of radiation counts, at: blog.safecast.org. □

GRANITE WORRIES

While exploring the Safecast map, I discovered that unusually high levels of radiation had been detected at a Napoleonic fort near my hometown in Kent. Thankfully, Sean Bonner was available to provide me with answers: "No need for concern, granite is naturally radioactive, and many older structures and monuments that are made out of rock have higher readings than surrounding areas."



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 magazine meets...

Drew Fustini

Put your hands up if you love open-source hardware...

If you're interested in open-source hardware and making your own devices, you've probably heard of OSH Park. It's the biggest (in the US, anyway) PCB manufacturer for hobbyists, and it makes perfect, purple PCBs.

Its representative among us is Drew Fustini, who spends a whole bunch of time visiting cool places and playing with excellent open hardware. We sat down for a chat with Drew, who let us in on the secrets of PCBs, why all the cool kids are making hardware instead of software these days, and how nuclear physicists are helping you with your next electronics project. →

Right ↗
PCBs don't have to be covered in squiggles – you can make art with them too

Credit
Jun Shena
junshena.com





Drew Fustini
OSH Park



Above

Drew's working on an e-paper badge based on an OSH Park customer's design

Credit

Jun Shena
junshena.com

HS Let's begin at the beginning: what does OSH Park do and how did it get started?

DF The company came out of an enthusiasts group in Portland called Dorkbot, which is a monthly meetup of electronics enthusiasts. It's kind of similar to a makerspace or hackerspace, but it's only a twice a month sort of meeting. Their tagline is "people doing strange things with electricity"...

In 2009, the person who started OSH Park – his name's James Neal but he goes by Laen, which is Neal spelled backwards. (He's a systems administrator in Portland, and there's another sysadmin in Portland called James Neal, so he started going by Laen.)

Anyways, him and some other people from Dorkbot PDX realised that if they took their small boards and combined them together they could save money, so when you go to a circuit board fab and you want this or that, whether it's a bigger board or a tiny board, they all have the same fixed setup cost.

We fit smaller boards together to make a bigger board. The circuit manufacturing fab sees that as one board, so that fixed cost is then divided across all the different projects on that panel.

The amount of time it takes for the cam engineer, the various manufacturing engineers' job, that's a fixed cost, and instead of you paying that or me paying that, we combine all our orders together into one big board and then share those fixed costs across the panel.

For the first four years, it was an informal thing. It took them many months for them to fill their first panel – a panel is 18 by 22 inches, so that's almost a thousand square inches. With just a small hobbyist group, it took them a long time to fill up a panel. They put up a webpage, and you could email Laen and send him your file. Then every one or two months, they would send a new panel.

It was a very small thing initially, then it picked up more steam and

became popular. He was spending his time as a UNIX sysadmin at Nike, and he was spending all night doing orders and dealing with customer support. So he decided that he either had to stop doing it or he had to turn it into a company. So, he decided to quit his job and make it a company. And that was the point where we needed a new name, because Dorkbot was the name of the meetup.

Technically, OSH Park stands for Open-Source Hardware Park, but the idea is that you can upload your design to our website. By default everything is private, your profile, but if you want you can share it so that other people can download the file and order it from us, so it makes it really easy. If you have a blog or a forum post where you can link directly to the page to order the circuit board, if you have documentation

all telling the machine where to etch and expose the copper, where to drill, and stuff like that. So this is what comes out of your circuit board CAD software; whether you're using Eagle or KiCad or RS DesignSpark, they're all producing these Gerber files, and that's what gets uploaded to our website.

Except in the case of Autodesk Eagle or KiCad, where the user can just upload the actual design file, which is kind of nice because they don't have to figure out how to go through that step in their software of producing the Gerber files.

One of the fun things we have in our office is a panel 16x22 inches stopped at every spot along the way. It starts out as copper-clad, a big giant board with copper on both sides, fibreglass board, and the next one is where its been exposed by the mask, and then the copper has been

etched away that isn't supposed to be there, leaving just the circuitry that you designed, and then the next one is where it gets drilled, and then it gets plated, and then it gets the solder mask applied, which is the purple for us or green for most circuit boards, and then finally you get the silk screen applied.

[When you upload a file to us] we show you an overall look at what the board will look like when you hold it, but we break that down into the individual layers because circuit boards are all about layers.

For example, you might have a pad that you're going to solder a component on, but if that opening's not also in the solder mask layer, you're going to get solder mask covering up your pad and you won't be able to add solder to it. That's why we show the different layers. It's much more intuitive than it sounds. If you just upload a file to it, you'll see the preview there.

HS It sounds pretty beginner-friendly.

DF Yes. The one downside of being a small company is that we have a lot of features that we like to roll out, and we don't necessarily have the staff to do it as quickly as we would like. The upside →

to being a small company is we're not going to be in a position where we outstrip our revenue, right? We're very conservative in that way.

I came from a large multinational corporation, and I like the fact that we're small, and we enjoy staying small and being methodical. Our motto is 'Perfect Purple PCBs' and we really mean that. We want you to get the perfect board. With a lot of services, it's super-cheap and you go into it thinking, 'maybe it'll work, but it's cheap so if it doesn't work, no matter'. We have two full-time support engineers, so our model is if you have any issue at all, email us, and we'll help the customer with whatever issue they're dealing with. Sometimes the fault is on the designer side, but we always try to help people.

HS Who's your typical customer at OSH Park?

DF We [started off serving the needs of] hobbyists, but we also have people who are designing a circuit board for the first time, we have students, people who are engineers at large companies as well. We have an expedited service that's five days, and if that fits within the timeline for their project... obviously some of these big companies can afford to go to a quick turnaround PCB facility. They're going to pay a couple of hundred dollars for a board like that for a one- or two-day turnaround.

And then the other reason I think is that we're very inexpensive: we're \$5 a square inch and you get three copies, so that's well below probably the approval limit for most engineers, so they can just order without having to get a purchase order and go through that headache.

HS And even if it would get approved, the finance person is on holiday so nothing is going to get signed off until next week, then it will be at the back of the queue and by then you could have had it done...

DF So, especially for smaller boards, basically anyone who is getting custom

circuit boards made is our customer. The most popular shared project is a board for doing an e-cigarette thing, one of those vaporisers, because there's a forum post on one of these e-cig forums that links to that project. "Hey, you wanna build your own e-cig vaporiser? Here you go."

Some of our customers have not designed circuit boards. Other ones are ordering projects that they may have found directly on our website or linked to our website from a different project website, like HackSpace, like GitHub.

Some people are professional PCB layout specialists, people who specialise in laying out PCBs. Big companies, there'll be a person who does the schematics, there'll be a team that does the layout of the traces, so we also

Whether they're a professional or a hobbyist, people love it when they get their circuit boards in the mail

have people like that. We have a large gamut of customers. We want to give individual attention as much as we can to our customers. That's why we've been successful.

It was really nice when I joined, because I was not used to having a lot of personal interactions with customers.

At my last job, it was a very large company, I worked on our website: transaction and commerce website. I knew people were ordering from us, but only occasionally would I interact with someone who knew about them.

It's fun working with more people now, though. One reason is that most people are getting a board made for a project that they're very excited about. Whether they're a professional or a hobbyist, people love it when they get their circuit boards in the mail. It's a joy every morning when I wake up and look at

the company's Twitter and see all these people saying "got my boards from OSH Park, now I can build my project." That's been the funnest thing for me.

HS Is there a lot of remixing of projects going on in the OSH Park community?

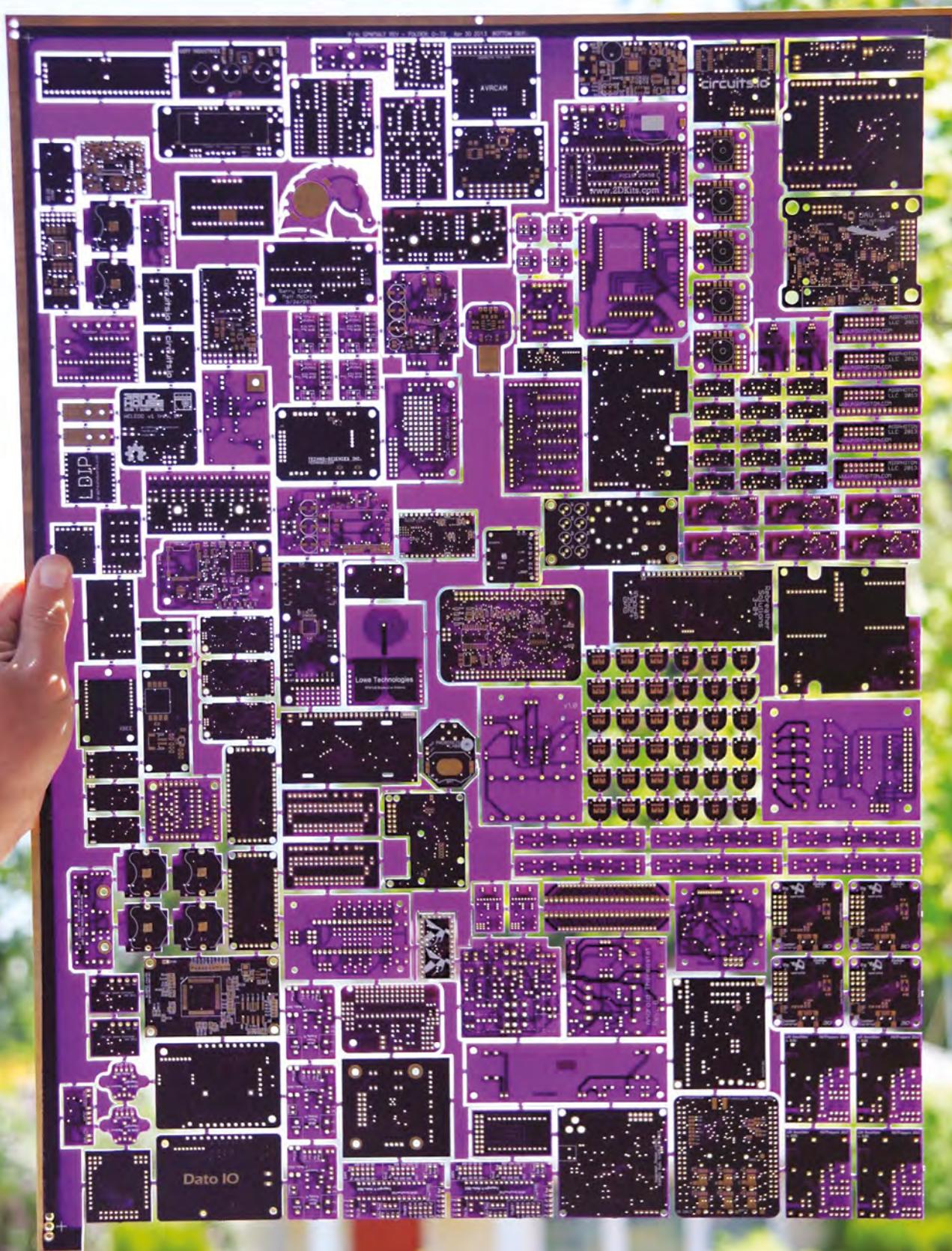
DF I do it myself with customers' projects. I have this nice thing where we all feed on each other. Someone does a project, I make my own version of it, someone might see that and make their own version of it.

A couple of hours ago, I ran into someone who uses the LilyPad, which is open-source hardware, and she was saying: "I wish that it had like two more holes." And the cool thing about open-

source hardware is that you can take the LilyPad – I think it's done in Autodesk Eagle, which is a beginner-friendly program (well, no CAD software is ever really that friendly, but there are lots of resources available because it's very popular. Adafruit uses it, SparkFun uses it, a lot of people use Eagle). But you can go into Eagle, add in the two more holes, get it made. And that's the wonderful thing about open-source hardware – you don't have to create the whole thing, you can just open it up and make your one little change and have the thing that's correct for your use case.

One of the things that drives us is our ability to panelise efficiently. You can see if you look at our panels. I would love to show you pictures of our panels, but they have our customers' boards on them and we don't know which ones are shared projects or not, so we don't share them. It's really cool because you have all these projects stick together on one board.

Any wasted space affects our profit, because the more boards we can put in there, the more we're dividing that fixed cost. The more orders we get, the cheaper it makes it for everyone. So our efficiency has improved over time, as we've gotten better at using different packing algorithms. →





Left ↗
Drew makes open-source hardware at OSHPark and (among other things) is on the board of the BeagleBoard Foundation

Credit
Jun Shena
junshena.com

One of the things that the fabs do do, which is quite interesting, is they have each of the layers of what your circuit board should look like and they have a machine called automatic optical inspection equipment, or AOI, that looks at it and then compares it to the image that it should be, and if there's any deviation it flags that. So generally we have really good... like 1 in 10 000 is a bad board, in terms of... I mean there are a lot of bad boards that are identified at the fab, and they remake them, which is like the whole point of the QA process that they do. But in terms of us getting a board and shipping it to a customer without knowing it, it's 1 in 10 000.

It's really frustrating for the customer if they get a board that has a defect, especially since most of these things are custom-designed, and most people blame themselves first. But it's really rare. We pride ourselves on quality. There are other services that may be cheaper that have less good quality, but the thing is when you have less trustworthy providers you never know whether it's a problem with your design or the manufacturing process...

With these two-layer boards, it's usually pretty obvious, because you can look at it

and see the circuitry was etched, drilled, and manufactured correctly. There is a different technique for four-layer circuit boards – which is what PocketBeagle is, for example; for that there is electrical testing that they can use, there's a machine with a bunch of little probes and it does continuity tests, but that adds cost, so we don't do that for everyone.

Since we've been around for nine years now, one of the big advantages we have is that we've worked out a lot of these different quirks, in terms of the manufacturing process and also the CAD tools side of things. I had no idea how many people it took to make a circuit board until I started working here.

On the project side of things, it's really fun to see people share their projects online.

HS How many do get shared, do you reckon, roughly?

DF It's really hard to say, because, especially for the team that de-panels, there'll be really interesting looking things on the panels – maybe it has the name of a company that's exciting – but unless they post about it online or write and tell

us, we don't know about it. There are a few times when we look back and think "I wonder what that was for?"

We have a sharing URL, but a lot of people will also share on GitHub or their blog, and we have no way to track that. But it's definitely a lot – I stay busy looking at all the projects people have shared on the internet.

It's really fun to design something in software and get the physical manifestation of it back. I think that's a reason why people do 3D printing so much: you design it in software and then the object appears, and that's a really powerful thing. I came from a software background, and it feels very abstracted a lot of the time. For me, the reason why I like embedded systems and electronics is that you design stuff, you write code, then code runs on something that you've soldered together or you've designed, and you can go into CAD software and draw these little traces.

PCBs in particular are a lot of fun because it allows you to take a schematic, which is a very abstract thing and turn it into something physical that you can personalise. Some of our customers are electrical engineers, but a lot of them are

Right ↗

PCBs go through several phases of manufacture, taking them from smooth copper board to finished product

coming from a software background. For a long time now I think, people that would have gone into electronics previously have been steered into software. Partly because I remember being frustrated that a lot of electronics classes that I used to see in the curriculum were not scheduled any more. Which makes sense, because there are more employment opportunities in software in general. But a lot of people went into software over the last 20 years.

It's one thing to see your program running on a dev board; it's another to fire up a circuit and see your code running in a circuit that you created. That was really eye-opening. I could take a bunch of components and put them together and make whatever I want. I was using breadboards and protoboards, which are... breadboards are nice for making sure your circuit works, but they suck because all the wires fall out.

HS Then there's perfboard, which is ideal for some scenarios but I wouldn't want to use it if I was going to make more than a couple of circuits.

DF Yeah, you just have a grid, like a breadboard, but you don't have anything connecting anything, so you have to solder everything. Whereas one of the nice things with CAD is that you just design the schematic, then you go in the layout and you just place all the components and then you're done.

HS What software would you recommend people get started with when they're designing a circuit board?

DF The more approachable CAD software like Autodesk Eagle, like KiCad, which is free software, is getting better. I'm a big fan of KiCad; I used it for all of my most recent projects. It's free and



open-source software. Autodesk Eagle does run on Windows and Mac; it's cost-free to use non-commercially, but it's not open source. It has a number of limitations on layers and size of boards. It's still the most popular software that our customers use, but KiCad is picking up a lot of steam, which I'm very excited about. One of the reasons is that some

other than Apple, all phones run Android, which uses the Linux kernel, because no-one's going to rewrite all that. There are always rumours of a secret project from Google that's going to replace the Linux kernel, but I'll believe it when I see it because there's a lot of hardware enabling in the Linux kernel that would have to be reproduced.

So, same thing with CAD tools, right? KiCad's been around for long over a decade. And it was like this kind of weird niche thing that a bunch of weird free software nerds used, and it crashed a lot. There wasn't even a release – you'd just get the nightly build out of a version control system and it was mostly just used by the people who made it. But then at some point, it transitioned to the point where it was a thing that non-KiCad developers were using, normal people. And then it got a lot better.

One of the things was CERN, the physics lab in Switzerland, they decided that they were going to start using it for all their designs, and they were going to donate money and hire people to work on it to add the features they needed to build some of the higher-end, more complex digital designs that they needed for their particle accelerator. □

No-one writes their own kernel any more, you just use Linux, because why would you be a crazy person and rewrite all the drivers?

organisations have, instead of paying for proprietary CAD packages, switched over to helping to fund development of KiCad.

HS That's just the way it should have been all along.

DF Yeah, and it's the way software has worked for a while now. No-one writes their own kernel any more, you just use Linux, because why would you be a crazy person and rewrite all the drivers? Which is why almost all phones use Linux;

PAPER PLATES



You are not supposed to play with your food, but no one said anything about the utensils



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.

People have always needed to eat, and we've been eating food from vessels since time immemorial. But plates, as we know them today, were invented in European potteries sometime shortly after 1708. And, even though paper has been around for thousands of years, paper plates are a 20th century invention. The idea of creating plates out of paper came to American inventor Martin Keyes, when he observed workers at a veneer plant in New York eating lunch on thin waste pieces of maple veneer. Keyes has often credited the ingenuity of these workers as the inspiration behind the idea for developing plates using moulded pulp.

Keyes spent the next couple of years perfecting a machine that would mash wood pulp and mould it into paper plates. He had to then fight a court battle before he could patent the machine, since someone had stolen his idea and created a similar machine. As an aside, Keyes' courtroom battle underlines the importance of documentation, since it was his daily log, as he laboured to perfect his machine, that convinced the court of his claims.

Armed with the patent, Keyes set up a small mill and started producing paper plates in 1904. His first paper plate moulding machine was capable of

manufacturing 50 000 paper plates per day. It was, however, an unfortunate event a couple of years later that helped scale his business. The large-scale rescue efforts in the aftermath of the San Francisco earthquake in 1906 created a huge demand for paper plates, and led to Keyes setting up multiple manufacturing units.

These days, paper plates adorn kitchen cupboards around the world. They might not be as valuable as your prized china, but their convenience is unparalleled. Over the years, paper plates have become an umbrella term that encompasses all kinds of single-use disposable plates. Generally speaking, paper plates aren't very expensive, though their price depends on the material they are made from. Besides being made from plant fibres, the use-and-throw plates are also made from plastic and Styrofoam. Also, while plain white paper plates are the most popular, you can get them in a variety of colours, sizes, and designs.

In addition to their value-add to a picnic, paper plates have been put to various other creative purposes, thanks to their malleable nature and dexterity. In addition to serving a meal, paper plates are a staple in every kid's arts and crafts class. Here's how you can use them to nudge kids into budding makers.

PAPERCRAFT SPACECRAFT

Dennis is one of the most prolific authors at Instructables.com, and never misses an opportunity to participate in the site's regular themed contests: "Somehow I got hooked on creating new projects for every contest that came along, each one trying to surpass the last, a vicious cycle in over-achievement that calls on all your maker skills, and the tools and materials at hand." When the papercraft contest came along, he leaned on his geeky childhood, of watching shows like *Star Trek* and *Lost in Space*, to fashion a spacecraft from paper plates. He decided to create a model of the Jupiter 2 spacecraft from *Lost in Space*, perhaps because the paper plates are already in the shape of its saucer section. Dennis used

9" and 6" paper plates, and his Instructables lays down the exact process to transform them into the top and bottom sections of the spacecraft's saucer, complete with the skirt for the saucer's engine ring. After gluing the two together, along with a bottle cap for the dome, he cuts out the windows and equipment hatches, and uses folded pieces of paper to create the landing gear struts. □

"HE LEANED ON HIS GEEKY CHILDHOOD OF WATCHING SHOWS LIKE STAR TREK AND LOST IN SPACE TO FASHION A SPACECRAFT"



Project Maker
DENNIS LOUIE

Project Link
hsmag.cc/jysmKb

Left ↗
The spacecraft is lit with an Adafruit FLORA and a pair of NeoPixel rings that run the basic goggles sketch, which incidentally provides the same light animation as the original Jupiter 2

PAPER PLATE MARBLE TRACK

To fuel the need for speed of her four boys, maker mom Sarah has been building marble tracks using everything from paper towel tubes to pool noodles. Fresh out of ideas, she was about to buy the boys a wooden set when she saw an article in a magazine that created tracks

" SHE CREATED AN ELABORATE TRACK THAT WAS A LOT STEEPER – TO COMPENSATE, SHE ADDED GUARDRAILS ON THE TURNS "

using the rims from the paper plates. It took her a little time to adapt the idea into a track of her own. Sarah suggests using plates that have a design on them and a smooth rim around the edge. With a little trial and error, she created an elaborate track that was a lot steeper than the one in the magazine. To compensate, she added guardrails on the turns by turning the paper plate rims in the opposite direction. Sarah suggests building the track from the ground up and that the less steep it is, the easier it'll be to keep the marble under control. She also placed a lid from a jar at the bottom of the track to keep the marble from rolling onto the floor where her toddler could get to it. □

Project Maker
SARAH DEES

Project Link
hsmag.cc/zvsGzg

Below ♦
While the article suggests using different-sized paper towel rolls as supports, Sarah hot-glued her track onto cardboard supports



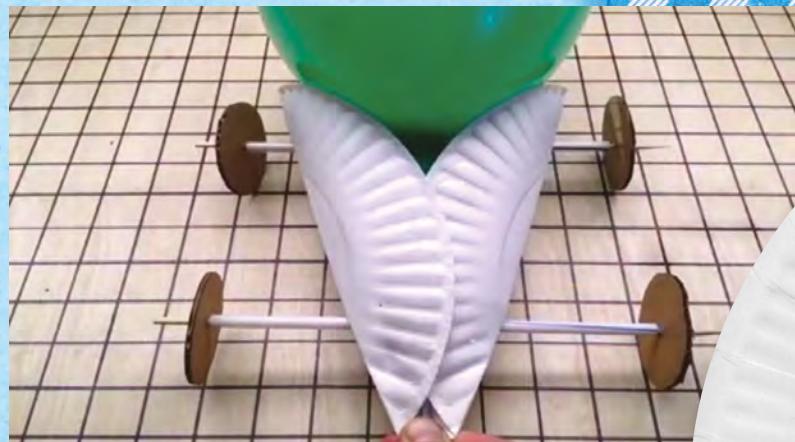
PAPER PLATE RACER

A

s the co-founder of DIY.org, Isaiah Saxon is well-known in the maker community.

One of the many hacks the filmmaker has posted on the website is a paper plate rocket car that's propelled by a balloon.

