

MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc

June 2018

Issue #07

Jun.2018
Issue #07 £26

9 772515 514006

07



BUILD A Z80 COMPUTER

Retrocomputing with the ultimate homebrewers' CPU

INTERNET OF EVERYTHING

Build your own smart devices to take over the world*

*Or make sure your plants are watered

HELEN STEER

On unicorns, making, and why millennials need to embrace creativity

DIGITAL BLACKSMITHS

How 3D printers are recycling African waste

SPHERIFICATION LASER CUT GEARS ERASERS UPCYCLING

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



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

APPROVED RESELLER

Raspberry Pi Zero W

Available at CanaKit!





Welcome to HackSpace magazine

Electronic devices have had interfaces for as long as there has been electronics – radios had knobs, lights had switches, and computers had keyboards. The revolution in the Internet of Things is that these physical interfaces, which were only usable by humans, become software interfaces (APIs as they're known) that are usable by computers.

Commercial product makers are still clumsily stumbling around with narrow functionality and locked down app-controlled platforms

stages of the Internet of Things, and it's far too early to see what direction it will go in. Commercial product makers are still clumsily stumbling around with narrow functionality and locked down app-controlled platforms. It's up to us hackers and makers to build the tomorrow we want to live in, and there's never been a better time to start building. There's a wide range of great hardware products to use as building blocks and the software is becoming easier to use all the time. Take a look at our cover feature on page 32 for inspiration, then get out and get making!

BEN EVERARD

Editor ben.everard@raspberrypi.org

GET IN TOUCH

- hackspace@raspberrypi.org
- <https://facebook.com/hackspacemag>
- <https://twitter.com/hackspacemag>

ONLINE

- hsmag.cc



EDITORIAL

Editor

Ben Everard
ben.everard@raspberrypi.org

Features Editor

Andrew Gregory
andrew.gregory@raspberrypi.org

Sub Editors

Nicola King, Jem Roberts

DESIGN

Critical Media

criticalmedia.co.uk

Head of Design

Dougal Matthews

Designer

Lee Allen

Photography

Brian O'Halloran

CONTRIBUTORS

Lucy Rogers, Andrew Huang, Mayank Sharma, Peter Kent, Cameron Norris, Jenny List, Graham Morrison, Dave Astels, Alex Eames, Jorim Bechtle, Ricardo Caja, Andy Clark, Les Pounder, Sophy Wong, Lynn Andrews, 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](tel:+44(0)2074294000)

SUBSCRIPTIONS

Select Publisher Services Ltd, PO Box 6337, BH1 9EH
[+44 \(0\)1202 586 848](tel:+44(0)1202586848)

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

118



06 SPARK

- 06 Top Projects**
The Guggenheim of DIY projects
- 16 Objet 3d'art**
3D-printed shadows and magic
- 18 Meet the Maker: Spencer Owen**
The Z80, chip of dreams, returns to claim its throne
- 22 Columns**
How to – the horror! – interact with human beings
- 24 Letters**
In praise of good old radio (and gin)
- 25 Kickstarting**
Get your hands on an electric adding machine
- 26 Hackspace Urban Hax**
A makerspace in the workshop of the world

Cover Feature

The cover features large, bold text: 'INTERNET OF EVERYTHING'. Below it is a quote: 'You've heard of 'The Interconnectedness of All Things' – you'll be surprised what you can connect these days...'. The number '32' is in the bottom left corner.

31 LENS

- 32 Internet of Everything**
Network your makes and build your own IoT Skynet
- 48 How I Made: Mini Cooper lights**
Control the lights on a Lego Mini Cooper from your phone
- 52 Digital Blacksmiths**
How 3D printing out of waste materials is building Africa
- 58 How to grow your hackspace**
Lessons in diversity from Oxford Hackspace
- 72 Improviser's Toolbox** Erasers
Bouncy rubberised toys ready for your Dremel

Tutorial

School of making:
Laser-cut gears



90 We love the smell of laser cutters in the morning

48



Direct from Shenzhen**Microscope****116**

Make tiny things appear bigger with the magic of optics

06**Interview****Helen Steer****64**

Why 'maker movement' is a buzzword (and that's fine)

**122****77****FORGE****78 SoM Arduino**

Temperature and humidity sensors the lazy way

84 SoM Homebrew computing

Only the elite keep their breadboards this tidy

90 SoM Laser-cut gears

Build a turntable with lasers and mathematics

96 Tutorial ESP32

Get more bang for your buck with this baby board

100 Tutorial PCB etching

Build your own PCB factory

106 Tutorial Spherification

Take your gastronomy to molecular levels

110 Tutorial Electroluminescence

Decorate the shed with an EL wire sign

115**FIELD TEST****116 Direct from Shenzhen Microscope**

Augment your puny human eyes with superpowers on the cheap

118 Best of Breed

The best tools for removing broken screws

122 Can I Hack It?

Make the robots welcome their new human overlords!