Isaiah begins by folding a paper plate into the body of the racer. He then uses some cardboard to cut out wheels and carefully places a couple of BBQ skewers into the centre to create a set of axles for the racer. The axles are taped to the bottom of the paper plate. To propel the contraption, he cuts the open end of a balloon and tapes a bunch of straws inside it to create a tailpipe. This tailpipe is then taped to the paper plate. To put the racer in motion, simply inflate the balloon through the tailpipe and let it rip in an open space. □



Project Maker
ISAIAH SAXON

Project Link
hsmag.cc/EMCvsL

Above ▷
Many characters in Isaiah's animation films indulge in some sort of DIY hacks. You can view his videos at isaiahsaxon.com

PAPER PLATE SKELETON

A

lthough Halloween's come and gone, former Instructables employee Karen has an easy and fun festive-themed craft – a paper plate skeleton that's not anatomically accurate, but

can be easily replicated by kids with minimal supervision and guidance. She used some paper plates that were 6" in diameter and some string to piece together a skeleton. Her Instructable has images of all the different body parts that she's cut out from the paper plates, including a skull, a shoulder, a ribcage, a hip, a couple of hands and feet, and eight bones. You can trace her designs into paper plates – but for the hands and feet, she suggests tracing your own. After cutting out the

Project Maker
KAREN HOWARD

Project Link
hsmag.cc/KIZVWa

Right ▷
The skeleton is easily customisable. Beef it up with more bones, or add a dash of colour



pieces, punch holes as shown in her images, and tie them together using a piece of string or twine. Karen suggests using some tape to hang the completed skeleton on a wall in any position you want. □

HackSpace

TECHNOLOGY IN YOUR HANDS

Download the app

Out now for smartphones & tablets



£2.29

rolling subscription

or

£26.99

subscribe for a year



FORGE

HACK | MAKE | BUILD | CREATE

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

PG
88

ROTARY TOOLS

Carve your designs with a mini spinny cutty thingy

PG
90

POWDERISING

Turn your foods into delicious powders and sprinkle on flavour

PG
92



K40 CONTROL

Use your network to load files onto your laser cutter

PG
96

ROBOT PART 2

Add some sensors to help your robot see the world

PG
78

SCHOOL OF MAKING

Start your journey to craftsmanship with these essential skills

- 78** Transistor circuits
- 82** Manual milling

PG
102

NEOPIXEL MASK

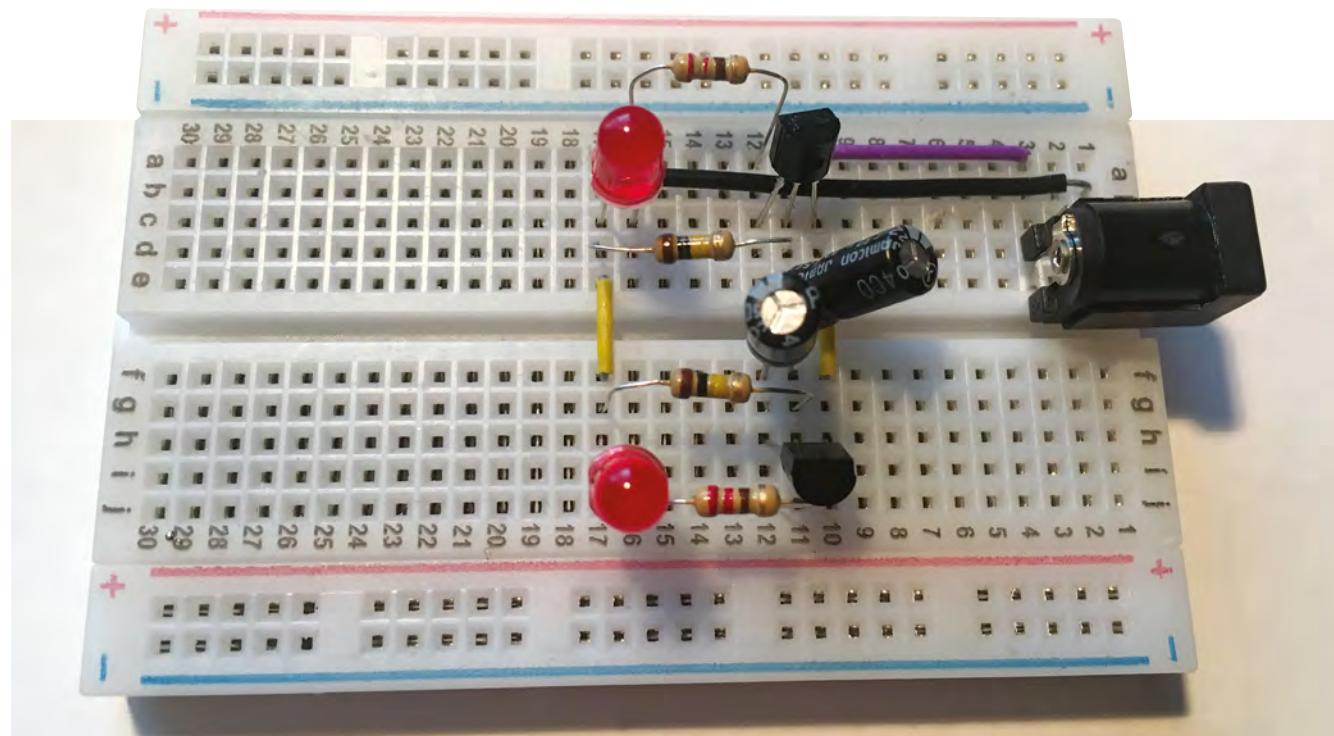
Use lights to make everyone jealous at your next party



PG
108

JUPYTER

Run CircuitPython interactively from your browser



Electronics 101.6: Simple transistor projects

Let's start putting theory into practice with these beginner circuits



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

In earlier parts of this series, we looked at the basic components, how they work, and how to use them. It's time to put that knowledge to use and look at some simple projects. To keep things simple, we'll look at circuits that run by themselves without external input. And to keep things interesting, we'll look at circuits that do something we can see. In particular, we'll start with a dual LED flasher.

ON AND OFF

In the earlier parts of this series, we explored electricity and basic components. You can take a look back at these if you need to, as we go over how to put this theory into practice.

The circuits in this part are all oscillators. Specifically, they are what's called astable multivibrators. That just means that they alternate

between two (in these circuits) states, without any outside stimulus.

Figure 1 shows the LED flasher circuit. Notice how the two halves of the circuit are identical, connected by the capacitors that couple the transistor collector of one side to the base of the other. Let's walk through how the circuit works. We'll start by assuming the Q1 is on (i.e. saturated) so that the voltage at its collector (and hence the positive plate of C1) is essentially 9V. Also, LED1 is lit.

1. C1 charges through R4, while the RC time constant of these two determines the rate.
2. As C1 charges, the voltage across it increases. Since the positive plate is fixed (by the saturated Q1) at 9V, the voltage on its negative plate is dropping relative to that 9V.

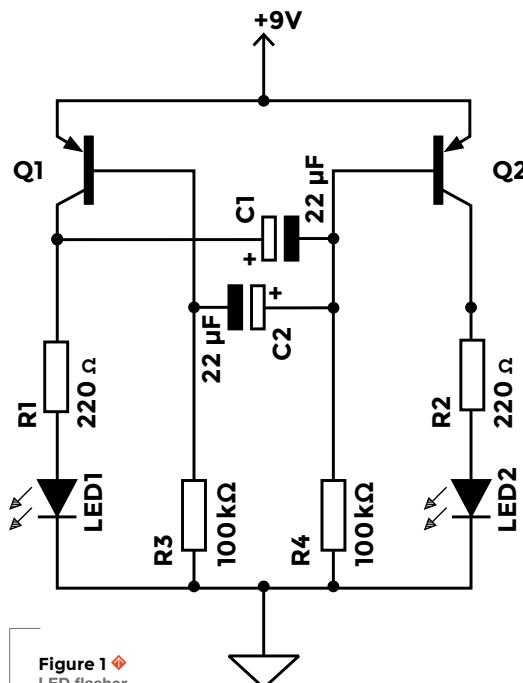


Figure 1 ◇
LED flasher

3. Eventually it reaches 8.4V. This means that the voltage on the base of Q2 is 0.6 less than its emitter. Since it's a PNP transistor, this causes it to saturate and switch on.
4. That causes LED2 to illuminate as well, setting the voltage on the positive plate of C2 to 9V.
5. Since any charge in C2 has long ago drained away through R3, the voltage across it is 0V, and so the base of Q1 is now at 9V just like its emitter, and that will turn it off.

6. We're now in the same state as we started, but with the other side of the circuit.
7. C2 now charges through R3 until there is 0.6V across it and its negative plate is at 8.4V.
8. Q1 then switches on, Q2 switches off, and the cycle continues.

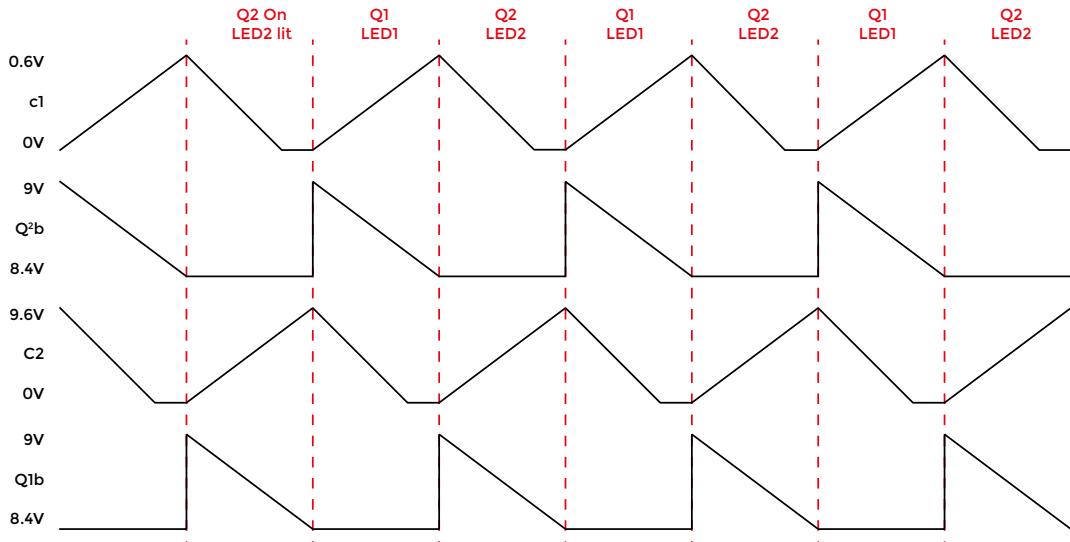
Put more simply: when Q1 turns on, it turns Q2 off and starts its capacitor charging. Eventually the capacitor charges enough to turn Q2 on. That turns Q1 off, and so on. Approximate signals for the voltage across the capacitors and on the transistor bases are shown in **Figure 2**.

TICK TOCK

Our next circuit is a metronome: it ticks regularly. R2 sets the frequency of the ticks, while R1 sets the maximum frequency; i.e. the resistance between Q1's base and +9V can't be less than the 10kΩ of R1. R2 lets the total resistance between Q1's base and +9V vary between 10kΩ and 60kΩ. That controls the time it takes for C1 to charge, and thus the frequency of the oscillator. The schematic is shown in **Figure 3**.

Let's start with C1 being completely discharged. That means there's 0V across it. The speaker has a resistance of 8Ω, which is pretty much nothing compared to the 10–60kΩ of R1+R2. This means that the voltage at the base of Q1 is essentially 0V. C1 charges through R1+R2 (and the speaker), →

Figure 2 ◇
Approximate LED flasher signals



YOU'LL NEED

- ◆ Solderless breadboard
- ◆ 9V power supply
- ◆ For the LED flasher
 - ◆ 2 × 2N3906 (or similar) PNP transistor
 - ◆ 2 × 220 Ω resistor
 - ◆ 2 × 100 kΩ resistor
 - ◆ 2 × 22 μF electrolytic capacitor
 - ◆ 2 × LED

For the metronome

- ◆ 10 kΩ resistor
- ◆ 50 kΩ variable resistor (potentiometer)
- ◆ 2N3904
- ◆ 2N3906
- ◆ 22 μF electrolytic capacitor
- ◆ 8 Ω speaker

For the siren

- ◆ 2 × 47 kΩ resistor
- ◆ 2N3904
- ◆ 2N3906
- ◆ 100 μF electrolytic capacitor
- ◆ 10 nF capacitor
- ◆ 2.2 μF electrolytic capacitor
- ◆ Momentary contact SPST push-button switch
- ◆ 8 Ω speaker

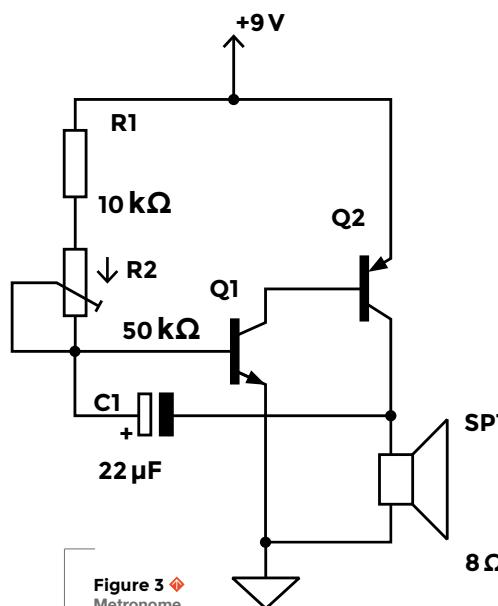
continues overleaf →

SCHOOL OF MAKING

YOU'LL NEED (continued)

For the voltage controlled oscillator

- ◆ 100 kΩ variable resistor (potentiometer)
- ◆ 2N3904
- ◆ 2N3906
- ◆ 10 nF capacitor
- ◆ 2.2 µF electrolytic capacitor
- ◆ 8 Ω speaker

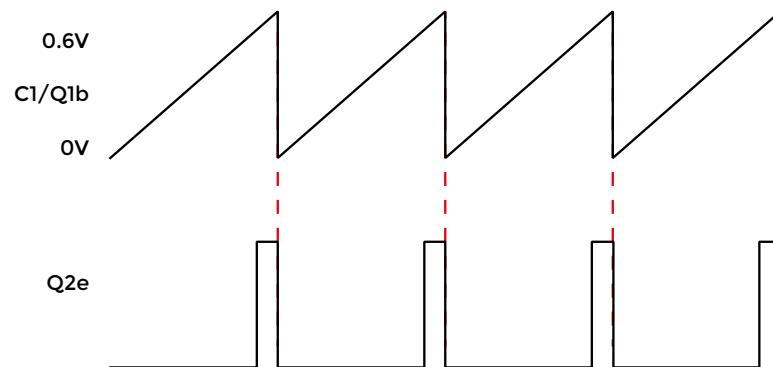


and when the voltage across it reaches $\sim 0.6V$, Q1's base-emitter junction is forward-biased and it turns on. This connects Q2's base to ground through Q1. That makes Q2's base far below the 8.4V needed to forward-bias its base-emitter junction. As a result, Q2 turns on. This does two things. First, it puts $\sim 9V$ across the speaker, making it twitch. It also connects the negative plate of C1 to the same voltage. That quickly drains off the electrons built up on it. That, in turn, means that the voltage across C1 goes to 0V, taking the base of Q1 with it. This turns off Q1, which then turns off Q2, disconnecting the speaker and C1 from +9V. The speaker snaps back and C1 starts charging again. **Figure 4** shows the approximate signals at C1 and Q2's emitter.

ALARMING!

Figure 5 shows the schematic for a siren circuit. The core of the circuit is similar to the metronome. The capacitor (C2) is much smaller, thus it runs faster, and instead of ticking now and then, it produces a tone.

Figure 4 Approximate metronome signals



This works in a similar way to the metronome. The two transistors work as an oscillator, using feedback through C2. This is much smaller than C1 in the metronome, so the frequency will be higher: tones rather than clicks. However, the theory is the same. The frequency of the oscillator is also affected by the charge on C1 (and thus the voltage across it). Remember that the frequency is determined by how quickly the charge/voltage of C2 brings the base voltage of Q1 to 0.6V to turn it on. Any voltage below that will keep Q1 off. The voltage on C1 affects the base voltage of Q1 in addition to the voltage on C2.

The frequency of the oscillator is also affected by the charge on C1 (and thus the voltage across it)

When S1 is closed, C1 charges through R2, and the voltage across it gradually increases. When S1 is open, C1 discharges through R1 and Q1. When the voltage across C1 is higher, it takes less time for the charging of C2 to forward-bias Q1's base-emitter junction (turning it on). That results in a higher oscillator frequency. When the voltage across C1 is lower, it takes longer and so the frequency is lower.

By alternately pressing and releasing S1, the frequency can be made to go up and down. The result is a warbling siren effect.

BALANCING ACT

We've looked at three different oscillator circuits. They all have things in common:

1. A capacitor charging (through a resistor), and the voltage across it controlling the biasing of a transistor
2. Feedback that causes the switching on of the transistor to drain the capacitor, which turns the transistor off

The resistor and capacitor combination determines how quickly the capacitor charges (this was discussed in HackSpace #10). This is shown in action in the metronome circuit by the use of a potentiometer to vary the resistance and thus the time constant of the RC circuit, i.e. how fast the capacitor controlling the base voltage, we can change the frequency of the oscillator.

CLASSIC CIRCUITS

Before we had Arduino and Raspberry Pis, electronics were built out of discrete components and simple ICs. Many people who learned electronics during this period (including the HackSpace mag editor) learned from the classic books sold through RadioShack by Forrest M. Mims III. Perhaps the most famous of these is Getting Started in Electronics.

More recently, Star Simpson brought some of the circuits in these books to life in the Circuit Classics series of soldering projects. You can get these (including the dual LED flasher that we've looked at in this article) at hsmag.cc/FVvkXM.

Changing the resistor is one way to alter the timing (and thus frequency) of the circuit. Changing the capacitor is another. Replace the capacitor(s) in the previous circuits and observe the effect on the operation. The LED flasher is especially interesting to play around with. With both timing resistors and capacitors of equal value, the LEDs are on/off equally. If the values differ between the two sides, the relative amount of time the LEDs are on/off will differ as well.

VOLTAGE CONTROL

The other way to control the frequency of an oscillator is shown in the siren circuit. In that case, the voltage from which the biasing capacitor is charging is controlled. Here it's the voltage on another capacitor which is charged and discharged using the switch.

We can use this idea to make an oscillator that's adjustable. See **Figure 6** for the circuit. This works identically to the two previous circuits, except that the frequency is controlled by a potentiometer that is working as a voltage divider. It serves to control the voltage that the capacitor is charging from, and thus the timing.

This is referred to as a voltage-controlled oscillator, or VCO, and is a building block of analogue synthesizers. You could easily use this in conjunction with the synth we look at in the cover feature this month, though we'll leave implementing this as an exercise for the reader.

Oscillators are a fundamental type of circuit. They can be found in many types of equipment, from a simple LED flasher to the clock generators of computers. We've looked at a few simple transistor-based oscillators that demonstrate the basic ideas. □

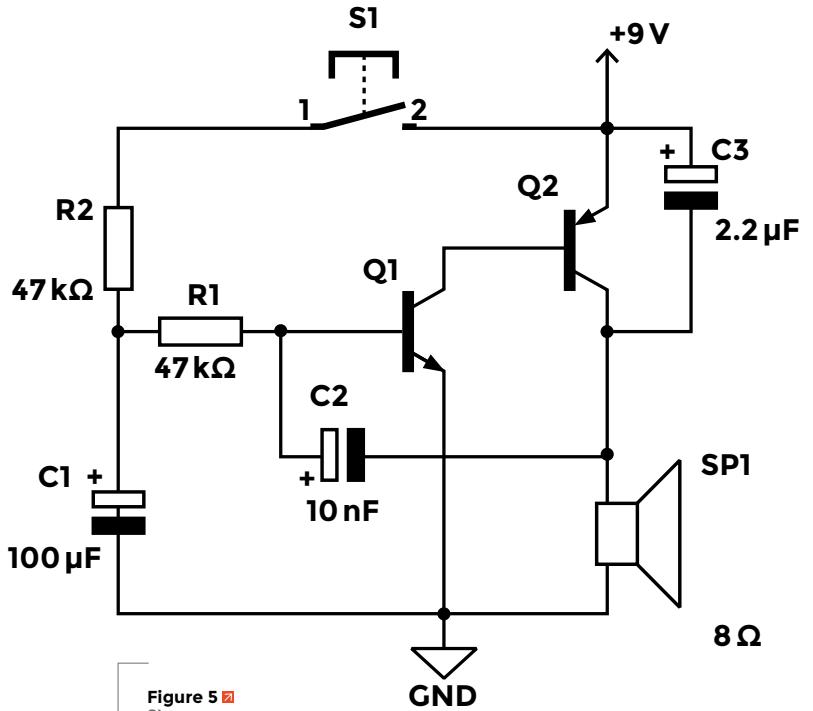


Figure 5 Siren

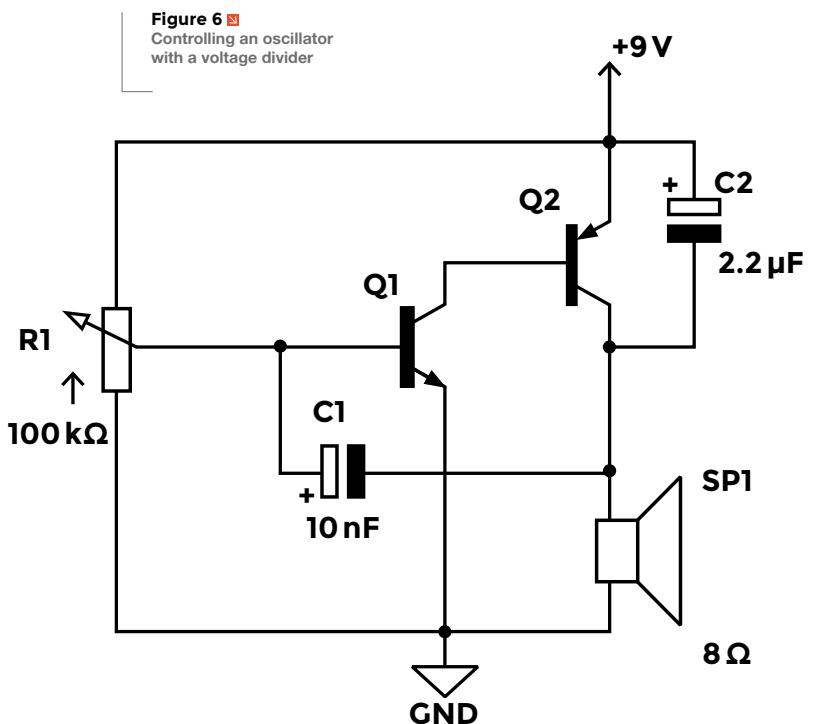


Figure 6 Controlling an oscillator with a voltage divider

Machining Basics: Manual milling

Trim blocks of metal until you've got the shape you want



Jo Hinchliffe

 @concreted0g

Jo Hinchliffe is a contributor to the Libre Space Foundation and sits on the UK Rocketry Association council. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open source. He also has a shed full of lathes, milling machines, and CNC kit!



Manual milling machines are versatile tools that often feature in hackspace and community workshops. While some spaces have a lot of people interested in machining, some don't, and so it often remains a mysterious machine not getting used. Manual milling machines are in some ways superseded by computer numerical controlled versions (CNC) and, whilst CNC routers and CNC milling machines are becoming more common, it's still really useful to have an understanding of how manual milling machines work. Many skills learnt on a manual mill later apply to CNC equipment and also, if you use both a manual mill and a CNC router, it's apparent that sometimes a manual milling machine offers some versatility hard to replicate on a CNC.

Milling machines come in a range of sizes and, generally speaking, larger, heavier machines will be more rigid and accurate and make larger cuts.

Many skills learnt on a manual mill later apply to CNC equipment and... offer some versatility hard to replicate on a CNC

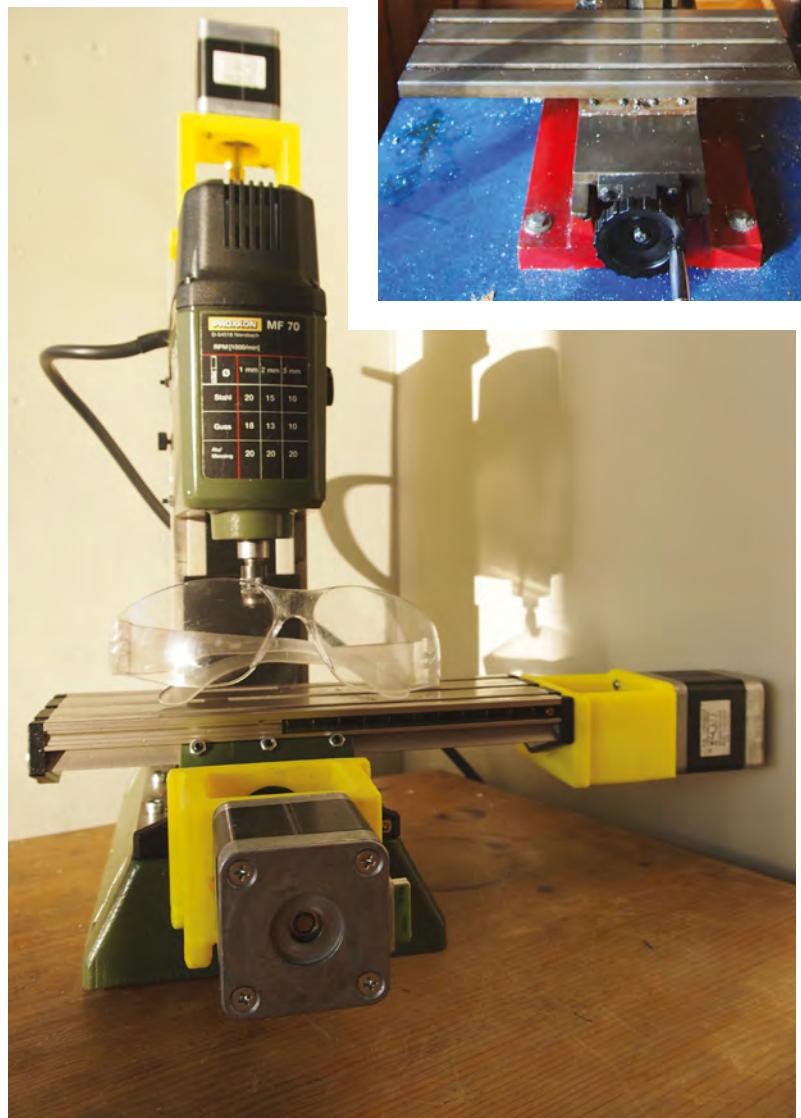
However, available space and budget may also dictate what machines people can purchase or access. The range of machines available is astonishing, from the tiny Proxxon MF 70 (**Figure 2**) through to the large classic Bridgeport milling machines, the dream machine for many a home machinist! A common small bench-top milling machine such as the Sieg SX2P (**Figure 1**) has reasonable capabilities when it comes to what materials it can handle. This machine will readily cut steel, cast iron, aluminium, brass, and plastics such as Delrin and nylon.

MANY FLAVOURS

Milling machines come in three basic flavours: vertical, horizontal, and universal. The Sieg SX2P is a vertical machine, and these are probably the most common in the smaller workshop or hackspace. It is described as vertical when the cutting tools are held vertically in the spindle, much like a drill bit is held in a pillar drill. Horizontal mills are where there is a horizontal shaft over the table (the moving area you attach a piece of work to), and a circular cutter is held on the shaft. →

Figure 1 ◇
A Sieg SX2P small benchtop vertical milling machine. This common home or hackspace machine can deliver great work across a variety of materials

Figure 2 ◇
Perhaps the smallest milling machine is the tiny Proxxon MF 70. This one is currently being modified into a CNC milling machine



SCHOOL OF MAKING

Finally, a universal mill has interchangeable milling head configurations so it can be set up as either a horizontal or a vertical mill.

MILLING MACHINE ANATOMY

The main flat section on a milling machine is referred to as the table and it can move in two axes: the X axis (left to right) and the Y axis (towards and away from the column). They usually have a collection of slots cast and machined into the table, which are used to help clamp the workpiece or a vice to hold

it down. Each direction of the milling table has a lead screw and nut arrangement underneath and a handle; turning the handle allows the table to be moved a known amount in the given direction. More advanced or industrial milling machines may have a powered feed assembly which enables the axis to be fed along under the power of an electric motor; this is to provide an even feed speed and saves hand turning, but stops short of being CNC control, which controls the position of the mill (we'll look at this in a future article).

The column is the heavy post which rises up behind the table; attached to this is a milling head that can travel up and down the column (the Z axis). On some mills the head is fixed and the table rises and falls, which is referred to as a knee mill, but these again are usually large industrial machines. Some smaller milling machines feature a tilting column which can offer some versatility in being able to make angled cut; however, some people find them very difficult to get accurately back to vertical and to 'tram' back to a perfect 90 degrees to the milling table!

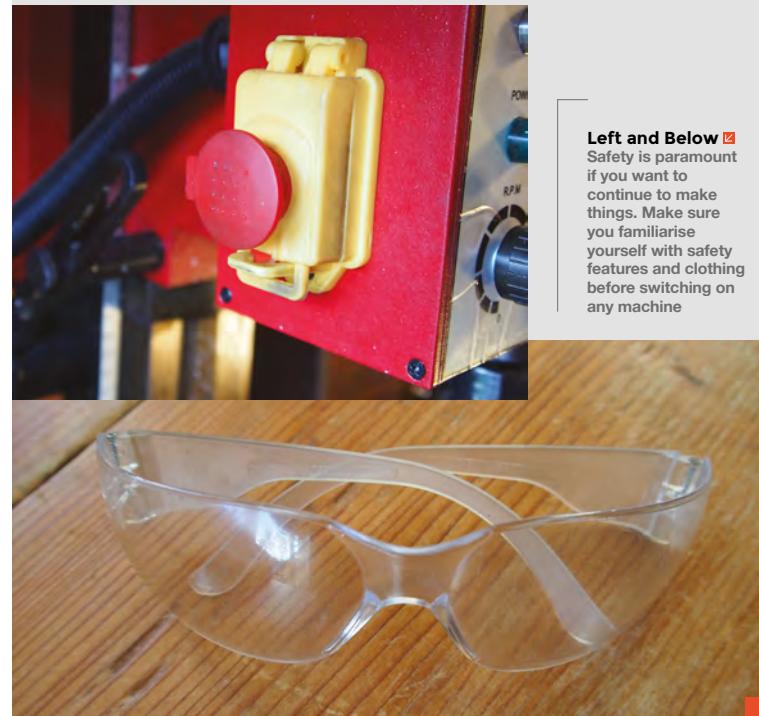
End mills are a common bit that can be used to remove material from the face or from the side of an object

STAYING SAFE

We need to remember that there are inherent risks when using milling machines: risks to eyes from flying debris – often referred to as 'swarf' or 'chips' – or even breaking tools, can be mitigated with good-quality safety glasses.

Entanglement in spinning chucks or tools is a real risk that has cost lives – care should be taken to ensure that hair, loose long sleeves, and jewellery are safely tucked away or tied back. Many industrial machines, and some in community workshops or hackspace, will have auto locks that won't allow the machine to run until certain physical guards/screens are in place.

Often workpieces and/or accessories can be very heavy and may require moving; opinions differ, but due to this, many organisations state it's a good idea to wear steel-toed shoes or boots to protect from drops and impact injury. Make sure if you are buying a second-hand milling machine that it has an accessible and working emergency stop switch: the ability to quickly shut down a machine is essential in an emergency. Before using a machine (even one you have used regularly), get into the habit of checking that the emergency stop button functions correctly.

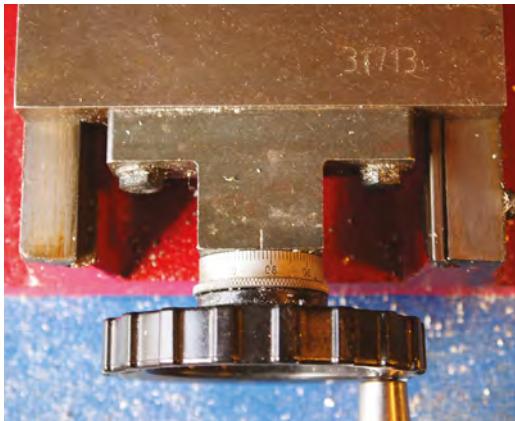


Left and Below Safety is paramount if you want to continue to make things. Make sure you familiarise yourself with safety features and clothing before switching on any machine

The milling head is the apparatus mounted on the column which connects to a motor system and contains the spindle; the spindle is the spinning section that rotates the tools held in it. There are different systems and types of spindles that use different types of attachment such as collets and chucks to hold tools.

Larger machines commonly come equipped with digital read-outs (DRO): panels displaying where each axis is positioned, making it easy to create accurate cuts. Many smaller machines do not come equipped with DRO, but will have graduated dials (**Figure 3**) on the handles that move the axis, allowing the machinist to know where they are.

Fitting either commercial DRO or DIY ones (often made from cheaper digital callipers) is a common modification for milling machine owners. It is also possible to use a dial test indicator (as in **Figure 7**, overleaf) clamped to measure the movement of the table to help maintain positional accuracy.

**Figure 3**

Graduated dials are used to keep track of the position during a machining operation or 'cut'. They often have a grub screw and after the tool is positioned, the dial can be rotated and set to zero. Once the grub screw is retightened, when we move the axis we can keep track of the distance travelled

**Figure 4**

Two common tools used in milling machines: a slot cutter and an end mill

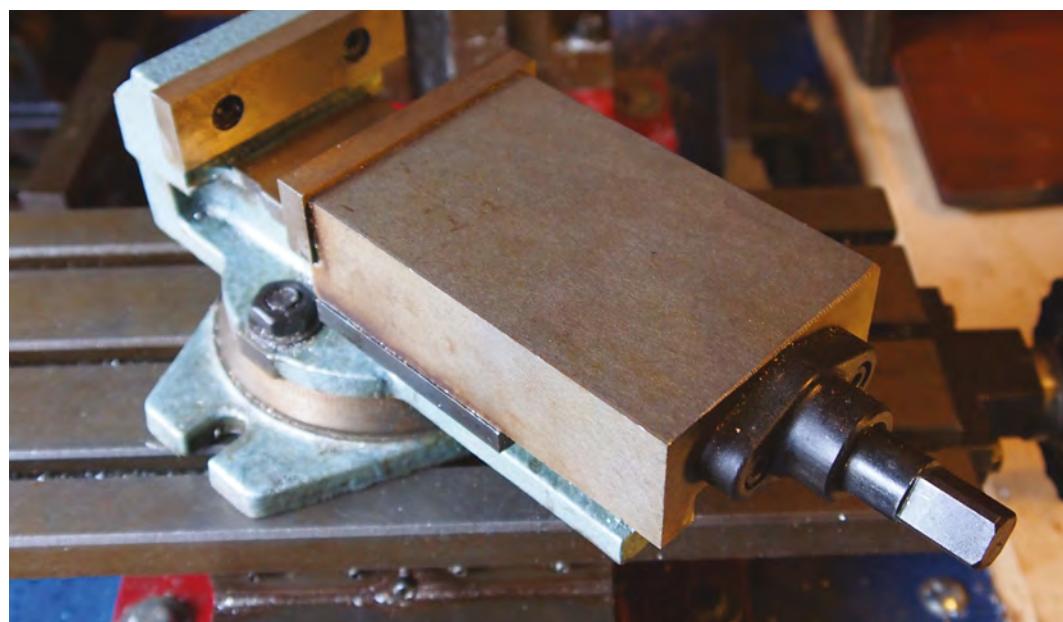
There are hundreds of types of tools that can be used on milling machines to perform a huge array of operations. However, there are a few common tools that we will mention.

TOOLS TOOLS TOOLS!

End mills (seen on the right in **Figure 4**) are a common bit that can be used to remove material from the face or from the side of an object using either the tip or the side of the tool, often with four or more 'flutes' (the sharp spiral cutting surfaces). Often, end mills cannot be plunged (driven vertically downward into the work) and therefore must approach workpieces horizontally. Slot cutters (on the left in **Figure 4**) are similar to

end mills but often have fewer flutes – two flutes is common, but other forms are available. Slot cutters are useful to cut slots (kinda obvious!), but also have the ability to be plunged into the work vertically and will drill their way into a workpiece. Fly cutters and face millers are a third common tool; they can have single or multiple cutting tips and are used to take shallow broad cuts across an entire surface of a workpiece to bring a piece to size with an even surface finish.

There are lots of accessories available for milling machines and it is easy to be bewildered with what you may or may not need; most home machinists slowly build up their collection of kit as and when they have a job that requires it. →



QUICK TIP

It's often said that when buying a milling machine or a lathe, you should budget the same amount again for accessories. The list of accessories for milling machines can feel never-ending!

Figure 5

An adjustable milling vice. The base of this vice can be clamped to the milling bed and the vice can be rotated to perform different angled operations

Machining Basics: Manual milling

SCHOOL OF MAKING

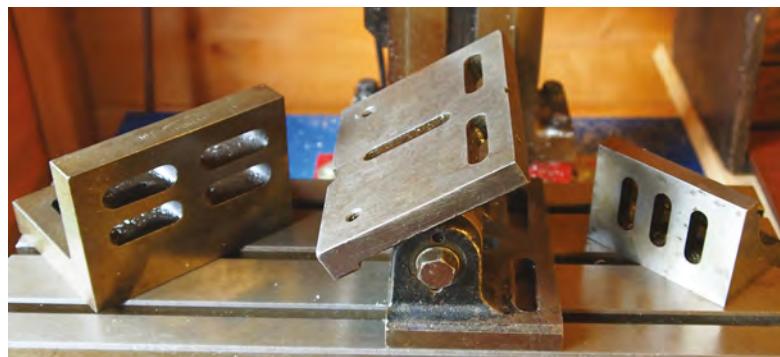


Figure 6 A small collection of angle plates: two fixed right-angle plates and one adjustable angle plate



Figure 7 A digital and analogue dial test indicator. These are super-useful items for a huge range of tasks!

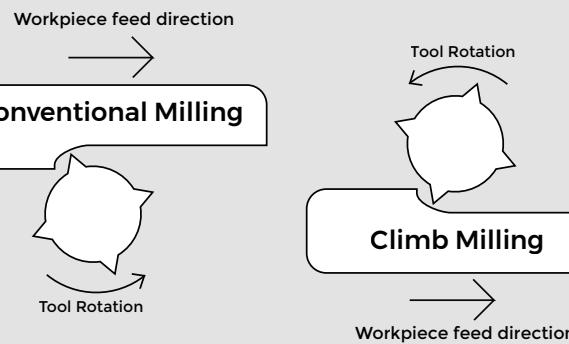
LET'S ACCESSORISE!

A milling machine doesn't come with everything you'll need because, well, there's no definitive list of what you need to mill. It all depends on exactly what you need to machine. Here are our favourite accessories.

A milling vice (Figure 5) is clamped to the milling table and can then in turn hold the workpiece. It can be useful to use a vice when you want to be able to perform operations on different faces of the same workpiece or you have a batch of workpieces you want to do similar operations on sequentially. Milling vices have a much larger, stiffer jaw to try to minimise any movement of the workpiece. They come in a range of types; some that can be adjusted/rotated and some that can be set to angles in two axes.

WHICH DIRECTION?

Essentially, milling involves moving the workpiece against the rotating cutting tool in the spindle. There are two basic ways that this can happen, referred to commonly as climb milling and conventional milling. The difference between them is that in conventional milling the cutting tool rotates against the direction of feed, and in climb milling the cutting tool rotates matching the direction of feed. There are numerous advantages and disadvantages of both. For example, conventional milling can minimise backlash (the play or slop in the machine where you may turn a handle slightly before it takes up the backlash and begins to move, which can affect accuracy) by pushing the workpiece against the cutter and closing the backlash. Climb milling has benefits in that the way the cutter removes material transfers more heat into the waste 'chip' rather than into the remaining workpiece, which could cause the workpiece to over-harden. Climb versus conventional milling is a broad subject and there is a lot of information (and heated discussion) to be found online.



Angle plates (Figure 6) can be used to set and clamp work vertically at a precise 90 degrees to the milling table. They are also used on a surface plate (a very accurately flat plate often made out of granite, steel, or cast iron) to help mark out work. Adjustable angle plates are very useful and can be adjusted to any angle and be rigidly clamped to the tables and have work or accessories clamped to them. Fixed and adjustable angle plates are usually precision-ground with very high accuracy.

Although many professional machinists would cry heresy to see wood being machined on a milling machine, using the milling machine and adjustable angle plates allows the author one of the best ways to add even chamfers onto rocket fin edges when rocket building.

Dial gauges (Figure 7) – mentioned previously, a dial test indicator can be used for many tasks and are a versatile addition to any work space. Often attached to an adjustable magnetic clamp base, they can be used to set work or set a vice square to the table, used as a precise positional readout, used to measure and compare the height of something (we call this using it as a comparator), or to measure the depth of a cut.

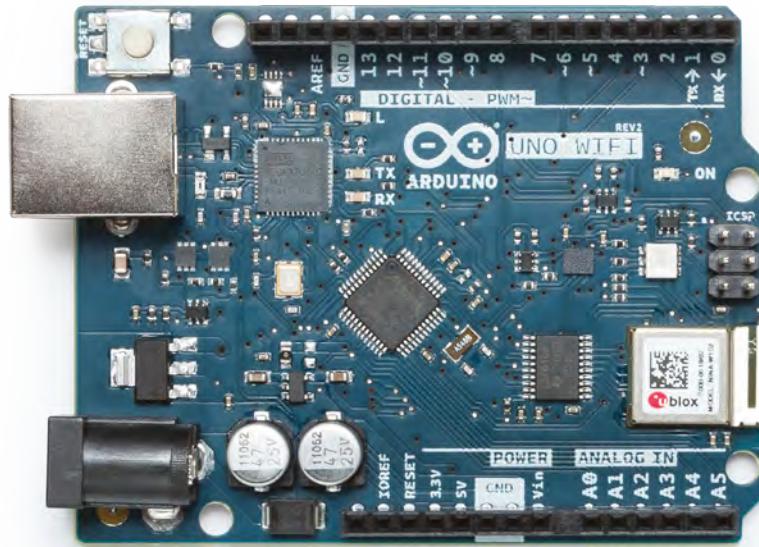
For example, in drilling operations they can be clamped to the underside of the milling head with the probe touching the surface of the work, zeroed, and then they can show the depth that the drill has travelled into the workpiece. They are useful across a range of machines (lathes, pillar drills, CNC routers, and more) and are available in analogue or digital forms.

Milling is simple in concept, but a hugely powerful way of creating parts. Hopefully you've now got an idea of how to get started. If you're interested in having a go, why not seek out an available mill in a community or hackspace? Once you have made a cut with a good finish, it is highly addictive! □

WIN!

— 1 of 10 —

Arduino Uno WiFi Rev 2



WORTH
£35

Go to
hsmag.cc/win



Enter by 15 February
for your chance to win!

Terms & Conditions

Competition opens on 19 December 2018 and closes on 15 Feb 2019. Prize is offered to participants worldwide aged 13 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families, or friends. Winners will be notified by email no more than 30 days after the competition closes. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. We don't like spam: participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered. Winners will be contacted by email to arrange delivery. Any winners who have not responded 60 days after the initial email is sent will have their prize revoked. This promotion is in no way sponsored, endorsed, or administered by, or associated with, Twitter or Facebook.

Maker's Toolbox: Rotary tools

Cut, sand, grind, engrave, and generally shape almost any material with one tool



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.

R

otary tools are often known by the trade name Dremel, and while there are now quite a few manufacturers available, Dremel is still a great company to turn to when you want one.

Rotary tools are, in many ways, like power drills in that they're a portable device for spinning a chuck that can hold various bits. However, they tend to spin smaller bits much faster than drills, and are used for a wide variety of tasks.

The versatility of rotary tools really comes down to the variety of tools available for them. Perhaps the most standard is the cutting disc. These are similar to cutting discs for angle grinders, only much smaller, and they come in a bewildering array of types. A few of the most common are:

Resin These are usually deep red and made of grit, bound together with resin. As you cut with these, they get smaller. Put any pressure on the side and they shatter, sending bits everywhere (although the bits are small and don't usually do much damage, as long as

you're wearing proper protection). You can get these discs very cheaply, but the bargain-basement ones tend to wear down very quickly and they can be time-consuming to swap out (you have to undo a screw in the end of the tool, then tighten it up over the new disc). They're mostly used for metal, but can cut plastic and glass. If you're very patient, they can get through some stone as well.

Fibreglass These are like a stronger version of the resin discs, and are hugely variable in quality. The Dremel SpeedClic ones are tough. Some cheap ones we've tried have been utterly useless.

Diamond Cut hard materials like stone and ceramic. These shouldn't wear down in the same way as the resin and fibreglass discs, but will dull over time.

Cross-cut saws You can cut wood with toothed blades. Some versions come with guards that protect you from flying wood and spinning teeth, while others just leave it up to you to keep your fingers safe (we strongly recommend the former).

Below A selection of rotary bits for drilling, grinding, engraving, cutting, sanding, and cleaning





After cutting, sanding is probably the most common thing you can do with a rotary tool. There are a few options including flap-discs, but spindle-style cylindrical sanding attachments are the most common. The small size of rotary tool sanders means they're best for getting into small places where larger sanders can't go. The counterpoint to this is that it's hard to sand a large surface without making it uneven. If you've got a powerful enough rotary tool, you can use a sanding attachment for shaping wood.

Similar to sanding, you can also get grinding bits in various shapes. These are great for removing burrs after cutting metal, or gently shaping metal. You can even use them for quick-and-dirty blade sharpening – the edge won't be smooth, but for rough gardening tools it can make quite a difference. Fine grinding bits, such as diamond burrs, can be used for engraving and can carve into metal or stone, but a steady hand is required for good results.

This is just a selection of the most common attachments. There's a huge range available to help you shape almost any substance in almost any way.

Rotary tools can be great for detail work – for example fine engraving or tidying up a joint. However, this sometimes needs to be done in awkward spaces. Although rotary tools are usually small compared to many power tools, they can be too big for some small spaces. You can get flexible shafts that allow you to work in particularly confined spaces.

PICK AND MIX

When choosing your multitool, the biggest things to consider are price, power, and portability. The power you need depends on what you're cutting, but if it's anything more than plastic, you'll want a powerful machine. This links in to portability, as battery-

powered rotary tools can be much less powerful than their mains-powered brethren, especially at the cheaper end of the range. Trying to cut metal on a underpowered machine is just a horrible experience.

We've been using a Dremel 3000 for some time now. At £38, it's towards the budget-friendly end of the spectrum, but it packs a punch. The 130W motor is powerful enough for getting through steel several millimetres thick, and makes short work of just about everything else. It's mains-powered, so does need to be used near a power outlet – but in a workshop, that doesn't cause a problem.

**Rotary tools can be great
for detail work – for
example fine engraving
or tidying up a joint**

Some rotary tools come with a vast range of attachments included, but bear in mind that the quality of these can be variable. While getting a wide range of bits for the minimum amount of money can seem attractive, the amount of time you can waste changing out tools can be significant. We'd recommend spending your money on the attachments you actually need, rather than trying to get as many as possible for the minimum spend.

The rotary tool is one of the most versatile tools in the workshop and we've used ours for everything from knife-making to resurrecting a vintage sun lounger. What do you use yours for? Let us know on Twitter: [@HackSpaceMag](#) or Instagram: [@hackspacemagazine](#). □

Above Our trusty Dremel 3000 is one of the most used tools in our workshop

SPIN SAFELY

As power tools go, rotary tools aren't too dangerous, but safety goggles are a must, and never go faster than the bits are designed for.



Changing states: Food powders

Hack your food to be the right physical state



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.

Above Peanut butter powder gives a delicious nuttiness to sweet or savoury dishes

Changing the physical state of something is an integral part of cooking. The liquid batter transforming into a cake, water turning to steam to cook veg, or butter melting to a liquid to fry food in, are all transformations that have been used in kitchens since time immemorial. Mastering these transformations can be the key to creating interesting dishes, and we're going to be reaching for some chemicals to help us convert liquids into a particular form of solid: powders. These powders give us a way of adding flavour to dishes with a little flourish. You can use them in a way similar to sauces but, as they don't add liquid, they won't make anything soggy.

Maltodextrin is a type of sugar; however, not all sugars have the same sweetness. General table sugar, or sucrose, is what we usually think of as adding sweetness, along with things like fruits

(which also contain other sugars like fructose and glucose). These are all sugars and fairly sweet. Other things are not sugars, but taste sweet – the most famous are the sweeteners, such as aspartame and sodium saccharin, that are far more sweet than sucrose. A tiny amount of these can add sweetness to a large dish. Similarly, there are sugars that aren't sweet, and maltodextrin is an example of this. The exact sweetness of maltodextrin does depend on the exact chemical make-up. It's made of chains of identical molecules – the longer the chains, the lower the sweetness. Maltodextrin can vary between slightly sweet and hardly sweet at all, but it's not easy to know what you're getting when you make a purchase as it's all classified as maltodextrin.

So, what use is a non-sweet sugar? Sugars absorb fats and oils, and this means we can use them to solidify things that would usually be runny. If you mix maltodextrin into something like olive oil or peanut butter, the result is a light, fluffy powder that still

has the flavour of the original. Because maltodextrin also dissolves in water, when you eat these powders they transform back to a liquid in your mouth, quickly filling your tongue with the delicious flavour.

The process for making this is really simple. You have to start with something that's high fat and low moisture. You then sprinkle on maltodextrin and mix until you've got something the right consistency. It's hard to say exactly what ratio of the two you'll

**Sugars absorb fats and oils
and this means we can use
them to solidify things that
would usually be runny**

need, as it depends on the exact make-up of your starting product. You'll probably find that you need somewhere between one and four times the volume of maltodextrin as original liquid. With something like peanut butter, that's already quite dry, you'll need towards the lower end. With something like olive oil, that's pure liquid, it'll be on the higher end.

Obviously these aren't going to be dishes in their own right, but they're garnishes that add an extra layer to a dish. For example, peanut butter powder sprinkled on top of a chocolate dessert is delicious, while olive oil powder can add an extra layer of savoury deliciousness to almost anything.

WATER-BASED

It's quite a limitation that we can only work with oil-based foods, so let's look at another way of achieving a similar result – gellan gum. There are quite a few additives that turn liquids to gels, through various different mechanisms.

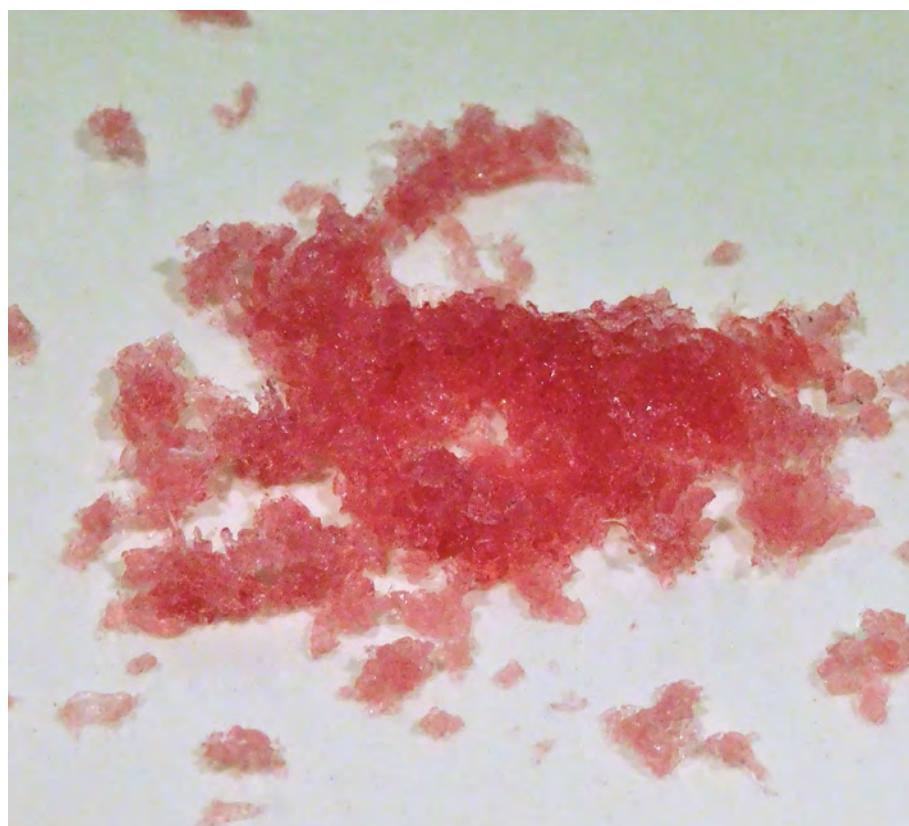
WHY POWDER?