124 Review NFC nail stickers

Discover near-field communication literally at your fingertips

126 Review BeoCreate audio board

Bring high-end audio speakers into the digital world

128 Review Marty the robot

A low-cost programmable robot for kids and teachers

129 Book Review When Thea LED the Way

Inspirational illustrated tales of engineering

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.

Bike Trailer

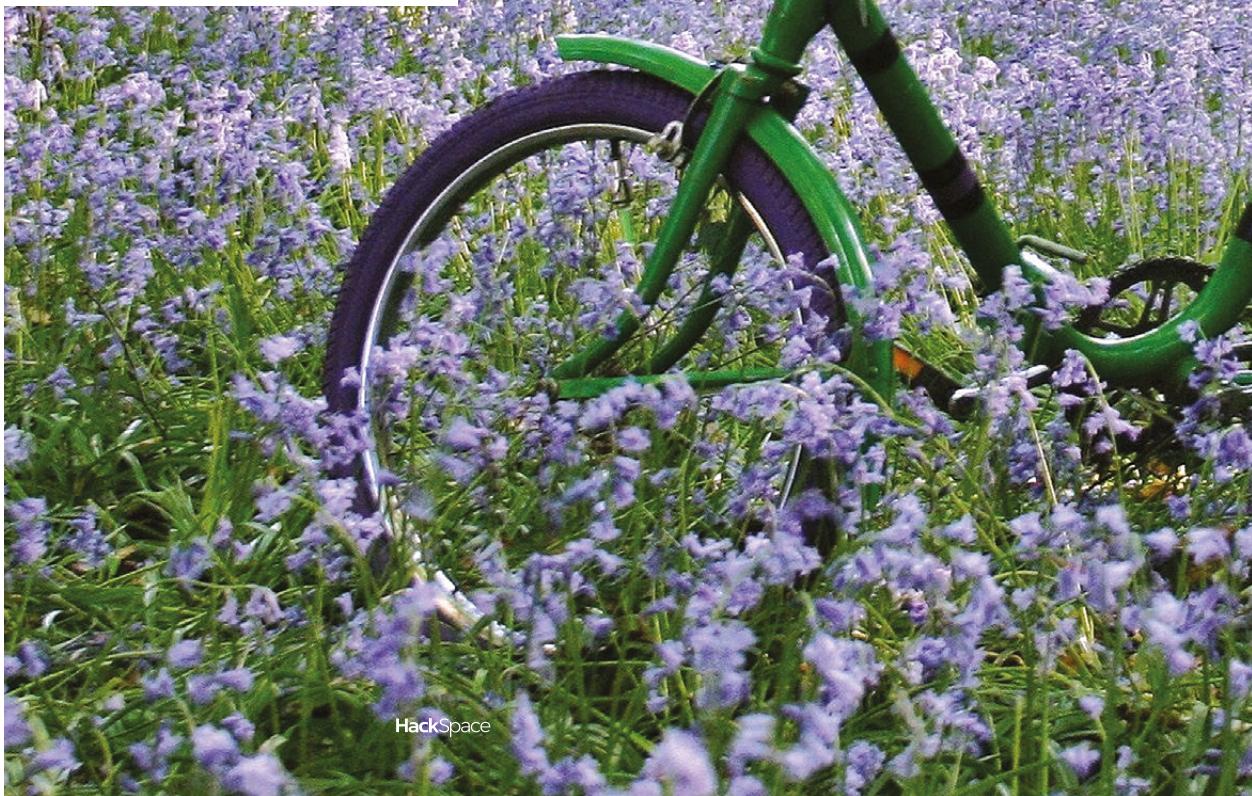
By Carolyn Ryves

@CazRyves

Alot of my projects involve bicycles, and using them to power different appliances/inventions – everything from boom-boxes to paint-spinning machines and bicycle bingo. The idea of the wagon was to create a fun trailer that I could store a pedal-powered sound system in, and use to transport it locally to events to power music or a small street-based open mic.