When you're thinking of food condiments, powders are probably top of your list, even if you don't realise it. Salt or pepper are, after all, powders. This form makes it really easy to add to a dish and pump up the flavour, without altering the dish in any other way. The golden rule when it comes to condiments is to make sure it's packed with flavour. Whichever method you use to create your powder, you need to start with something full of taste, otherwise it risks getting lost in the dish, and as you'll only be adding a little, there's not much risk of overpowering the main flavour.

Gels are an odd physical state where a substance is mostly made up of liquid, but they have the properties of a solid because they contain solids dispersed throughout the liquid that impart form on the liquid. Gels can range from the soft to brittle. At one end of the scale, there's jelly (that's Jell-O to our transatlantic friends) that's soft and wobbly. At the other end, there are more rigid gels.

Gellan gum comes in two forms – low acyl and high acyl. Both will form gels, but the low acyl kind will form the sort of rigid gel that we're after. It's fairly easy to use – just dissolve about 5g of gellan gum per 100ml of liquid while the liquid is cool, then heat it up to boiling point until it's fully hydrated. At this point, pour it into a container for cooling. After a couple of hours, it should be fairly solid and you can use a fine cheese grater to convert this into a powder. This powder will retain its structure even at fairly high temperatures, so you can do this on hot food. What you use this for is up to you, but sprinkling some powdered fruit juice over a dessert adds a delicious tang without making anything soggy, while powdered vegetable juice can have a similar effect on savoury dishes. □

Below ♦
Blackcurrant powder
adds fruitiness to
desserts without the
mess of a sauce



Control a K40 laser cutter over a network

The easy way to send jobs to your cutter



Steve Pelland

@StevePelland

Steve works in IT by day and can be found in his garage shop at night. When not in his shop he can be found at the local hockey rink watching his son play. He enjoys working with the Scouts teaching STEM badges.



K40 laser cutters are a staple of workshops around the world, as they're one of the most affordable routes into computer-controlled manufacturing. However, by default, they're very bare-bones, and it takes a bit of work to get a setup you're comfortable with. One problem with the off-the-self setup is that you have to leave your computer plugged in via USB to run your job. Let's take a look at one way of removing this dependency, allowing you to run jobs without your main machine plugged in. This is particularly useful if you've got a desk in your workshop away from where your laser cutter is set up.

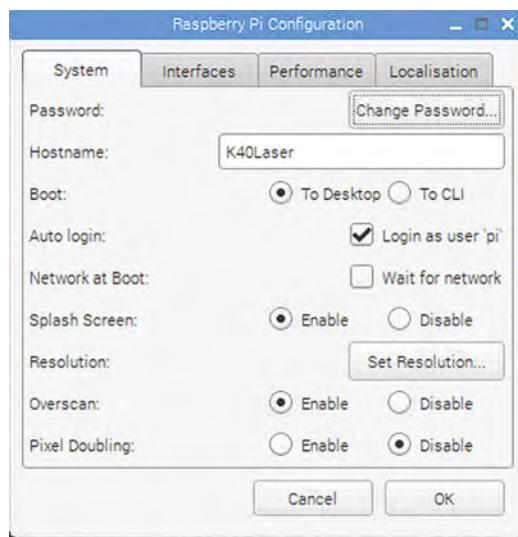
Setting your laser cutter up like this will let you control it from anywhere. However, we can't recommend this. Laser cutters essentially run via

controlled burning, and the line between controlled and uncontrolled burning is slim. Even when running a job you've done before, it doesn't take much to set fire to your machine. This setup means you can keep your computer in a more convenient part of your workshop, so you can keep an eye on your machine. And, of course, don't forget to keep a fire extinguisher on hand in case something does go horribly wrong.

We tested this using a Raspberry Pi 3B+ with Raspbian Stretch June 2018 installed. In theory, any RPi should work, but we'd recommend either a 3B+ or a 3A+ as you'll get better performance. You'll need this setup before starting the tutorial.

It's easiest to set up the RPi with a mouse, keyboard, and monitor attached, but we'll remove these once everything is running.

Above K40 lasers can be an affordable way to start cutting



Once your machine has booted, run the Raspberry Pi Configuration application from the main menu – click on the raspberry icon in the upper left-hand corner of the screen – then click on Preferences and then Raspberry Pi Configuration.

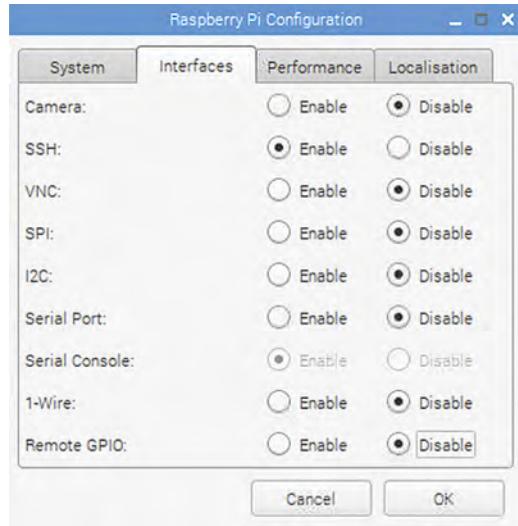
Under the System tab, you can set the host name, change password, etc.

The important part is in the Interfaces tab, where you can enable the camera, SSH, and VNC. For this tutorial we only need to enable SSH.

It makes it easier to keep track of your Raspberry Pi if it has the same IP address every time it boots. Normally these are assigned by your router every time you turn your machine on. On some networks, you can set the static IP directly on your RPi. To do

Below

You can turn on or off a variety of low-level protocols on your Raspberry Pi



this, right-click the WiFi icon in the upper right-hand corner and configure your chosen interface. However, on some networks you'll have to make this change in your router's configuration – this is a little different for each router, so check your manual if you're not sure.

Now we can shut down the RPi, disconnect the keyboard, mouse, and monitor, and move it next to the laser cutter.

Upon booting, use your SSH client of choice. If you don't have an SSH client of choice, PuTTY or MobaXterm works fine for Windows (we like MobaXterm because of the tabbed connections).

Our setup in MobaXterm is as follows:

- The remote host is the static IP address that you set up previously
- The default username and password are Username: pi and Password: raspberry
- Enable X11 Forwarding
- Click OK

Double-click the session on the left-hand side if you are using MobaXterm or connect with whatever client you are using. Enter your password when prompted, and you should be dropped into a command prompt.

You'll also need some software to display the images on your local machine. This is known as an X server. If you're using Windows, Xming (sourceforge.net/projects/xming) works well, and if you're on a Mac, XQuartz (xquartz.org) is a good option. →



Left

We named our RPi K40Laser so it's easy to identify on the network

Below

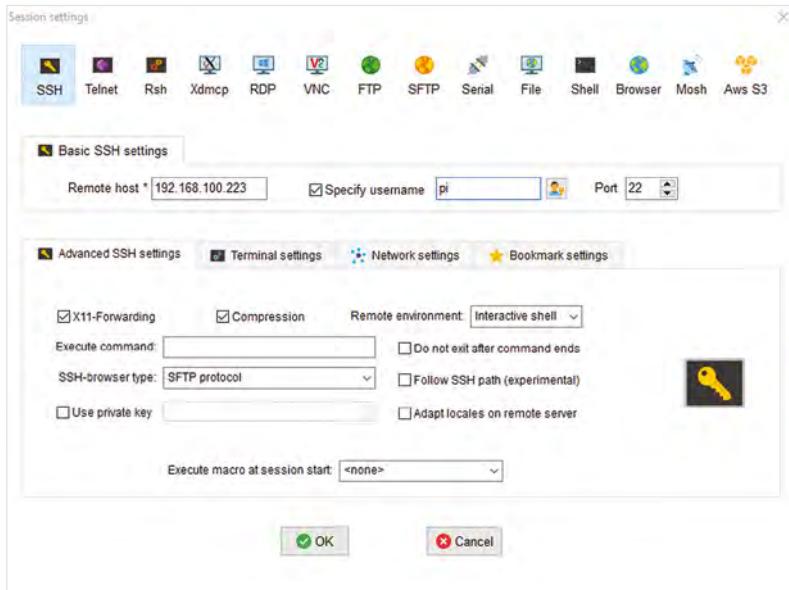
Setting a static IP helps you keep track of your machine across reboots

YOU'LL NEED

- ◆ Raspberry Pi
- ◆ K40 laser cutter
- ◆ A desktop computer
- ◆ A display you can temporarily plug your RPi into for the OS installation and configuration
- ◆ USB cable (should have come with your K40)

Control a K40 laser cutter over a network

TUTORIAL



Above
The default SSH settings to connect to your Raspberry Pi

That's the basics set up. We can now install the software we need to control the RPi. SSH into your machines and run the following:

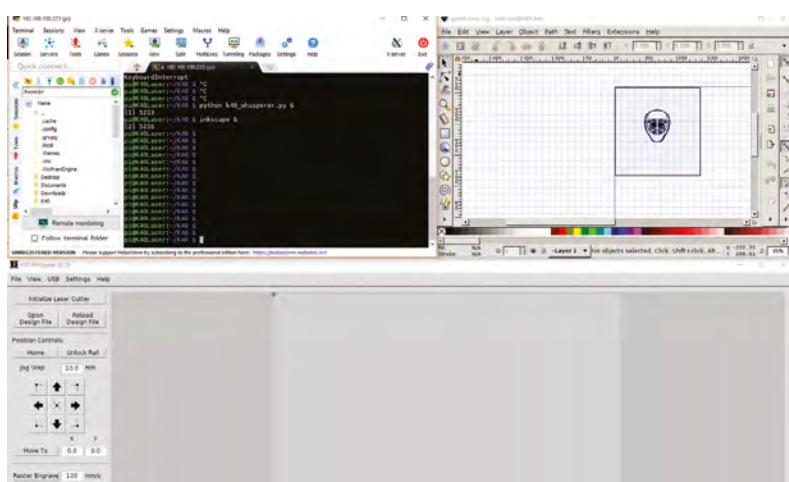
```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install -y libxml2-dev libxslt-dev
sudo apt-get install nano
sudo apt-get -y install inkscape
```

You'll now need to turn on your K40.

We will need to find out the USB device ID of the K40: we can use the `lsusb` command. When executed, we received the following output (but yours will be slightly different):

```
Bus 001 Device 011: ID 1a86:5512 QinHeng
Electronics CH341 in EPP/MEM/I2C mode, EPP/I2C
adapter
```

Below
K40 Whisperer and Inkscape running on our desktop



Copy this to a text editor for future use. In this example '1a86' is the VENDOR ID and '5512' the PRODUCT ID.

Create a udev control file for your laser cutter. You'll need to do this using a text editor running as root, which you can launch with:

```
sudo nano /etc/udev/rules.d/97-ctc-lasercutter.rules
```

Put the following text into the file and replace [VENDOR ID] and [PRODUCT ID] with the information you obtained from lsusb:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="[VENDOR ID]", ATTRS{idProduct}=="[PRODUCT ID]", ENV{DEVTYPE}=="usb_device", MODE=="0664", GROUP="lasercutter"
```

Press **CTRL+X** to save and exit. You'll need to reboot your RPi to make these changes take effect.

Once it's back up and running, you'll need to SSH in again and download the K40 Whisperer source code, for example **K40_Whisperer-0.XX_src.zip**.

Use the `wget` command to download the file:

```
wget http://www.scorchworks.com/K40whisperer/
K40_Whisperer-0.17_src.zip
```

Version 0.17 was current at the time of writing, but any later version should work.

Once downloaded, let's unzip to a directory.

```
mkdir /home/pi/K40
unzip K40_Whisperer-0.17_src.zip -d /home/pi/K40
```

Go to the **K40 Whisperer** source directory, and install the required packages:

```
cd /home/pi/K40/
pip install -r requirements.txt
```

Everything's now installed and ready to run, but we just need to do a bit of jiggery-pokery to get the windows to appear in the right place (on our main machine, not on the monitor that's no-longer attached to the RPi). This means using `xauth` to point to the right place. We've written a script to do it all for you. Save this in a file called **sudox** in your home directory:

```
su - pi -c 'xauth list' |\
grep `echo $DISPLAY` |
```

```
cut -d ':' -f 2 |\
cut -d '.' -f 1 |\
sed -e s/^/:` |\
xargs -n 3 xauth add
```

Now make this file executable with:

```
chmod a+x /home/pi/sudox
```

One more reboot should make sure everything is set up and ready to run.

Once rebooted and logged in via SSH, you can change directories into the **K40** directory and launch K40 Whisperer.

```
./sudox
cd K40/
python k40_whisperer.py &
inkscape &
```

(The & symbol tells Linux to run the program in the background, allowing you to issue more commands with another prompt.)

Now that we are up and running, we need one more thing. How do we send our jobs to the RPi? You can just open a browser from the RPi, download items, work in Inkscape, and then run in K40 Whisperer. What if you designed something on your workstation and just need to send it to K40 Whisperer? Easy. Let's just set up a share on the RPi, map the drive in Windows, and then you can drag and drop or even just save files directly to the share. Then you have the freedom of choice – work from the RPi or work in conjunction with your workstation.

So, let's get Samba installed. Samba will allow us to share our work directory easier with Windows clients.

```
sudo apt-get -y install samba samba-common-bin
```

Let's create the directory:

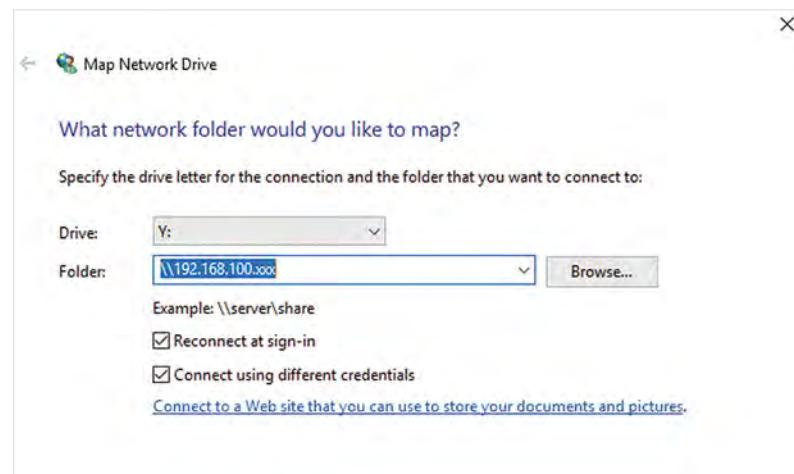
```
mkdir ~/K40Projects
```

Now we need to edit the Samba configuration file:

```
sudo nano /etc/samba/smb.conf
```

And add the following to the bottom of the file:

```
[PiShare]
comment=Raspberry Pi Share
path=/home/pi/K40Projects
browseable=Yes
```



```
writeable=Yes
only guest=no
create mask=0777
directory mask=0777
public=no
```

Above Link your two computers together with file sharing

Now you can set the password for file sharing. Enter the following in a terminal:

```
sudo smbpasswd -a [username]
```

ENTER THE PASSWORD TWICE

Now you can connect to this from your Windows machine. Go to Windows Explorer, right-click on My Computer and select Map Network Drive.

For the folder, use **\[ip address of pi]** and then click on Finish.

You will then see a password prompt. Enter your password. And your drive should be mapped.

That's everything set up and ready to you. You can now use your laser cutter from other machines – the only thing you have to decide is what to cut next. ☐

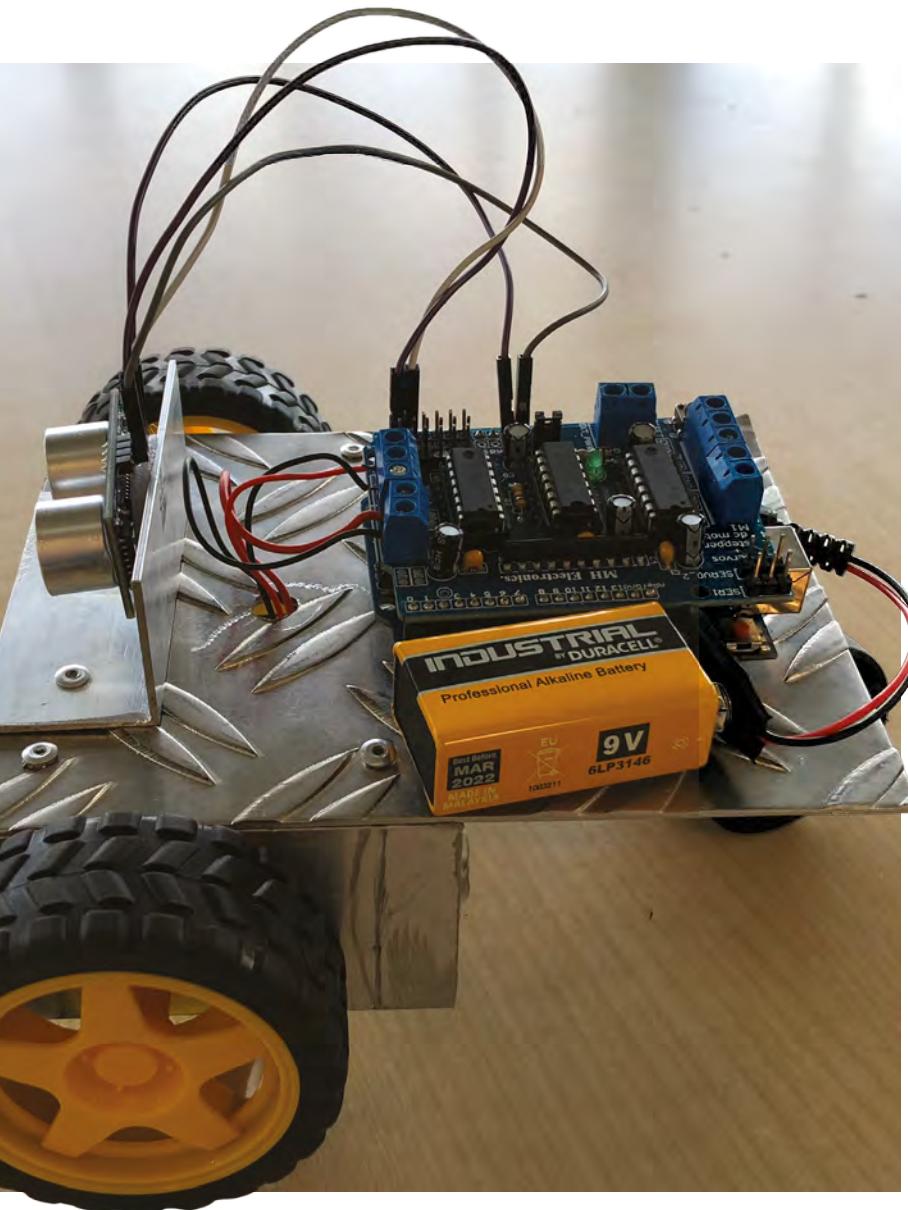
MORE INFORMATION

We learned lots from the online community of Raspberry Pi and laser cutters. You can find out more about the process we went through here at the following resources:

- Setting up K40 Whisperer and Inkscape on the RPi
raspi.tv/2018/run-a-k40-laser
- Forwarding applications utilising sudo through an SSH tunnel
joelinoff.com/blog/?p=729
- Creating shares on Raspberry Pi
raspberrypihq.com/how-to-share

Build an Arduino robot from scratch part 2

Adding brains to help your robot drive



In part 1 we built the body for our robot rover.

This section will give life to your creation. You will move your robot with code, program it to return, and then provide your vehicle with the ability to see and touch and react to these senses.

In part 1 we gave life to our motors; setting the speed and moving backwards and forwards. The objective now is to move a set distance and return to the start point. Working with your original code (make a copy), measure 2m (or 6ft if you prefer). Adapt your code so that it moves your rover forward for the measured distance. Most of your original can stay. Edit the loop so that it looks something like the following:

```
void loop() {
    motor3.run(FORWARD);
    motor4.run(FORWARD);

    delay(4000);
    motor3.run(RELEASE);
    motor4.run(RELEASE);

    exit(0); }
```

The **delay** figure will depend on the speed of your motors and surface. Experiment with this to get as close as possible to two metres. The **RELEASE** command sends a stop signal to the motor. The **exit(0);** command stops the processor. Remember that, depending on how you've wired your motors, **FORWARD** might not be forward, so this could read differently.

BACK TO THE START

Once you are happy with the two-metre test, you need to consider how your robot will return home. The easiest method is a simple reverse. Adding the **BACKWARD** command for both motors before the **RELEASE** code should do it. But is a simple reverse enough?



Michael Jones

↗ hsmag.cc/jfjyXg

CS teacher in Kent, long-term CAS member and maker. Determined to turn abstract into practical teaching videos.

Far more elegant would be to consider your code and get your rover to turn around – this will be useful later when we equip our robot with senses.

The best way to turn your robot is to stop one motor and set the other in reverse. Remember to apply a **delay** command to this – two seconds is a good starting point, but you'll have to experiment to see what works for you.

```
motor3.run(FORWARD);
motor4.run(RELEASE);
delay(2000);
```

You should be able to sandwich this inside two copies of your straight-line code to move in a straight line and back to the start.

Now you are comfortable with getting from A to B and back, edit your code so that your rover can move in a rectangle. This will require some degree of tweaking. The essential code that we use is modified from the reverse code: repeat this four times and don't forget to add a **release** and **exit** at the end of the code:

```
motor3.run(FORWARD);
motor4.run(FORWARD);
delay(2000);
motor3.run(FORWARD);
motor4.run(RELEASE);
delay(2000);
```

At this stage, your code starts to look a little unwieldy. In the rectangle sketch, development code is repeated four times. For those of you familiar with programming techniques, you can place the 'forward, turn' code into a loop. Adapting this code with more steps and shorter delays will provide a routine for your robot moving in a circle.

GETTING SENSITIVE

The particular motor shield used in this project requires soldering. The shield has a number of slots

available to solder header pins for power, analogue, and ground. In the bottom-left corner of the shield, you will see three rows of holes labelled AO-5, Gnd, and +5. Fire up the soldering iron, break off three sets of six pins, and solder them in.

THE SENSE OF SIGHT!

The ultrasonic sensor has four pins: VCC, Trig, Echo, and GND. The sensor works by firing a pulse from the trigger and measuring the time that the pulse takes to bounce back (Echo).

Attach four female-female jumper wires to the sensor. On your newly soldered motor shield, attach the GND to a GND pin and VCC to +5 on the shield. Checking the wiring for the remaining two pins, Trig is connected to Analogue A5 and Echo to A4.

You can download the code for this from hsmag.cc/issue14. It's in the folder **Sight1**. If you upload this to your Arduino, you will start to read the distance. In the code, lines 22–24 report the distance. Go to Tools > Serial Monitor. If your sensor is functioning, the monitor will report the distance. Move the sensor a little to check distances are being reported.

When you're happy that the sensor is working, attach it to the front of the robot with Velcro. →

DOUBLE VISION

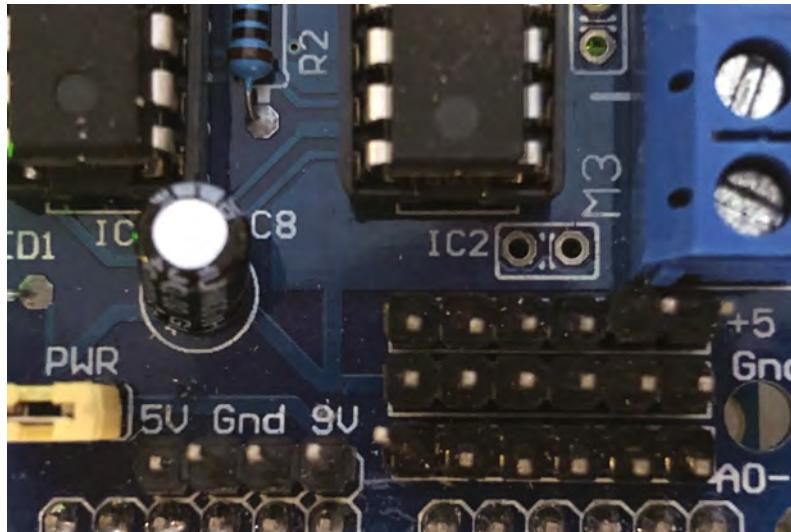
Want eyes in the back of your head? Attach a second sensor to the back of the robot. Make sure you modify the code to reflect additional Trig and Echo.

Above ↗
The core sensory item for our robot is the HC-SR04

Below ↘
View of Arduino analogue pins



TUTORIAL



Above View of motor shield pins

Right Front and back view of ultrasonic sensor

AUTOMATIC MOVEMENT

Now you have provided 'sight' to your robot, it is time to link the motor and sensor code. The image contains the necessary code to move your robot. A sensible pulse rate to check distance is one tenth of a second. This can be modified, depending on the speed of your motors. Try slowing your speed and increasing the delay between reading.

A good starting point is to set the trigger distance to 20cm, (8 inches). At the moment, the code turns one way only when an object is seen. This can be modified to turn the other way according to preference.

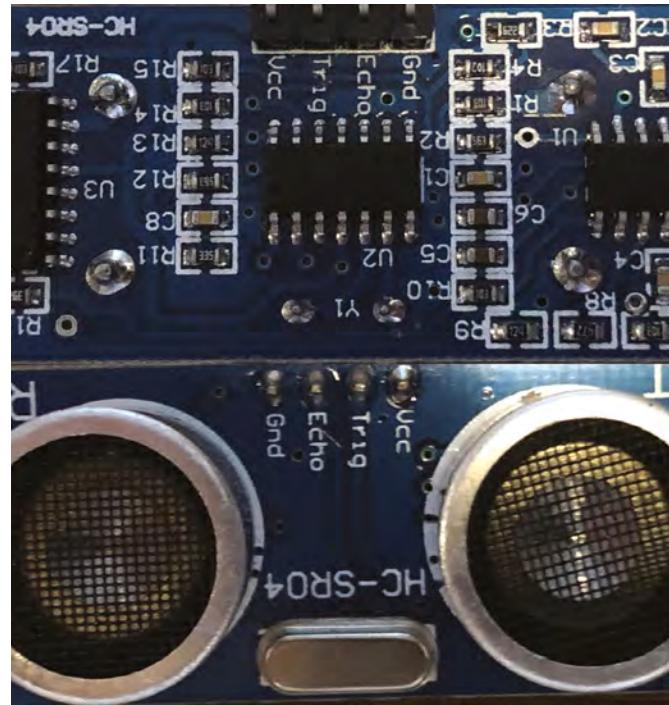
The following code (also available from hsmag.cc/issue14) reports back on the distances. Use this to test your sensor.

```
#include <AFMotor.h> //import your motor shield library
const int trigPin = A5;
const int echoPin = A4;

AF_DCMotor motor3(3, MOTOR34_1KHZ); // set up motors.
AF_DCMotor motor4(4, MOTOR34_8KHZ);

void setup() {
    Serial.begin(9600);
    pinMode (trigPin, OUTPUT);
    pinMode (echoPin, INPUT);
    motor3.setSpeed(150); //set the speed of the motors, between 0-255
    motor4.setSpeed(150);
}

void loop() {
    unsigned long duration;
    float distance;
```



```
digitalWrite(trigPin, HIGH);
delay(100);
digitalWrite(trigPin, LOW);
duration = pulseIn (echoPin, HIGH);
distance = (duration / 2.0) / 29.0;

Serial.print("Data:");
Serial.print("d = ");
Serial.print(distance);
Serial.println(" cm");
delay(100);

if (distance < 20){
    Serial.println ("Close Obstacle detected!");
    Serial.println ("Obstacle Details:");
    Serial.print ("Distance From Robot is ");
    Serial.print ( distance);
    Serial.print (" CM!"); // print out the distance in centimeters.

    Serial.println (" The obstacle is declared a threat due to close distance. ");
    Serial.println (" Turning !");
    delay(250);
    motor3.run(FORWARD); // Turn as long as there's an obstacle ahead.
    motor4.run (BACKWARD);
    delay(250);

}
else {
    Serial.println ("No obstacle detected. going
```

YOU'LL NEED

- ◆ Built robot from part 1
- ◆ Ultrasonic distance sensor HC-SR04 (one minimum, two if you mod up)
- ◆ 2 x Limit switch
- ◆ Velcro dots
- ◆ Header pins
- ◆ Jumper cables
- ◆ Hot glue gun
- ◆ Cable ties
- ◆ Heat shrink tubing
- ◆ Soldering iron and solder
- ◆ Solder pump (optional but useful for solder mistakes)

```

forward");
delay (250);
motor3.run(FORWARD); //if there's no obstacle
ahead, Go Forward!
motor4.run(FORWARD);
}
}

```

SENSE OF TOUCH

Now that you have a seeing robot, it's time to add touch to your rover. For this part you will use two limit switches. These have three pins, typically labelled C/Com, NO, and NC. The C pin is on one end of the switch, NO is the Normally Open connection, and NC is Normally Closed. When the switch isn't pressed, C and NC will be connected; when they're closed, C and NO will be. You could use either NC or NO, but we'll use NO. Connect C to ground and NO to your input.

Using two lengths of wire with female ends on one side, solder the wire to each of the limit switch pins, add a length of heat shrink (check it is snug and covering the metal), and apply heat. Attach a cable tie to the metal arm of the switch to act as a whisker (we used hot glue). Fix the switch mechanism to the chassis of the robot with hot glue. Make sure this is placed the correct way on the body. Repeat this for your second switch.

Attach the ground wires to the pins on the motor shield. Use analogue pins A0 and A1 for the signal (one for a limit switch on each side). Carefully use the code in the twoSens downloadable code.

Create the code, compile, upload, and off you go. The switches are coded to turn left or right based on the sensing of objects to the side.

```

#include <AFMotor.h>

const int trigPin = A5;
const int echoPin = A4;
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);
int lbumper = A0;
int rbumper = A2;
int lstate;
int rstate;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  motor3.setSpeed(150);
  motor4.setSpeed (150);
  pinMode(lbumper, INPUT_PULLUP);
}

```

GREATER ACCURACY

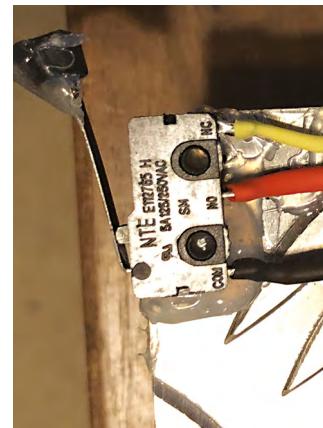
In this tutorial, we've used simple timings to work out how far a motor has gone. This works up to a point, but can cause problems – for example, if one motor encounters some resistance from something on the ground.

If you want to move a robot more accurately, you need a more complex control setup. One option is to use stepper motors. These can be used to turn a specific amount each time, and they can be used to make very accurate movements. However, they can be slow and expensive. What's more, they don't always work, as they can skip steps under load. The most accurate option is to use a motor with a feedback mechanism that shows exactly how far it's travelled. An optical encoder can sense the rotations that a motor makes, and this can be used to adjust the motor power to ensure that the motors move exactly the required amount.

```

pinMode(rbumper, INPUT_PULLUP);
}
void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delay(200);
  digitalWrite(trigPin, HIGH);
  delay(200);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  lstate = digitalRead(lbumper);
  rstate = digitalRead(rbumper);
  motor3.run(FORWARD);
  motor4.run (FORWARD);
  if (distance < 20 )
  {
    motor3.run(FORWARD);
    motor4.run (BACKWARD);
    delay(500);
  }
  if(digitalRead(lbumper) == LOW )
  {
    motor3.run(FORWARD);
    motor4.run (BACKWARD);
    delay(500);
  }
  if(digitalRead(rbumper) == LOW )
  {
    motor3.run(BACKWARD);
    motor4.run(FORWARD);
    delay(500);
  }
}

```



Above Adding fingers to your robot

Your robot is now able to sense its way around the world. There's not much to it, but you can build from here into almost any wheeled contraption you can imagine. □



£12.99

FREE WORLDWIDE SHIPPING

BUY TODAY AT
STORE.RPIPRESS.CC

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



Audio-reactive NeoPixel mask

Light up the party with the sound of your voice



Above ↗
NeoPixels + Shiny
+ Audio = a great
looking party outfit

B

ring your project to life with sensors! Sensors let your microcontroller react to the environment around it, by sensing sound, light, vibration, and more. If you have already used microcontrollers

to control outputs like LEDs, you're ready to start using inputs to drive those outputs. In this project, we will use a microphone to sense sound, and turn that data into light, with a Gemma M0 and NeoPixels from Adafruit.

The connections on microcontrollers are commonly called GPIO pins. 'GPIO' stands for 'general-purpose input/output', meaning the pins can be used for either an output like an LED, or an input like a microphone. In our CircuitPython code, we'll set up one of our pins as an output for the NeoPixels, and another as an input for the microphone. A little bit of maths will turn sounds picked up by the microphone into colour shown by the NeoPixels.

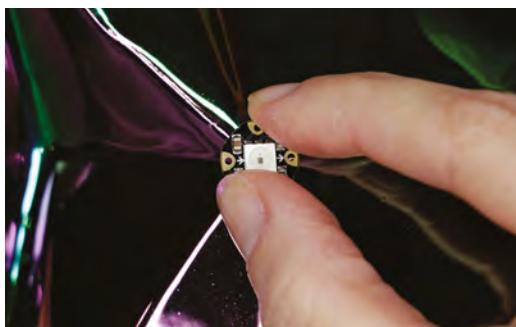
Sensors can either be digital or analogue. Digital sensors only detect two states: on or off. Analogue sensors can detect a range of states; for example, volume from 0% to 100%. Notice that the GPIO pins on the Gemma M0 each have two labels: D0, D1, D2, and A0, A1, A2. This is because each pin can be used as either a digital or analogue pin. NeoPixels are digital devices, so we'll use that pin as a digital output (D0), while our microphone will be an analogue input (A1).

NeoPixels are a type of LED that can display any colour. They come in lots of different shapes and sizes, and we'll be using both sewable NeoPixels and a NeoPixel Jewel on this project. A Jewel is just a pre-wired collection of NeoPixels.

The mask used here was purchased by your author for about \$5. Look for a mask made of plastic soft enough to drill holes into, but rigid enough to hold the electronic components securely. It's also helpful if

Below

Sewable NeoPixels are a great form factor for wearables



the shape of the mask creates some recessed areas that can house the electronic components without pressing them against your face. The cat mask in our project has large, exaggerated features that worked well. Find a mask that inspires you, and let's get started!

DESIGN YOUR MASK

The first step is to decide where the components will go on your mask. The NeoPixels will be on the outside, and all the other components will be on the inside. NeoPixels can be very bright, so think about the shape of light you will create when the LEDs are lit. Our mask is very angular, so we placed the NeoPixels symmetrically in a V shape, accentuating the angles of our cat ears. If you have a more organically shaped mask, you may want to try a swirl or a more random pattern. It can be helpful to use small bits of adhesive putty to hold things in place while you design. Holes in plastic are permanent, so make sure you like the placement and fit of everything before cutting into your mask.

Think about your circuit, and remember that all the NeoPixels will be connected together in a daisy-chain, with the Gemma M0 connected to the first pixel. The arrows on each NeoPixel indicate the direction that data flows through the chain, and must all point in the same direction: away from the Gemma M0. The NeoPixel Jewel can be inserted at any point in the chain. Try to find flat areas on the front of your mask that the NeoPixels can be adhered to securely, and keep in mind that you'll need to drill holes behind the four pads of each NeoPixel.

On the inside of the mask, look for recessed areas that do not press against your face while the mask is being worn – these are good locations for components. Find a spot near the first pixel on the inside of the mask for the Gemma M0 and the →

Above Blu-Tack is a great temporary adhesive to test out how things can look

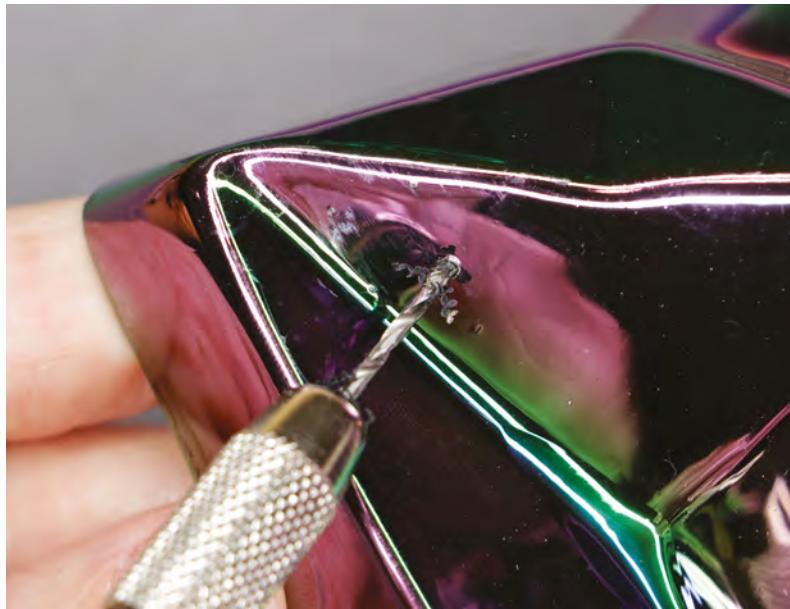


Sophy Wong

@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

TUTORIAL



YOU'LL NEED

- ❖ **Plastic mask**
- ❖ **NeoPixel Jewel**
(product #2226)
- ❖ **4 × Flora RGB Smart NeoPixels**
(product #1260)
- ❖ **Electret microphone breakout** (I used #1713, but #1063 will also work)
- ❖ **Gemma M0**
(#3501)
- ❖ **Coin cell battery holder, with JST connector**
(holds two batteries)
- ❖ **Silicone-coated 30 AWG wire**
- ❖ **Soldering supplies**
- ❖ **Hot glue gun and glue sticks**
- ❖ **Awl**
- ❖ **Craft knife**
- ❖ **Small hand drill for crafts**
- ❖ **Removable adhesive putty**
- ❖ **Electrical tape or Kapton tape**
- ❖ **Masking tape**

Above

A hand drill gives more control than a power drill

battery holder. You can mount them separately, as we did, or if space is tight, you can glue the Gemma on top of the battery holder. Just make sure you can still change the batteries and access the on/off switches. You'll also need to find a spot for the microphone. If you want the NeoPixels to light up when you speak, place the microphone somewhere close to your mouth. You'll mount it face-down on the inside of the mask, and make a hole for the microphone to fit through.

MAKE HOLES

When you've sorted out where you want everything to go, you're ready to make some holes. Use an awl to mark the holes you need to drill – one for each of the four pads on the NeoPixels. On the last NeoPixel in the chain, you do not need to drill a hole for the data-out pad.

Use a small hand drill to make the holes. Check to make sure that two wires will fit through the holes for power and ground.

To make the hole for the microphone, trace a circle around the microphone onto a piece of masking tape, and cut it out to make a template. Use this template to draw a circle for your microphone hole, then carefully cut the hole out with a craft knife. You may want to start the cut by drilling a smaller hole somewhere inside the circle, and then slowly widening the hole with the craft knife. Check the fit often, and remove just enough material so that the microphone slides in easily but is a snug fit.

CONNECT THE NEOPIXELS

The wiring for this project is fairly straightforward. The NeoPixels all have four connections: data-in, data-out, power, and ground. Power and ground are connected to the positive and negative terminals on the battery. The NeoPixels themselves are chained so the data-in of the first pixel is connected to D0 on the Gemma, and data-out is connected to data-in on the second, data-out on the second is connected to data-in on the third and so on. The Jewel also forms a link on this chain.

Cut lengths of wire for the first NeoPixel, making them at least 5 cm longer than the distance they need to cover. Remember that both the power and ground pads will need two wires each. Feed wires through the holes from the inside to the front of the mask. To easily thread two wires through the small holes, strip some insulation off the ends of the wires and twist them together. Thread the wires through the mask, then through the NeoPixel from back to front. Check the orientation of the NeoPixel, making sure the arrows are pointing in the direction data will flow (away from the Gemma M0). Solder the wires to the pads on the front of the NeoPixel, and trim with angle cutters.

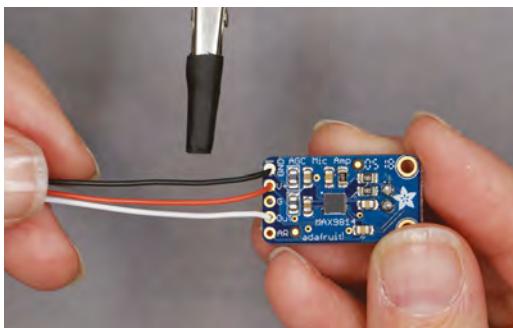
On the inside of your mask, pull the wires gently so the NeoPixel is snug against the front of the mask. (Don't glue anything in place just yet: we'll test the circuit before we make anything permanent.) Spread the wires out along the direction of your circuit. You should have three wires pointing back towards the beginning of the circuit: data-in, one power, and one ground. You should also have three wires pointing forward to the next pixel in your circuit: data-out, and the two other power and ground wires.

Repeat the soldering process for the next NeoPixel in the chain, adding another wire for power, another wire for ground, and a data-out wire. Don't

Below

Heat shrink tube over crocodile clips helps protect components





worry too much about keeping the wires very short: it's better to give yourself some extra length to make it easier while soldering. The Jewel is added in the same way.

Remember that the last NeoPixel in the chain will not have a data-out wire, and only needs one power wire and one ground wire. When your last NeoPixel is soldered, it's a great time to grab your multimeter and check the circuit for continuity. Power and ground should connect from the first NeoPixel to the last, and the data-out of each NeoPixel should connect to the data-in of the next.

Although CircuitPython code can be written in any text editor, we recommend using the Mu editor

Find the free wires at the beginning of the NeoPixel chain for power, ground, and data-in. Solder these to the Gemma M0: power to Vout, ground to GND, and data to D0.

CONNECT THE MICROPHONE

Cut three wires for the microphone breakout, making sure that they will easily reach from the microphone to the Gemma M0 on the mask. Feed the wires through the holes on the board from back to front, one each to GND (to the power supply negative), Vdd (to the power supply positive), and Out (to the Gemma A1). Solder the wires on the front of the board, and trim any bare protruding wires.

Temporarily hold the microphone in its place on the inside of the mask with adhesive putty, and solder the wires in place.

PREPARE TO PROGRAM

Now that everything is connected, it's time to program the Gemma M0 with CircuitPython! Connect the Gemma to your computer with a micro USB cable, and turn the Gemma on. The Gemma will appear as a drive called CIRCUITPY.

If you're new to CircuitPython, head over to hsmag.cc/ZKvDqa for an overview and resources for learning more, and troubleshooting. Even if you've got a brand new Gemma M0, follow the installation instructions there to make sure your Gemma M0 is updated with the latest version of CircuitPython.

For this code, you'll also need the latest version of the neopixel.mpy library. Visit hsmag.cc/KEvJSv to grab the latest version of all the Adafruit CircuitPython libraries, then copy the **neopixel.mpy** file into the **lib** folder on the Gemma M0. If there is already a **neopixel.mpy** file in the **lib** folder, replace it with the new one. If you don't already have a **lib** folder on your Gemma M0, create one.

Although CircuitPython code can be written in any text editor, we recommend using the Mu editor, which includes handy helpers like code checking and a serial monitor for debugging. Adafruit has a great tutorial on using Mu here: hsmag.cc/NtvMQt. You may already have a text file on your Gemma M0 called **code.py**, but if not, use your editor to create one for your new code. If you already have a **code.py** file, delete its contents so we can write our program from scratch.

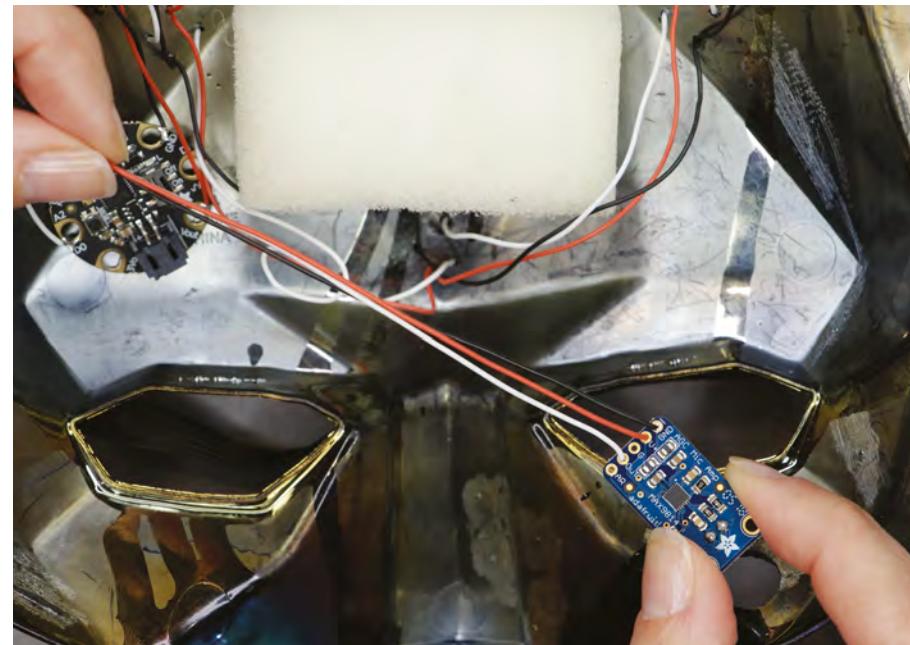
As always, our code begins with importing libraries. In addition to the usual Time, Board, and NeoPixel libraries, we'll also be using Random and AnalogIO.

```
import time
import board
```

Left ♦
A microphone ready to be connected

Below ♦
The amplifier board sits inside with just the microphone protruding

Bottom ♦
The inside of your mask doesn't have to be pretty, but it does have to be comfortable



Audio-reactive NeoPixel mask

TUTORIAL

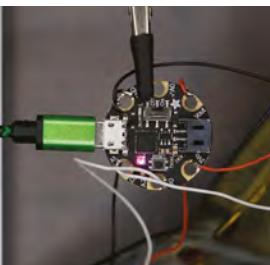
```
import neopixel
from analogio import AnalogIn
import random
```

Next, we'll set up our NeoPixels as a strip on the D0 pin. The variable `numpix` is the number of pixels connected together in our mask: the four individual NeoPixels plus seven on the NeoPixel Jewel makes eleven. We'll also set up the microphone as an analogue input on pin A1.

```
numpix = 11
pixpin = board.D0
pixels = neopixel.NeoPixel(pixpin, numpix,
brightness=0.3, auto_write=False)
microphone = AnalogIn(board.A1)
```

We'll create a few more variables while we're at it. `blinkColor` is the colour that the NeoPixels blink when the microphone doesn't detect any new sound. Let's call this the idle state. `colorMultiplier` is an easy way to shift the colour value of the NeoPixels during the active state, when sound is detected. Set this value to a number from one to six. Anything higher may shift the value beyond 255, and won't generate any colour on the NeoPixels.

Next, we'll tell the microphone to take a baseline reading and call it `baseLevel`. Finally, with `minVol`, we'll set the minimum volume increase needed to trigger the active state. You can fine-tune the tolerance of your setup by increasing or decreasing `minVol`.



Below ♦
You'll need access to the Gemma's USB port for programming

Bottom ♦
The reflective surface of our mask complements the NeoPixels perfectly



```
blinkColor = 200
colorMultiplier = 4
baseLevel = int((microphone.value)/1000)
minVol = baseLevel + 2
```

Next, we have `wheel`, a helper function that changes a number from 1–255 into an RGB colour value that the NeoPixels can output. To learn more about `wheel`, check out Adafruit's tutorial on CircuitPython for NeoPixels: [hsmag.cc/cDYuuM](https://mag.cc/cDYuuM).

```
def wheel(pos):
    if pos < 0 or pos > 255:
        return (0, 0, 0)
    if pos < 85:
        return (255 - pos * 3, pos * 3, 0)
    if pos < 170:
        pos -= 85
        return (0, 255 - pos * 3, pos * 3)
    pos -= 170
    return (pos * 3, 0, 255 - pos * 3)
```

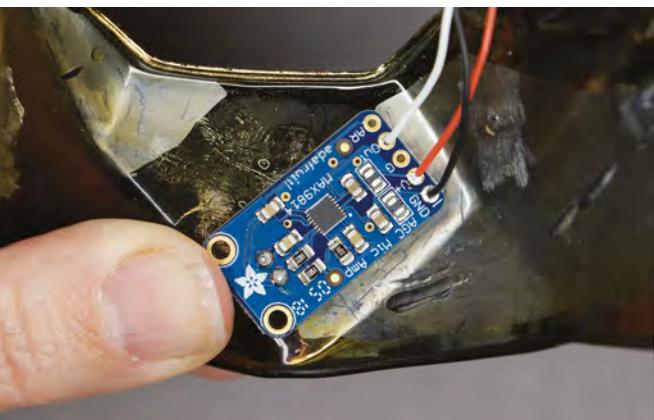
Now we're ready for the main loop! We will start by taking a reading from the microphone, and then turning this analogue value into a colour. We'll call this value `activeColor`, for use in the active state of our mask.

```
while True:
    micInput = microphone.value
    volume = int((micInput)/1000)
    activeColor = volume*colorMultiplier
```

Next, we'll use a conditional statement to create the two states of our mask. The `if` section is our active state. If the microphone detects new sound, the Gemma will set the NeoPixels to the `activeColor` and turn them all on.

```
if volume >= minVol:
    pixels.fill(wheel(activeColor))
    pixels.show()
    time.sleep(.001)
```

The `else` section is our idle state, and it contains another conditional statement. When no new sound is detected, the Gemma will choose a random pixel and a random number between 1 and 100. If the random number is 40 or higher, the chosen pixel will flash the `blinkColor`. If the number is less than 40, the chosen pixel will be black (off). Blinking the NeoPixels on and off like this creates a twinkling animation that means

**Above**

Coloured wires aren't essential but they help avoid mistakes

the mask still does something interesting even when there is no sound.

```
else:
    led = random.randint(0, (numpix-1))
    blinkChance = random.randint(1, 100)
    if (blinkChance <= 40):
        color = wheel(blinkColor)
    else:
        color = (0, 0, 0)
    pixels[led] = color
    pixels.show()
    time.sleep(.001)

pixels[led] = (0, 0, 0) pixels.show()
```

Saving the code in the **code.py** file on the Gemma M0 will automatically restart the device and run the program. Your mask should twinkle in its idle state, and speaking at your microphone should make all the NeoPixels illuminate. If your NeoPixels don't seem to be reacting to sounds as expected, try pressing the reset button on your Gemma to take a new baseline reading. If your microphone seems too sensitive, try increasing the tolerance by increasing the amount added to the **minVol** variable at the top of your code.

GLUE IT ALL DOWN

Finish up by attaching everything securely to the mask. On the front of the mask, gently pull away each NeoPixel, dab a little hot glue behind it, and press it back into place.

On the inside of the mask, remove any adhesive putty and use some small dabs of hot glue to attach the microphone. Try to keep the hot glue away from any soldered components on the board. Glue the Gemma M0 in place as well, making sure that both the micro USB and JST ports are still accessible.

Plug the battery holder into the Gemma M0, and glue it in place on the inside of the mask.

To tidy up the wires, gently gather them into bundles and affix them to the inside of the mask with hot glue. Do not glue the wires of the battery holder, in case you want to switch to a different type of power source in the future, like a LiPo battery.

None of the components should touch skin while the mask is being worn, but it's still a good idea to insulate anything that might come into contact with skin. For example, we used some Kapton tape (electrical tape would work too) to cover the back of the microphone breakout board. For comfort, you could also cover the electronics with a bit of felt for a soft barrier.

Insert batteries, and flip on the power switch to turn on your mask! Both the Gemma M0 and the battery holder have power switches, so make sure they're both on. It's fine to leave the switch on the Gemma M0 on all the time, and use the switch on the battery holder to turn your mask on and off.

SOUND IN, LIGHT OUT

Wear your mask to your next karaoke party and bring your own audio-reactive light show to the stage! Remember that every time the mask is turned on (or reset), the Gemma M0 will sample the ambient noise for a baseline. Press the reset button on the Gemma M0 to re-sample as needed. And don't forget some extra batteries!

Adding an audio-reactive feature is just one way to create wearables that come alive. Microphones and other sensors can give your project an awareness of the environment around it, and create ways for other people to interact with your wearable. Working on a great interactive project? Show us at hackspace@raspberrypi.org! □

**Above**

A dab of glue holds the microphone permanently in place

Below

A bit of tape keeps everything safe

Bottom

Now you just need a social occasion to show off your new mask



CircuitPython development the interactive way

Use Jupyter to modify your code on the fly



Ben Everard

@ben_everard

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

YOU'LL NEED

◆ **Any board running CircuitPython** e.g. a Circuit Playground Express or ESP8266. This needs to already be set up to run CircuitPython

◆ **A desktop machine running Windows, Linux, or macOS**

Right ◆ Any CircuitPython board will work, but the Circuit Playground Express comes packed with NeoPixels and sensors

W

hen writing code for microcontrollers, we usually edit the code on a computer, then upload it to a board, restart the board (wiping any data from previous runs), and

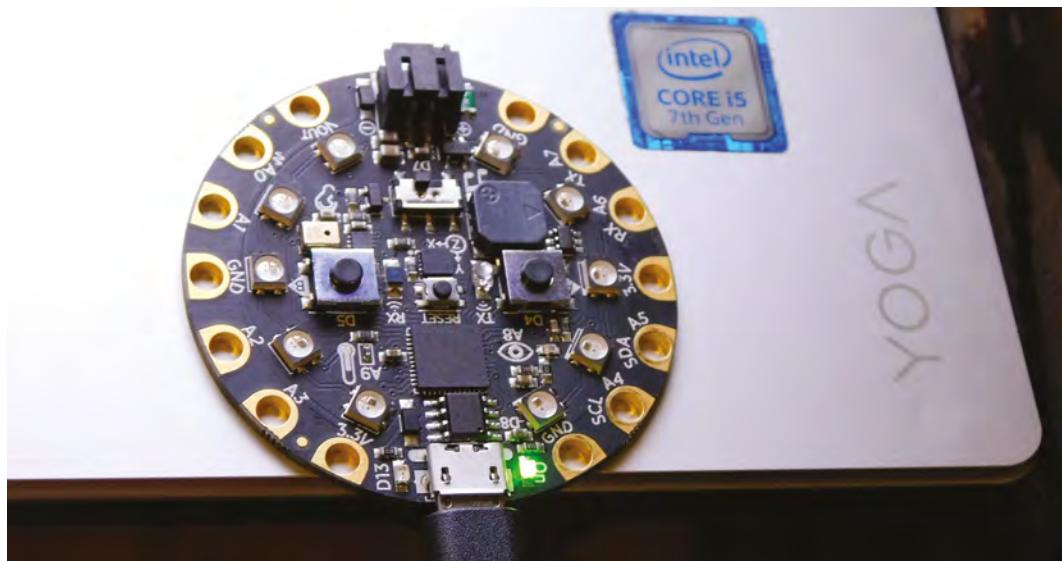
start from the top. This gives us a blank slate to run our code every time. More recently, boards have become available that allow us to type code that runs as we type it in a read-evaluate-print-loop (REPL) that is great for testing things out. However, every time we restart the machine, we lose all the code we've written. Wouldn't it be great if there were a half-way house, where we could code on our main machine, save it, and run it from a blank state, but also run it selectively as and when we want it? That's what we're going to do with Jupyter.

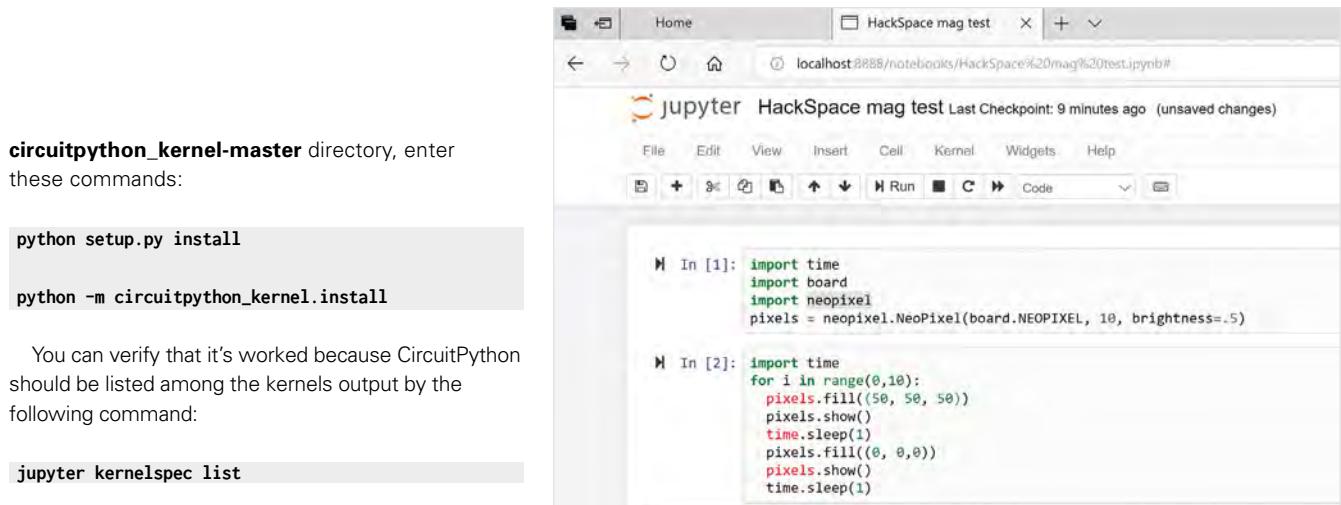
The easiest way of getting Jupyter on Windows is via the Anaconda Python distribution (on Linux or macOS, take a look at Adafruit's guidance here: hsmag.cc/ZeLQYq). Just download and install it from anaconda.com. If you follow the defaults, you'll get Jupyter as well as the Python tools.

Jupyter started life as a Python tool and although it can run a range of languages, for each language you will need a kernel – and CircuitPython is no exception.

To obtain the kernel, you have to download it from hsmag.cc/DRwyUx using Clone or Download > Download Zip. Save this in a convenient place and then unzip it.

Open an Anaconda terminal and navigate to this unzipped directory (use the `cd <location>` command to change directories). Inside the new





circuitpython_kernel-master directory, enter these commands:

```
python setup.py install
python -m circuitpython_kernel.install
```

You can verify that it's worked because CircuitPython should be listed among the kernels output by the following command:

```
jupyter kernelspec list
```

RUNNING ON HARDWARE

This works with any circuit Python board (including the ESP8266), but we'll use the Circuit Playground Express, so if you're a twelve-month subscriber to HackSpace, you can test this out with your subscriber gift.

One of the really useful bits about using Jupyter is that it allows you to run your code bit-by-bit and stop to rerun bits when you want to change code. Let's take a look at how we do this with the on-board NeoPixels.

First, we need to get everything set up, so first start Jupyter – you'll find it in your Start menu. This will open a console window and (after a few seconds) a browser tab. The new button will open a drop-down where you can select CircuitPython, and this will open a new notebook. Don't forget to plug your Circuit Python board in so the code will have somewhere to run.

Notebooks contain cells, and it's these cells that contain code. Your new notebook will contain just a single cell. As a simple test, click into it and type:

```
print("hello world")
```

Then press the **>|** button next to the cell and it should run.

Now, add a new cell (using the **+** icon) and put the following code in it:

```
import time
import board
import neopixel
pixels = neopixel.NeoPixel(board.NEOPixel, 10,
brightness=.5)
```

We've lumped all these together because they're the bit of code that we want to run exactly once to set up the board. We may fiddle about with other bits of code, but we'll need these imports and we'll need the NeoPixels set up. Click on the **>|** icon to run the cell. You shouldn't get any output other than a 1 appearing in the square brackets.

Now click the **+** symbol to add a new cell. You can group your code in cells however you want, but it makes sense to put it in chunks that you're likely to want to run together. For example, we could make a cell to blink the lights ten times:

```
import time
for i in range(0,10):
    pixels.fill((200, 200, 100))
    pixels.show()
    time.sleep(1)
    pixels.fill((0, 0, 0))
    pixels.show()
    time.sleep(1)
```

You could also create one to fade in and out:

```
for i in range(0,100):
    pixels.fill((i, i, i))
    pixels.show()
    time.sleep(0.02)
for i in range(0,100):
    pixels.fill((100-i, 100-i, 100-i))
    pixels.show()
    time.sleep(0.02)
```

With these, you can click on either one to run that block of code.

If something goes wrong, you can restart the kernel using the icon in the toolbar. If you do this, don't forget to run the cell with the setup code before you start working with your NeoPixels.

We have taken a look at a really simple example here, but it's easy to see how this can be useful, particularly for code with a long setup time. Whether you're fine-tuning the movement of a robot, or debugging a communications protocol, the flexibility in running code offered by Jupyter can make your life far easier. ☐

Above The web-based interface is easy to use, but if you run into trouble, it's well documented at: hsmag.cc/CzpdPQ

QUICK TIP

As well as running CircuitPython, Jupyter can run a range of languages including Python and Julia. It's great for interactively exploring data sets.

DON'T MISS THE BRAND NEW ISSUE!

WIN! 10 NEW CLUSTER HATS

The MagPi

The official Raspberry Pi magazine

Issue 77 | January 2019 | magpi.cc

1970s Computing
Go back in time with the PiDP-11

20 Amazing Accessories

Best Raspberry Pi kit & components

MAKE WITH RODDE

Learn Python & control electronics with Raspberry Pi

PLUS! FREE CASE, THREE COVERS & CABLES

SUBSCRIBE FROM JUST £5

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

FREE PI 3 A+*
With your 12-month subscription to the print magazine
magpi.cc/12months

* While stocks last

Buy online: store.rpipress.cc

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG
112

**DIRECT FROM SHENZHEN:
SEWING
MACHINE**

What can you make with the
cheapest sewing machine around?

PG
120

**CAN I HACK
NERF GUNS?**

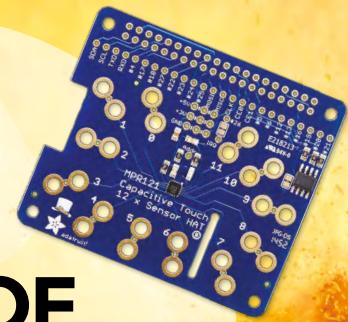
Pew Pew Pew! We pull a laser gun
to pieces and rebuild it better



PG
114

BEST OF BREED

Touch boards – ditch the buttons and
use touch input for your projects



REVIEWS

122 GIGGLEBOT

Give your micro:bit motion

126 Sensything

Monitor almost anything

128 AIR602

A cheap and powerful WiFi board

129 Brief answers to the big questions

Discover how the universe works



DIRECT FROM
SHENZHEN

Miniature sewing machine

Can you get a new, working sewing machine for under £10?

By Ben Everard

 @ben_everard



Above The sewing machine can take conductive thread, but is a bit too unreliable to make conductive circuits

Sewing machines are one of the keystone maker tools that unlock huge amounts of maker potential – like soldering irons and hand drills. Strictly speaking, there's not much that you can do with a sewing machine that you couldn't do by hand, but the difference in time taken, and precision between hand-sewing and machine sewing is so big that it makes far more ambitious projects possible.

However, all this making comes at a cost. Typically, an entry-level machine comes in at close to £100, and prices go up as high as you'd like to spend – with new features coming in at different prices. However, are these machines needlessly complicated? Is it possible to simplify things and cut the price to a fraction of

this? We decided to find out by seeing what you could get at the very lowest end of the market.

We bought a 'NeedleWork Mini Sewing Machine' from U-ZONE Top Supplier on AliExpress for £6.90, including delivery, to see exactly what you can get at the budget end of the range.

The unit is indeed mini. It fits comfortably into one hand. There are only two controls, an 'on' button, and a wheel to wind the machine through manually (when you need to reach a particular point in the stitch cycle). These are similar to, though much more basic than, the controls on a normal full-sized sewing machine. It takes four AA batteries or a 6V DC supply. It came with neither of these, but it did come with two bobbins of white thread, a spare needle, and a needle threader.



Left ↗
The foot is permanently spring-loaded down, with no release lever

Although it's a radically different form factor, the basic parts will all be familiar to someone who's used a full-sized machine. There's a foot to hold the fabric in place, feed dogs to move the fabric through the machine, and a spring-loaded thread tensioner. To sew, just slot your fabric under the foot, and press the button.

This machine can sew a stitch in a straight line, however, it's different from the stitch used on most sewing machines. On most machines, there are two spools of thread – one coming from the top, and the other from the bottom. The needle pushes one thread through the fabric, and loops it through the thread at the bottom. However, in this machine, the thread is pushed through, and a portion of it held beneath the fabric as the needle goes back up. When the needle comes back in, the thread from the previous stitch is pulled forwards and looped under. In this way, it's perhaps a little more like a single line of knitting than a typical sewing stitch. The fundamental problem with this type of stitch is that the integrity of any stitch relies on the integrity of the stitches either side of it. If one gives way, then the entire row of stitches rapidly disappears into thin air, leaving just a line of holes and length of thread blowing in the wind. This can happen if it's not properly tied off (which is a more tricky procedure than on a normal sewing

machine), if the machine skips a stitch (which is not uncommon), or if the thread breaks at any point.

We tried it with a range of threads, including the ones it came with, and conductive thread. The machine seemed to work fairly well with all of them, but none changed that basic point that the stitches were just too prone to coming out all in one go.

With this inherent weakness of the stitch, it's hard to think of a good use for this style of sewing machine. If you need to put a temporary stitch in somewhere hard to reach, then it could be useful, but given that it's a bit fiddly to get the start and stop knots in, it's unlikely to be easier than pinning in place. If it were safer, it might make a fun toy for children, but with a needle flying up and down, that's not a good idea.

It would be great to have a small, cheap, portable sewing machine for on-the-go fabric work, but unfortunately, this isn't it.

If you're after a sewing machine and money's tight, a far better option is to browse the classified sections for second-hand options. The basic way that sewing machines work hasn't changed in decades and, although there are some great features on newer models, older machines can still perform just fine, and are sometimes available for not much more than these miniature models. Older machines won't be small, or particularly portable, but they should sew things together, and that's far more important. □



Although it's a radically different form factor, the parts will all be familiar to someone who's used a full-sized machine



DIRECT FROM SHENZHEN

**ONLY THE
BEST**

Please do touch!

A roundup of touch-friendly boards

By Marc de Vinck

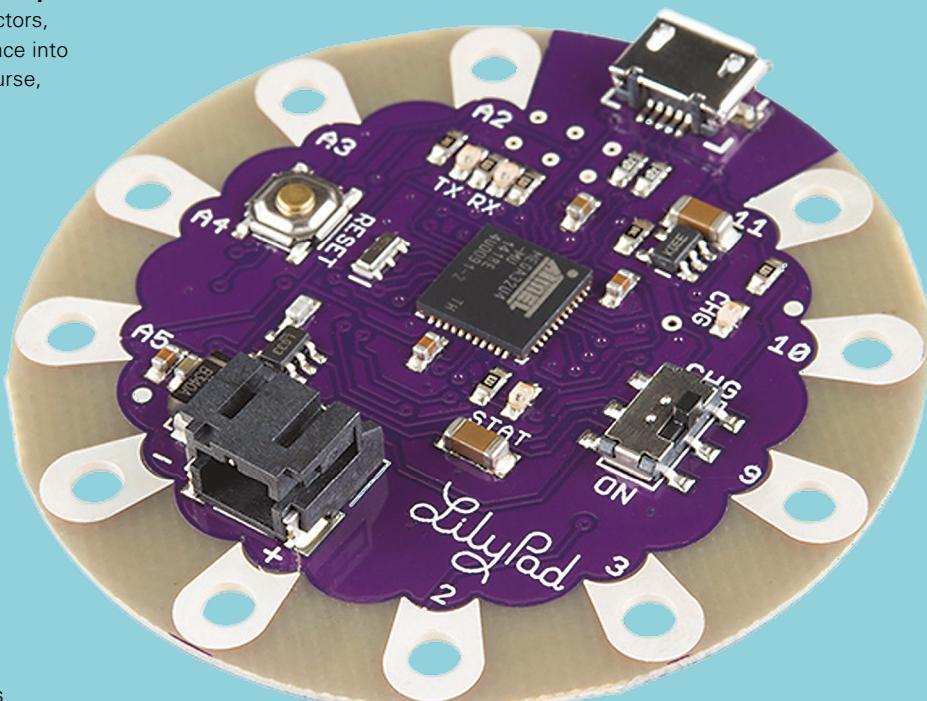
 @devinck

Last month we looked at conductive materials, and a great way of using these is to add interactivity to your projects. Using those conductors, you can turn almost any surface into a touch-sensitive area. Of course, you need a controller capable of measuring the input, and that is what we are looking at here.

Most of these products feature capacitive touch-sensing capabilities. Some, like the micro:bit, use resistive touch, but in all cases they like to be touched, thus making them great for connecting to conductive materials.

One interesting thing to note is that all these boards work really well in a solderless environment by utilising alligator clips instead of traditional soldering. Maybe the title of this article should be 'electronics and alligators'? However, they all work well with copper tape, or conductive inks and paints too.

And if alligator clips or copper tape isn't a permanent enough connection for your next project, the same footprint that makes them perfect for those types of applications also makes them perfect for sewing conductive thread. Of course, you could go old-school and just solder wires to the pads on these boards; it's the flexibility to do what you want that makes them great for a wide range of projects. □



Above The LilyPad Arduino popularised the circular, large pad design of many touch boards, but doesn't have capacitive touch itself

Bare Conductive Touch Board vs Circuit Playground Express

Microcontroller boards built for touch input

BARE CONDUCTIVE \$69 | bareconductive.com

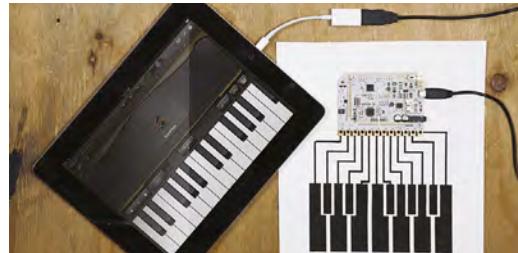
CIRCUIT PLAYGROUND EXPRESS \$24.95 | adafruit.com

The Bare Conductive Touch Board allows you to easily integrate the firm's conductive paint with its hardware, allowing you to build some interesting circuits. And although conductive paint creations is where this board really shines, it also works well with crocodile clips, copper tape, traditional soldered wires, and more. You can quickly make a capacitive sensor that detects distance, a MIDI controller, or really any type of device that detects an on-off state.

One thing to note about the board, which we found very impressive, is the fact that it comes preprogrammed with an audio guide of the features that can be found on the board. That's a really clever

In the past few years, Adafruit has really made its mark in the world of microcontrollers. We're a big fan of the Feather Ecosystem, but when it comes to making a quick and fun project or prototype, it's hard to beat Adafruit's Circuit Playground Express board. What makes it so powerful? It not only has seven capacitive sensing pads, but it also features ten on-board NeoPixels, a sound sensor, speaker, accelerometer, temperature sensor, infrared receiver and transmitter, buttons, switches, and more.

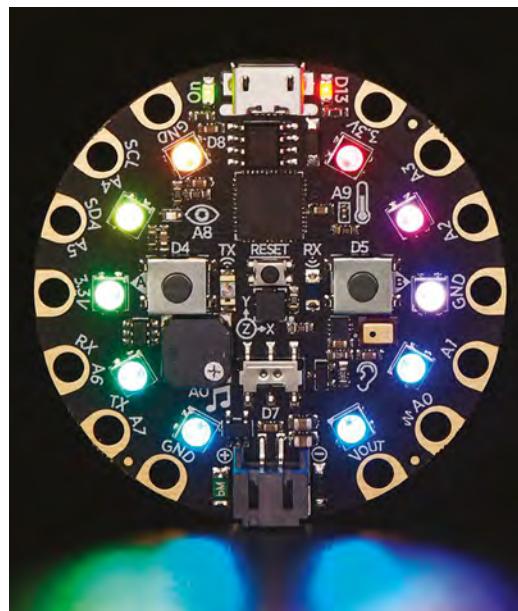
This is one powerful board, which has become a favourite microcontroller for beginners and advanced users alike. The power comes from an ATSAMD21 ARM Cortex-M0 processor, running at 3.3V and 48MHz. It also features 2MB of SPI flash storage, and a micro USB port for programming and debugging. You can even use the USB port to act like a serial port, simulating a keyboard, mouse, joystick, or MIDI controller. □



way to get users up and running fast. You can really do a lot with the Bare Conductive Touch Board, since it's also programmable through the Arduino IDE. That makes its claim of 'One tool, a million possibilities' pretty accurate. □

Left Connecting conductive ink to hardware is made simple with this board

Below An amazing amount of technology is packed into one board



VERDICT

Bare Conductive Touch Board

A well-designed board, with good documentation.

9/10

Circuit Playground Express

So many capabilities

10/10

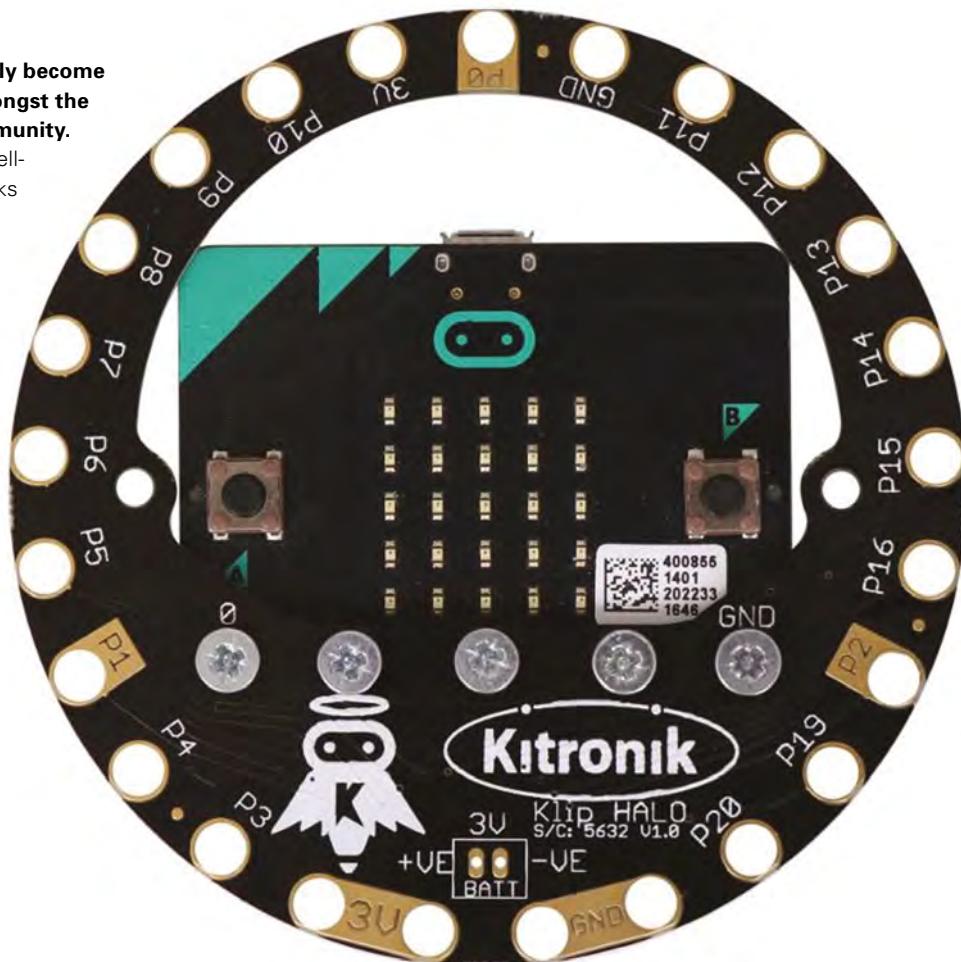
Klip Halo for the micro:bit

Upgrade your bit with extra outputs

KITRONIC £9 | kitronik.co.uk

The micro:bit has really become a popular board amongst the DIY electronics community. It's very affordable, well-documented, and packs a lot of great features, making it a great first microcontroller. However, there are times when making connections to the micro:bit that you wish you had some extra room, and that's when the Klip Halo comes into play. It allows you to easily connect to the micro:bit through one of the various large circular connection holes, using conductive thread or alligator clips. It also breaks out more of the pins, allowing you to build larger, or more complicated, projects. If you own a micro:bit, and like to work with conductive materials, the Klip Halo is a great addition to your toolbox. □

Right This board makes accessing additional pins much easier



VERDICT

A nice accessory, but restricted to resistive touch

8 /10



Left Make your own game controller from almost anything

Makey Makey Classic

Make your PC interface with the real world

MAKEYMAKEY \$50 | makeymakey.com

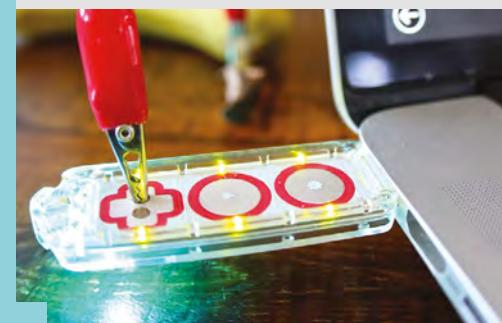
Sometimes you don't want to code up something from scratch just to make a simple button do a specific task on your computer. Or maybe you would like to make your own game controller or musical instrument, and that's where the Makey Makey comes into play. The concept is simple. The Makey Makey uses touch sensors, that you create from almost anything conductive, to simulate key presses on your computer.

You might think that a simple key press emulator is limited, but when you couple that concept with a program like Scratch, you can create some really complicated routines. The Makey Makey website features a bunch of useful applications and projects like a timing system for racing cars, a slackline scoreboard, a puppy video chat system, musical instruments, and more. This is the perfect gift for anyone who's interested in electronics but is a little hesitant to jump into soldering. □

MAKEY MAKEY GO

\$20 | makeymakey.com

When you want to create on the go, the aptly named Makey Makey Go is for you. Although there are fewer inputs compared to its bigger sibling the Makey Makey Classic, the small form factor and direct USB plug-in capabilities make creating on the go fun and easy.



VERDICT

A fun board with lots of examples online.

8/10

BEST OF BREED

Adafruit Capacitive Touch HAT for Raspberry Pi

Get caught with your thumb in the Pi

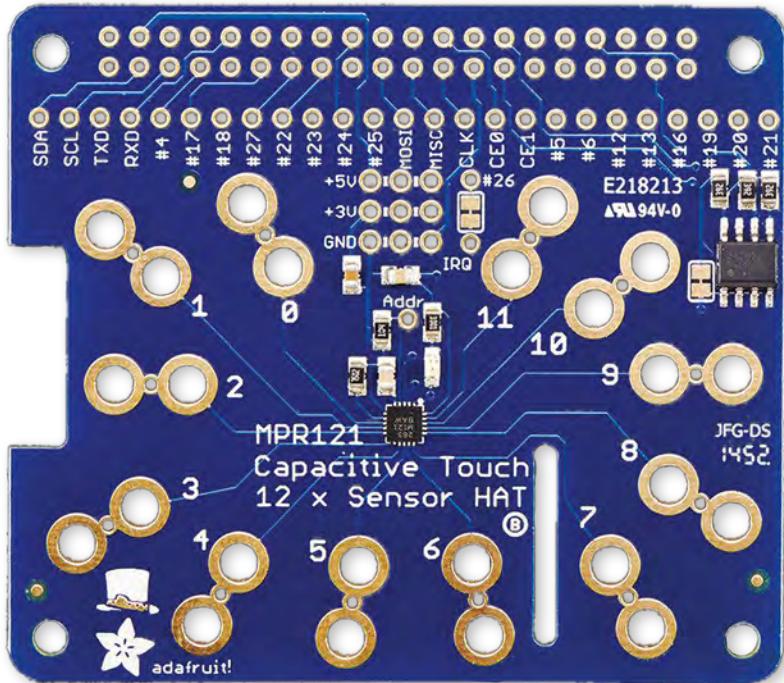
ADAFRUIT \$15 | adafruit.com

The Capacitive Touch HAT for Raspberry Pi adds twelve capacitive touch sensors to your Pi project.

The HAT has been designed with twelve figure-eight holes that allow the user to easily attach alligator clip cables. When you attach one side of the alligator clip from the HAT to something electrically conductive, your Raspberry Pi will be able to detect a touch.

Adafruit has created a Python library to help you get started fast. What's really nice is, since the Capacitive Touch HAT is designed with twelve sensors, you have enough inputs to create a musical instrument that can play a full chromatic scale. □

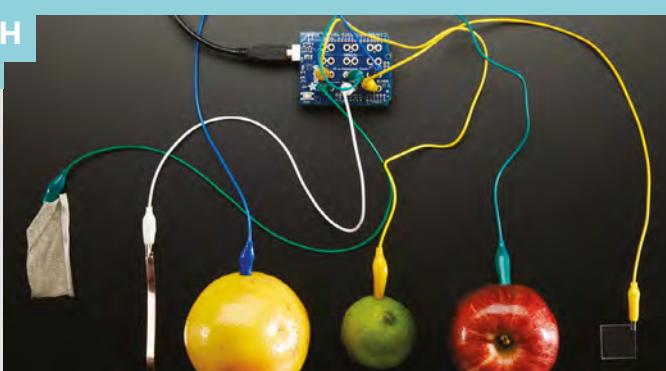
You have enough inputs to create a musical instrument that can play a full chromatic scale



12 x CAPACITIVE TOUCH SHIELD FOR ARDUINO

\$12.50 | adafruit.com

Don't worry if the Raspberry Pi isn't your thing. Adafruit makes a 12 x Capacitive Touch Shield for Arduino too. And, like the Pi version, this one supports up to twelve different input channels, making it a perfect choice for your next musical Arduino project.



Above Adding capacitive touch to your Pi is easy with this board

VERDICT
An affordable addition to your Raspberry Pi

9 / 10

CodeBug

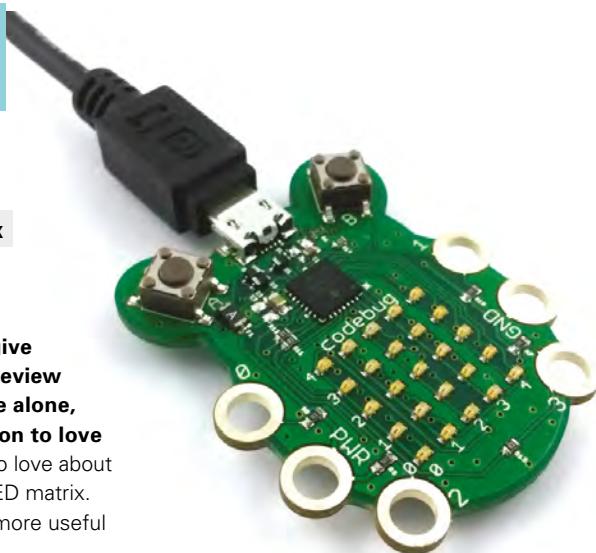
Creepy crawly touchy sensey

CODEBUG ◇ \$21 | codebug.org.uk

Although we'd like to give the CodeBug a good review just for the cute shape alone, that really isn't a reason to love this board. What we do love about it is the included 5x5 LED matrix.

That makes this diminutive board much more useful and fun.

The CodeBug was designed as a wearable device with six electrical 'legs', four of which can be used with external components like additional LEDs, buttons, or switches. Those four legs can also be



Left ◇
A fun board, with an integrated LED matrix

VERDICT
A nice board featuring a 5x5 LED matrix

9/10

FLORA V3

Touch you can wear

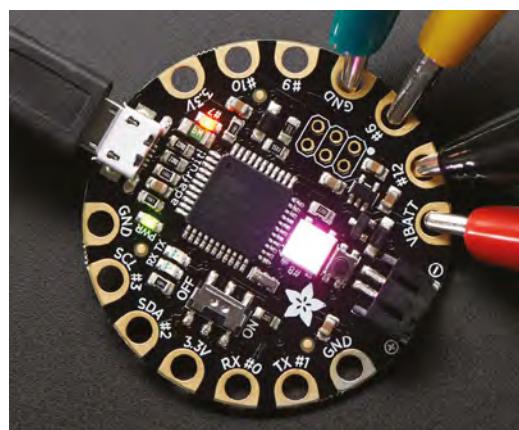
ADAFRUIT ◇ \$15 | adafruit.com

If you don't need all the advanced features of the Circuit Playground for your wearable project, the FLORA from Adafruit might be a good choice.

Not only is this board very affordable, it still packs some valuable features like built-in USB HID support, so you can emulate a key press, or a mouse.

The FLORA also features 14 pads for connecting alligator clips or conductive threads, support for external battery packs, and an RGB NeoPixel. All this makes getting your code up and running fairly simple, even for a beginner. Need some inspiration? Be sure to check out the dozens of FLORA tutorials on the Adafruit website. □

" The FLORA also features 14 pads for connecting alligator clips or conductive threads "



Left ◇
This board is perfect for wearable projects

VERDICT
Save a little time with this pre-made conductive ribbon.

9/10

A laser gun?

CAN I HACK IT? ■



Can I Hack It? A laser gun

Destroy your enemies with infrared light!



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 biglesp.com

YOU'LL NEED

Nerf Laser Ops Pro Alphapoint

COST

£31.99

WHERE

hsmag.cc/JZjTnC

Nerf is mostly known for its range of foam dart-based guns that are sold around the world. However, the firm has recently released its Laser Ops range, which combines its plastic weaponry knowledge with Bluetooth and infrared (IR). What can we do with it, and can we hack it? Well, what are we waiting for?

GENERAL CONSTRUCTION

The guns are made of a stiff, yet easy-to-work, plastic that can be worked using hands and light power tools. The chassis is made up of three main parts: a left and right side that has the profile of the gun, and the solid front 'barrel' section that is sandwiched between the profile pieces. The chassis is held together with 16 machine screws and a light amount of friction. When taking the gun apart, we see that the functional parts, barrel, IR dome, and the status panel are all modules that can be removed with ease.

POWER

Just under the barrel there is a single machine screw, which when removed gives access to a pull-out tray that is used to store 4 x AA (LR6) batteries, totalling 6V. We tested the guns with four rechargeable

Above

Two guns means that we have twice as much to hack with, so we can create elaborate IR-triggered props with sound effects

batteries (4.8V) and they worked exceptionally well. This means that we can use a 5V USB power bank; for example, those 'candy bar'-shaped USB batteries fit in the space as if made for it. All we need to do is fabricate a male USB A power lead that connects to the gun's power connections. This can be 3D-printed so that the battery slots in, or we can hack a USB lead to provide a quick power solution.

ELECTRONICS

At the rear of the gun there is a single circuit board. This board controls all of the functions present. It plays a sound when the trigger is pressed; it responds to a 'hit' and updates the rear status panel – very reminiscent to that used in the Halo video games. The main board has connectors for the status panel, which is a simple three-colour LED setup that shows the ammo and health status using green, yellow, and red. This board can be easily removed and repurposed in another project. The IR dome, a large purple plastic dome which sits on the top of the gun,

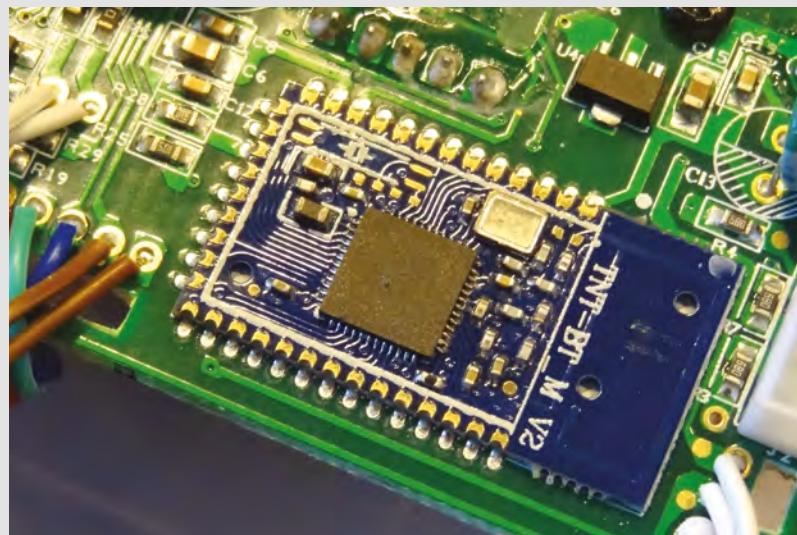
houses three IR sensors which cover 270 degrees. The IR sensor unit can be removed and connected to an Arduino, which can be used to sniff out the IR transmissions, and possibly copy them for use with a sneaky extra gun. The dome also houses two typical LEDs that light up when the player is shot – these can be easily replaced should you wish to add a little flair to your gun.

On the top of the gun there is a semi-transparent plastic shell that houses an RGB LED, which is also inside a half shell of white plastic that diffuses the glow of the LED. This LED is used to show which team the player represents – typically games are red versus blue, which means that friendly fire is disabled. However, if the gun is set to purple, then any player can shoot any other player – this is selected by pressing the magazine load, located at the bottom of the grip when powering up the gun. Simply bump the magazine until the desired colour is selected, then press the trigger to start. This leads nicely on to the trigger and magazine reload electronics, which are simple open switches that connect to the main board. Be careful opening the unit: because the trigger has a spring to return it after each shot, it will ping away across your workshop.

If you look closely on the main board, you will see a Bluetooth connectivity chip – this enables your gun to connect to an optional app on your phone, which shows live health and ammo status. This 121MB app also provides live GPS location data for players (if they are also using the app) and it can be used with VR goggles to create a virtual shooting gallery using VR drones, or set up team battles in a lobby system.

INFRARED

The Nerf Laser Ops Pro guns use IR signals similar to those found in the remote controls for your TV, Blu-ray, and cable boxes. But why? Well, lasers are a bit dangerous and will cause eye damage. But IR is safe and it has decent range, which we can easily change via a switch on the gun. So, setting the gun for outside use means we can run around and hide, taking shots when enemies approach. So what can we do with IR? Well, using an Arduino, Raspberry Pi, or other board and an IR receiver, we can determine the contents of the IR transmission. Once we understand what the guns send, we can instruct the Arduino to send that transmission. So we can build our own IR guns that react to sound, sensors, and start firing. Or, we can create our own targets that react to being shot, and maybe shoot back!



At the rear of the gun there is a single circuit board; this board controls all of the functions present

CONCLUSION

If you're building a prop weapon for a convention, then the Nerf Laser Ops Pro range is an ideal base on which to build. The electronics are modular and easy to hack, and they can be used in other prop weapons quite easily: all you need is an Arduino, a multimeter, and a little patience while you determine the pinout for the IR sensor, LEDs, etc. In fact, the status panel is begging to be used in a Master Chief cosplay. Hacking a better power source using USB batteries is a quick and easy project, and one that should be the first thing that you do. The next step is to hack the Bluetooth connection and use it to trigger effects in remote devices. Reload the magazine and have your prop future soldier cosplay light up. We can do this using a Raspberry Pi, scanning for Bluetooth MAC addresses. When discovered, it can trigger a reaction. All you need is a Pi and a little Python code! ☐

Above The main board houses all of the electronics for Bluetooth, sound effects, and keeping track of our health and ammo



Left At the top of each gun is a dome that contains three IR receivers, giving us 270 degrees in which we can be hit

Dexter Industries GiggleBot

A micro:bit robot that's ready to roll

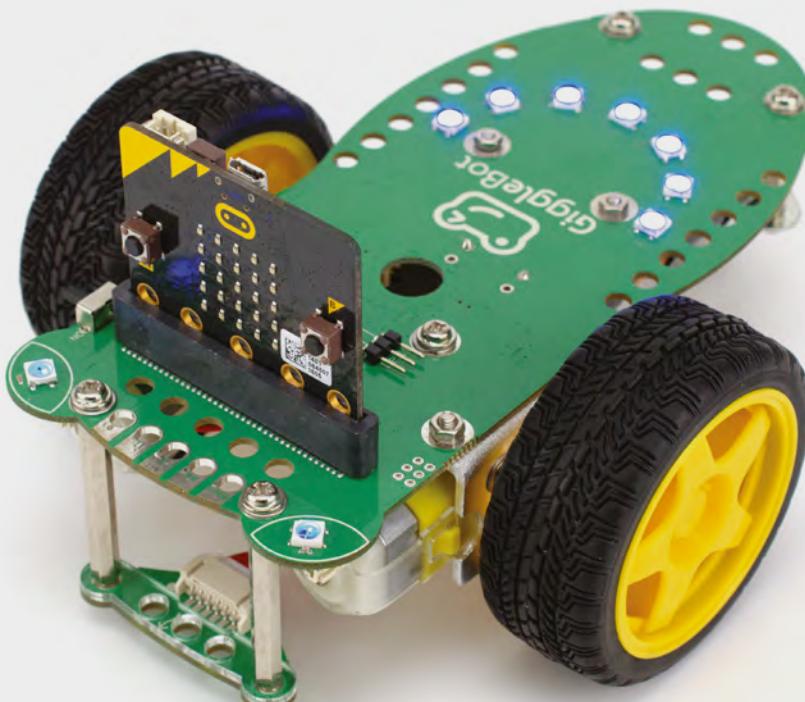
DEXTER INDUSTRIES ♦ \$60 | gigglebot.io

By Gareth Halfacree

 @ghalfacree

Below ♦
The GiggleBot brings clever robotics, and very colourful lights, to the micro:bit

After making a name for itself with a range of robotic devices powered by the Raspberry Pi, Dexter Industries is branching out into a new arena by taking its robot know-how and applying it to the micro:bit platform. The result: the GiggleBot.



At first glance, the GiggleBot feels familiar: like most of its competition in the low-cost robotics arena, it's a two-wheel design with a castor at the rear for stability. The robot arrives mostly assembled, with only the wheels needing to be pushed onto the motors' axles – something which requires a little more force than younger roboticists may be able to muster – while paying attention to the shape of the axles in order to avoid damage.

Once the wheels are on, and three AA batteries are inserted into the underslung battery holder, controlled by a physical on-off switch to the left of the robot, the GiggleBot only needs its 'brain' installing: a single micro:bit, which is slotted into an edge connector on the front of the robot with the LED matrix facing out.

EXTRA INPUTS

It's during this installation that you might start to notice a few of the features that make the GiggleBot stand out from the competition. The three main input/output pins of the micro:bit are broken out at the front, along with the 3V and ground pins, for starters; there's a pair of RGB LEDs forming 'eyes' at the front, and another seven at the rear curved into a 'smile'; there's even an additional circuit board slung under the front of the robot, providing line- and light-following capabilities.

Dig deeper, and the features keep stacking up: there's a hole in the middle of the board which allows a pen to be pushed through to turn the GiggleBot into a drawing turtle; there are connectors for a pair of

optional servo motors, in addition to the main motors already fitted; there's a pair of Grove-compatible connectors for sensors and other hardware which communicates over the Inter-Integrated Circuit (I²C) bus; and a series of smaller holes through which to mount said additional hardware, which pleasingly turn out to be compatible with Lego blocks.

The best hardware in the world is no good without software, of course, and here Dexter Industries supports three main programming languages: JavaScript, via the Microsoft MakeCode environment in block- or text-based flavours; MicroPython; and EduBlocks, a block-based environment for Python.

It's here where the first, and only, snag is to be found: the GiggleBot documentation is by far the weakest point of the bundle. Available on the official website, a quick-start guide walks you through installing the necessary extensions to support the GiggleBot in the programming language of your choice, but then abandons you to your own devices. For self-directed learners, it's a welcome challenge; for the less confident, though, some example projects would have been nice.

A HELPING HAND

There are some videos on the site, though with poor audio quality which makes them hard to follow, and a selection of short blog posts introducing basic programming concepts like button inputs and loops. These don't go into any detail about the GiggleBot itself, though, but could be useful for those for whom the GiggleBot is a first introduction to programming as well as robotics.

For those confident to explore, using the GiggleBot in MakeCode is a pleasure. Additional blocks, installed through the 'Giggle' package, allow for control of the motors, servos, sensors, line-follower, and LEDs – though, oddly, the LEDs can't easily be controlled individually – in a logical fashion, while also providing rudimentary support for controlling the GiggleBot wirelessly using a second micro:bit. The EduBlocks experience, however, is not quite so polished, missing a few of the features found in the MakeCode variant.

Regardless of language used, the act of 'flashing' the program onto the micro:bit is identical to using it without the GiggleBot: a USB cable is connected, the compiled program downloaded, and



the resulting HEX file dragged onto the removable drive which appears. It's best to do this with the micro:bit disconnected from the GiggleBot, or at least the power switch in the off position, to avoid the robot attempting to drive off with the USB cable still connected!

Above Assemble is quick and easy: just pop the wheels on to the motors

“ In addition to the main motors already fitted, there's a pair of Grove-compatible connectors for sensors and other hardware **”**

The GiggleBot lends itself well to experimentation. With its Grove connectors, it's possible to add on a wide range of extra components: Dexter Industries itself offers a distance sensor add-on (\$19.99 plus \$4.99 for mounting hardware) shaped like a pair of eyes, though any Grove component for which there is micro:bit support is also compatible. Support for the distance sensor is included in the official GiggleBot libraries, as is the ability to control the optional servo motors – used in one video on the site to add a dump truck-style tipper to the rear of the GiggleBot.

The ability to continue using the micro:bit's ports and LED matrix display is also to be welcomed, especially when many add-ons for the platform hide these away in favour of their own hardware. □

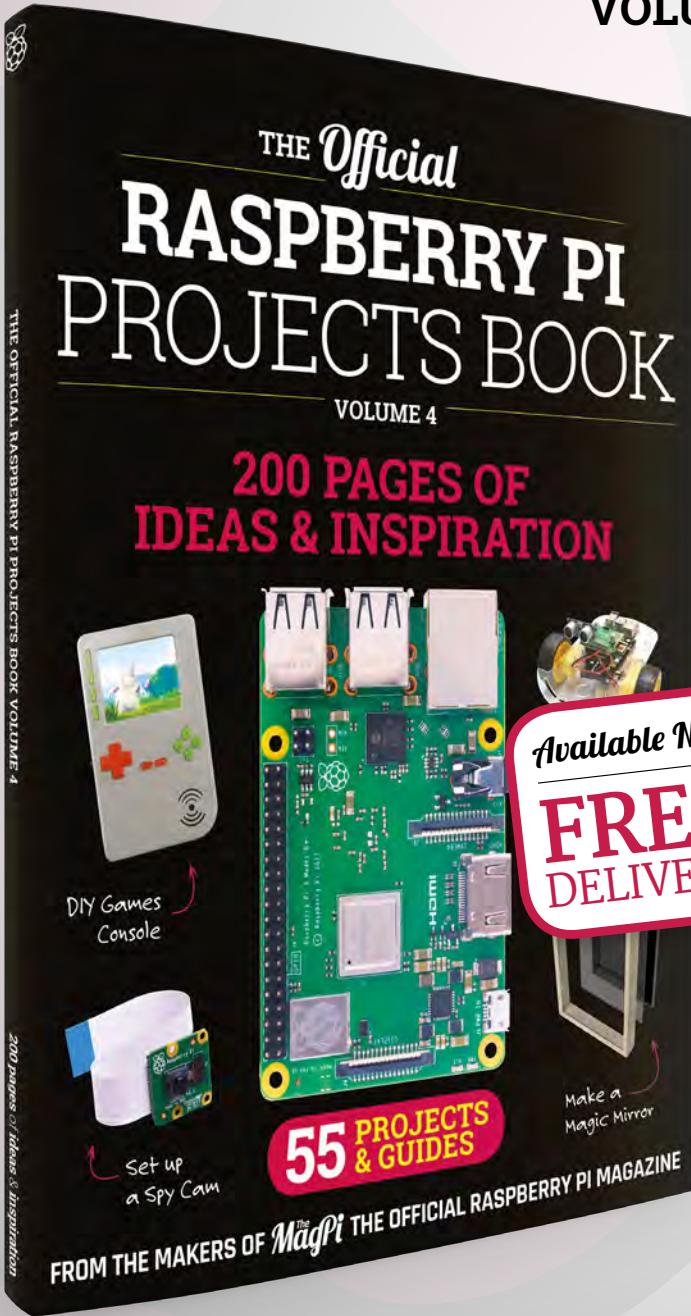
VERDICT
The GiggleBot is an impressive expandable robotics platform which needs better documentation.

8 /10

£12.99
200 pages of
Raspberry Pi

THE *Official* RASPBERRY PI PROJECTS BOOK

VOLUME 4



Amazing hacking
& making projects

from the creators of
MagPi magazine

Inside:

- How to get started coding on Raspberry Pi
- The most inspirational community projects
- Essential tutorials, guides, and ideas
- Expert reviews and buying advice

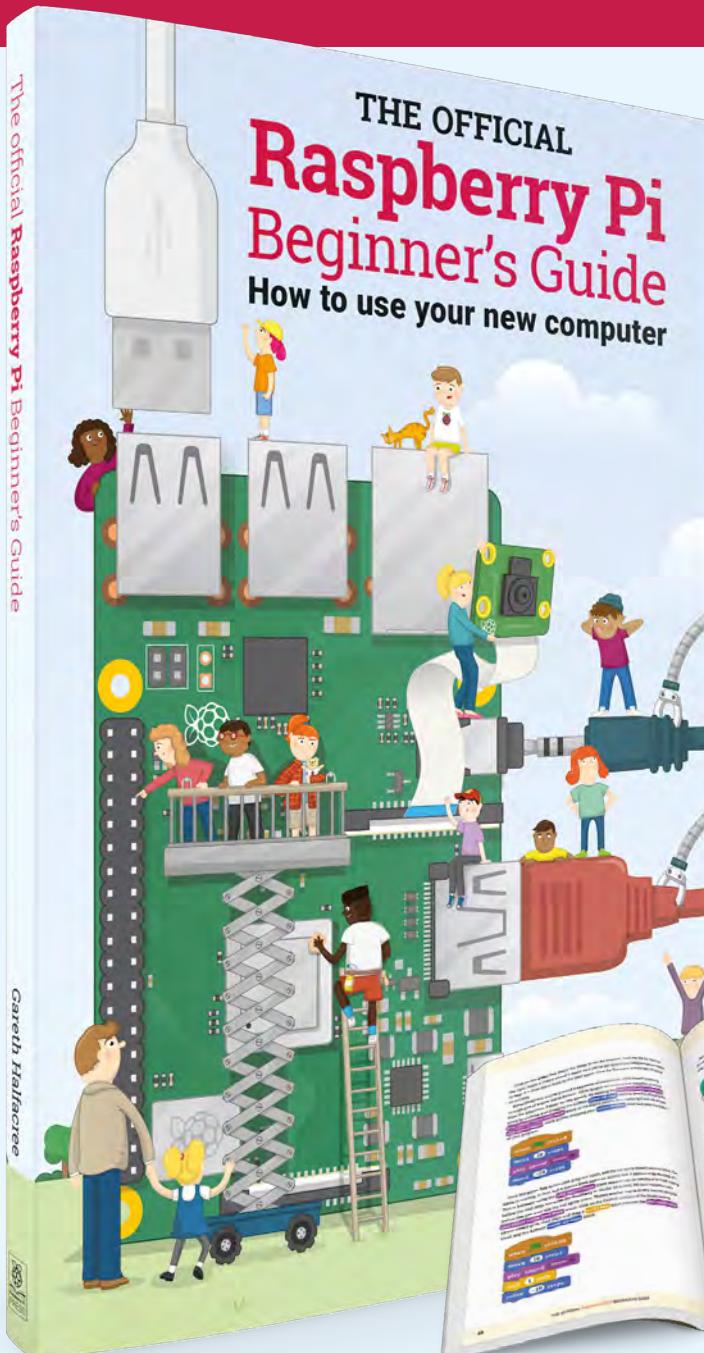
store.rpipress.cc

plus all good newsagents and:

WHSmith **BARNES&NOBLE**



THE OFFICIAL Raspberry Pi Beginner's Guide



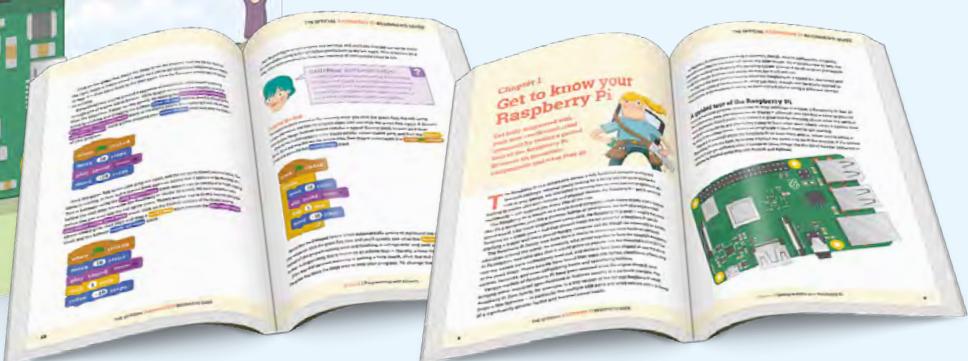
**The only guide you
need to get started
with Raspberry Pi**

Inside:

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

Plus much, much more!

£10 with **FREE
worldwide delivery**



Buy online: hsmag.cc/BGbook

Sensything

Record and view sensor data from your phone

PROTOCENTRAL ♦ From \$59 | crowdsupply.com

By Ben Everard

 @ben_everard

Below ♦
The large pads are great for prototyping, but you can also solder to them for permanent circuits

Let's be honest, there's a lot of nonsense peddled about the Internet of Things (IoT) and smart devices.

Almost anything that could conceivably have a microchip crowbarred inside has been touted as a revolutionary smart device. Most of the time, it's nothing

but marketing hyperbole, but in the rush to sell trinkets, a lot of the potential benefits of the IoT get lost by the wayside.

One great thing that the IoT can provide is monitoring. The ability to see how a particular sensor's reading changes over time without having to manually read and record it can be hugely useful, and that's exactly what the Sensything does.

The basic use of the board is to take input from a sensor, record it, and display how it has changed. There are a few different ways of making this work. Input can come in from either analogue sensors or Qwiic-style connectors, and you can read the output either using a mobile app that connects via Bluetooth (available for Android and iOS) or using a web server that can run on the board.

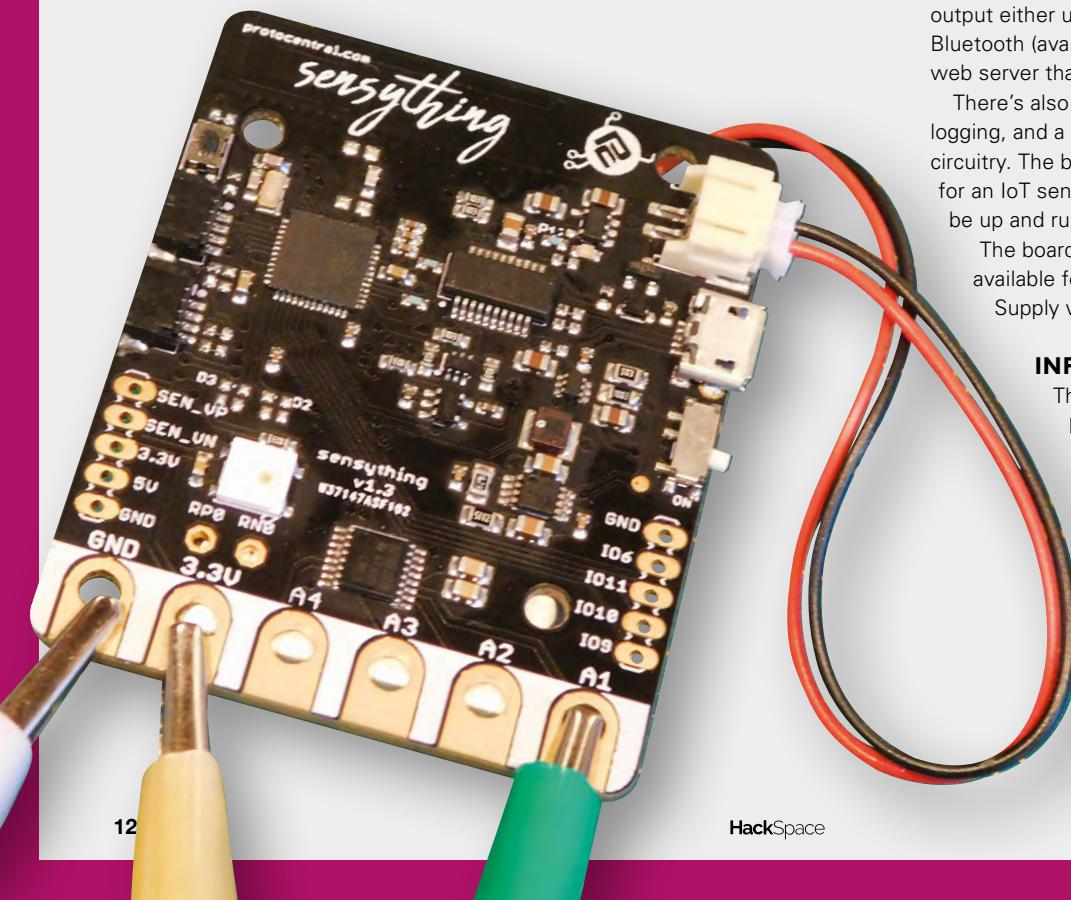
There's also a microSD card slot for data logging, and a LiPo battery adapter with charging circuitry. The board is an almost complete setup for an IoT sensor. Just connect a sensor and you'll be up and running.

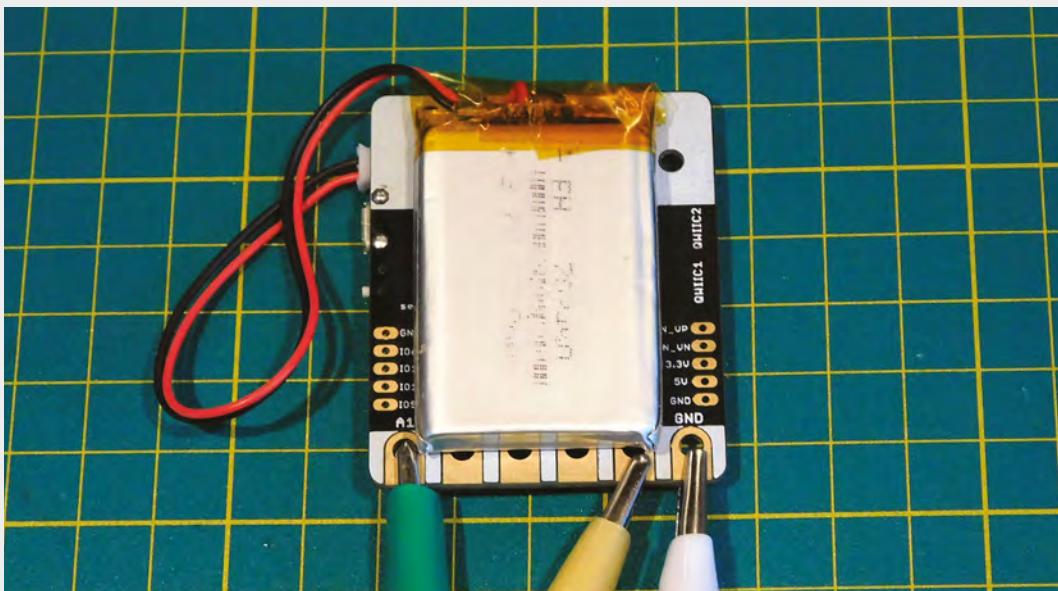
The board was initially crowdfunded and is now available for general sale through the Crowd Supply website.

INFORMATION INSIDE

The four analogue inputs can each take between 0 and 2.048 volts, so you need a sensor that can output this. In some cases that'll mean just taking the output of a sensor module, in others it'll mean building a voltage divider circuit that can convert a resistance-based sensor into a varying input. This is a really simple circuit and just requires a single extra resistor.

The biggest problem with the analogue inputs is that



**Left ♦**

Newer versions of the hardware also include an SD card slot for recording information

currently, an input to one pin leaks over and has some influence on adjacent pins. ProtoCentral has promised a firmware to solve this, but at present, this will cause you problems if you're using more than two of the analogue input channels.

As well as analogue, the board has Qwiic connectors for Qwiic inputs. These are a connector from SparkFun that builds on I²C by adding a physical standard for connecting boards together. This means no soldering, just plug in and start reading values. This is well suited for more advanced sensors than the analogue input. The data is transmitted digitally, so it's more easily calibrated. There's a wide range of Qwiic boards available at sparkfun.com/qwiic. Although there are only two connectors, Qwiic boards can be daisy-chained, so you can add many sensors in each connection.

The Bluetooth app worked well for us. It installed easily and immediately connected to the Sensything, pulling in data from both the analogue inputs and the Qwiic connectors. One slightly unusual thing is that the data from the analogue inputs is displayed on a graph, while the Qwiic data is not.

The promised web interface does not yet exist. One feature that we'd like to see, but isn't in the current plan, is the ability to upload data to some cloud service. There are obviously security and privacy concerns with this, so it won't be for everyone or every application, but if the firmware could push data to something like Adafruit.io which could then interact with IFTTT, it would be a much more powerful product.

If the default firmware isn't what you're looking for, you can program the board yourself. It's built on an ESP32 and you can upload code from either the ESP IoT Development Framework or the Arduino

IDE. There's also an additional four GPIOs that you can use. This could enable you to output some data, control some hardware, or bring in data from another source.

QUICK START

Starting at \$59, the Sensything isn't cheap, and you'll part with \$95 if you want a battery and connectors as well as the basic board. Depending on your particular requirements, you could probably build something

“The board is an almost complete setup for an IoT sensor. Just connect a sensor and you'll be up and running”

yourself for significantly less than this, but the extra money means you can get started straight away. Want a temperature sensor? Light sensor? Moisture sensor? Just wire it all up and go.

It's this ease of development that justifies the price tag. When the interesting part of your project is the thing that you're measuring, not the thing doing the measuring, then it can make sense to get an off-the-shelf data collector, and this could be it. However, the product is still new and the software is not yet ready for regular use. If ProtoCentral can sort this out, then the Sensything will be a great way of getting started with IoT development.

ProtoCentral does have a track record of fixing firmware and documentation after crowdfunding (such as with the HeartyPatch and HealthyPi), so we do expect to see this in the coming months, but it's worth checking what the current state is before ordering. □

VERDICT

Really interesting hardware waiting on the software to catch up.

L **7** /10

Air602

A cheap, capable, and connected development board

SEEEDSTUDIO ♦ From \$1.90 | seeedstudio.io

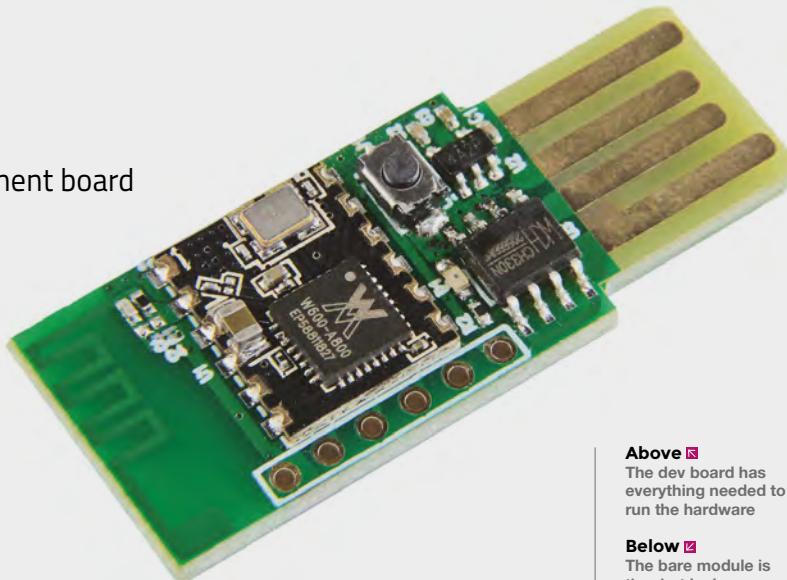
By Ben Everard

 @ben_everard

The basic Air602 module costs just \$1.90 and has an ARM Cortex M3 processor with a WiFi hardware and a hardware cryptography module, as well as a CPU and a reasonable range of connectivity including two UARTs,

I^C, I^SS, and seven GPIOs. Oh, and did we mention it's only \$1.90? This price does just get you the raw module, though – you need to splash out an extra \$1 to get it as a development board, which includes a USB for power and serial connection and an antenna. However, this form factor is limited to just five through-hole GPIOs. While this is a bit limited in terms of IOs, there are still plenty of applications where just a single IO is enough, particularly the Internet of Things (IoT) where it often makes sense for a single controller to read or control a single thing.

This price is amazing for this specification, but it comes with a huge caveat. The default firmware is Luat, which is documented almost entirely in Chinese, as is the software used to program it. It's obviously not a problem for a Chinese company to make hardware for Chinese speakers, but as English



Above ▶

The dev board has everything needed to run the hardware

Below ▶

The bare module is tiny, but lacks some key features, such as an antenna

speakers, this does make it a bit tricky for us to use it. The Luat firmware includes both the ability to execute Lua scripts and process AT commands. The latter is better documented than the former, but it's very limited and you're not likely to be able to achieve much with just these simple commands.

You can program the board using the ARM toolchains, such as Keil, but this is a more involved

“ Seeedstudio is working on an Arduino port for the Air602, but it doesn't yet have a planned release date ”

process than hobbyist tools, and not something we'd advise taking on unless you have a particular interest in the finer points of microcontroller development.

Seeedstudio is working on an Arduino port for the Air602, but it doesn't yet have a planned release date. For now, most casual tinkerers – especially those not able to understand Chinese – are better served by other boards: ones that come with support for more common development tools. With the Air602, you might save a few pounds, but it'll cost you far more in time than you save in money.

However, if a more common firmware package works on the Air602 – such as the planned Arduino IDE support – it suddenly becomes a very attractive option. It's small, cheap, and connected. If it becomes easy to use as well, then it'll be a very compelling board. □



VERDICT

Cheap, capable hardware, it will be worth a higher score if the software improves.

6/10

Brief Answers to the Big Questions

Stephen Hawking ◇ £14.99 | John Murray

By Richard Smedley

 @RichardSmedley

The late cosmologist Stephen Hawking has been the public face of science for a generation, instantly recognisable (the wheelchair, the motor neurone disease, and the pithy delivery necessitated by the laborious process of communicating via twitching facial muscles at a computer interface – and that trademark laconic computer voice).

Those terse sentences, communicated so painfully, and a lifetime spent – necessarily – in his mind, exploring the universe ("imagination remains our most powerful attribute") make Hawking's work

a model of clarity. This is the book he was working on when he died, and it was completed by, among others, his daughter Lucy, a long-time collaborator with her father on STEM education books. The authorial voice is unaffected, and there is no sign of other authors at work.

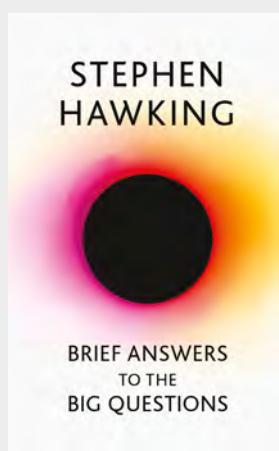
Hawking's other work and public pronouncements were used to fill the gap in this collection of ten essays answering big questions: on beginnings of the universe, and the possible existence of God, and of other intelligent life; black holes, time travel, and the conquering of space; and what we can know of the future, or if the human race even has one, and what we can do about it.

Answers impart knowledge from the sub-sub-atomic to the galactic scale, taking in many contemporary theories in physics – with the last 20 years in the field described as "really creative accounting," looking for a theory to unify quantum theory (and all of its mind-boggling consequences) with Einstein's general relativity.

Every chapter is leavened by humour – with the occasional dark turn, but a spirit of warmth and hopeful (but not blind) optimism for humanity's future. The vision is demonstrated in Hawking's projects like Breakthrough Starshot's ambitious plans for fast, interstellar travel and exploration – and a realistic look at the potential and risks of AI.

On AI, DNA manipulation, and continuing nuclear threat, Hawking says: "Our future is a race between the growing power of our technology and the wisdom with which we can use it. Let's make sure that wisdom wins."

Recommended for any non-physicist looking for real insight into current scientific understanding of our universe, in a readable form. □



VERDICT

A joyful, probing, accessible, and entertaining look at some very big questions indeed.

9/10

next issue

issue
#15

ON SALE
24 JAN

FEATURING COLIN FURZE

MADCAP INVENTOR, SHED OWNER, AND PLUMBER

ALSO

- PROGRAMMABLE SEWING
 - BOAT BUILDING
 - GLOWING CRYSTALS
 - LATHES
 - AND MUCH MORE
-

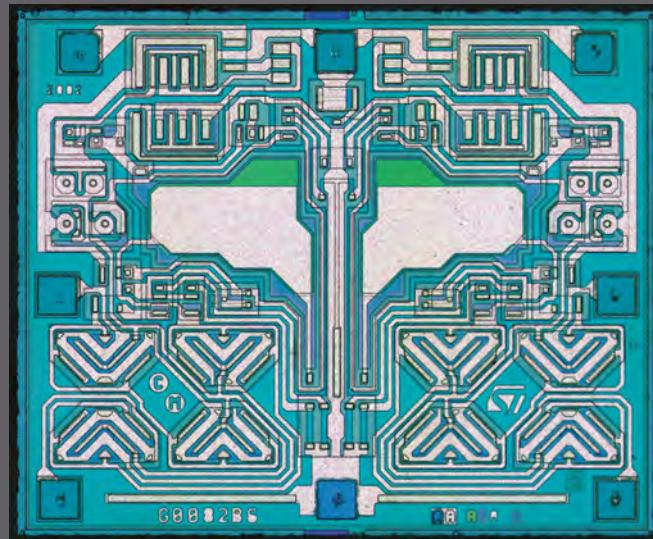
DON'T MISS OUT

hsmag.cc/subscribe



HackSpace

TECHNOLOGY IN YOUR HANDS



TL072

The TL072 chip contains two operational amplifiers with low harmonic distortion and low noise. We used one this month to condition the output from our synth

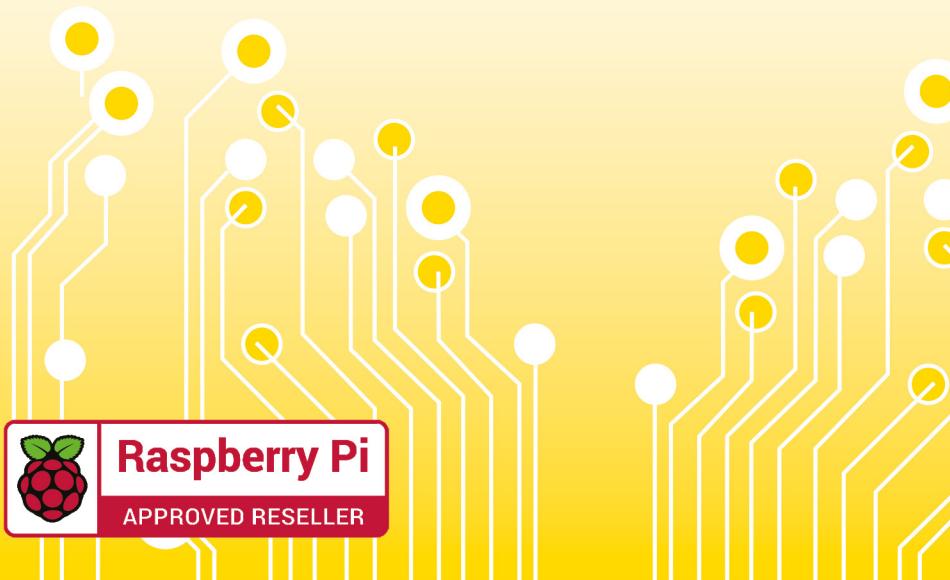
— hsmag.cc —

CanaKit Raspberry Pi 3 Complete Starter Kit



Kit Includes:

- ✓ *Raspberry Pi 3 Model B Quad-Core 1.2 GHz 1 GB RAM*
- ✓ *On-board WiFi and Bluetooth*
- ✓ *32 GB MicroSD Card (Class 10)*
- ✓ *CanaKit 2.5A Power Supply*
- ✓ *High Quality Case*
- ✓ *HDMI Cable with CEC support*
- ✓ *MicroUSB Reader*
- ✓ *Set of Heat Sinks*
- ✓ *GPIO Quick-Reference Card*
- ✓ *CanaKit Quick-Start Guide*



Raspberry Pi

APPROVED RESELLER

Available for worldwide shipping at:

WWW.CANAKIT.COM

\$74^{.99}
US DOLLARS

£59^{.99}
EXCLUDING VAT

€64^{.99}
EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation.
CanaKit is a registered trademark of Cana Kit Corporation.