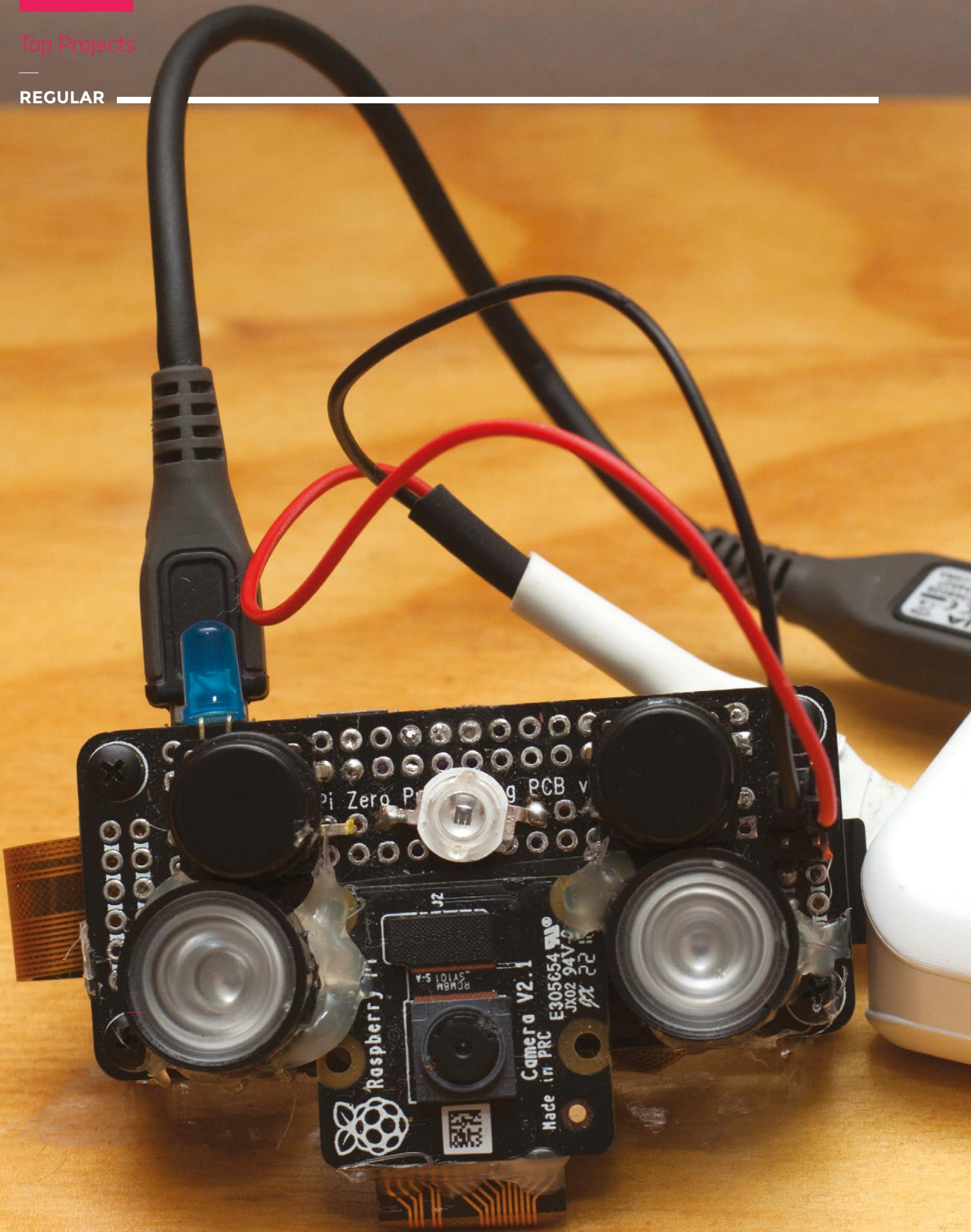
So this was really the start of the project... my next part is to turn some old suitcases into speakers – and maybe try to make the bicycle look like a horse!

Someone donated their old bicycle trailer to me and this was the foundation and original inspiration to build from. The sides and main frame were made from scrap ply and other offcuts of wood, and the doors were cut out from panels of my old childhood wardrobe! The support for the canopy cover is made from blue water pipe, and the canopy was made by dyeing an old curtain and attaching it to a camping groundsheet so it is waterproof... I'm a fan of using scrap and recycled materials and seeing where they lead me. (Sometimes, to an endless pit of despair!) □

**Right**

Carolyn's bike trailer was upcycled from old bits of wood including wardrobe doors





IR photo installation

By Felipe Gomes

 felipesg.cargocollective.com

This project consists basically of two Raspberry Pi boards, one for taking pictures in dark environments and the other for receiving these photos, via WiFi, and printing them on thermal paper.

The idea behind this setup came after an invitation to do an art installation for a party, where the act of having the photos printed from an instant camera can generate a buzz among the guests, and the lack of light would be a problem in the documentation of the event. The camera's low profile aims to capture more spontaneous moments.

The camera is made from a Raspberry Pi Zero W, a Pi NoIR Camera, and three IR LEDs as a flash. It runs a Python script that controls two buttons (one for photo and the other for video, both for power-off). It should be simple enough for any guest to use, as we are planning to share it during the party.

The printer also uses a Raspberry Pi Zero W, but with a thermal printer inside a wooden totem that will stay at a fixed point, also accessible to all guests.

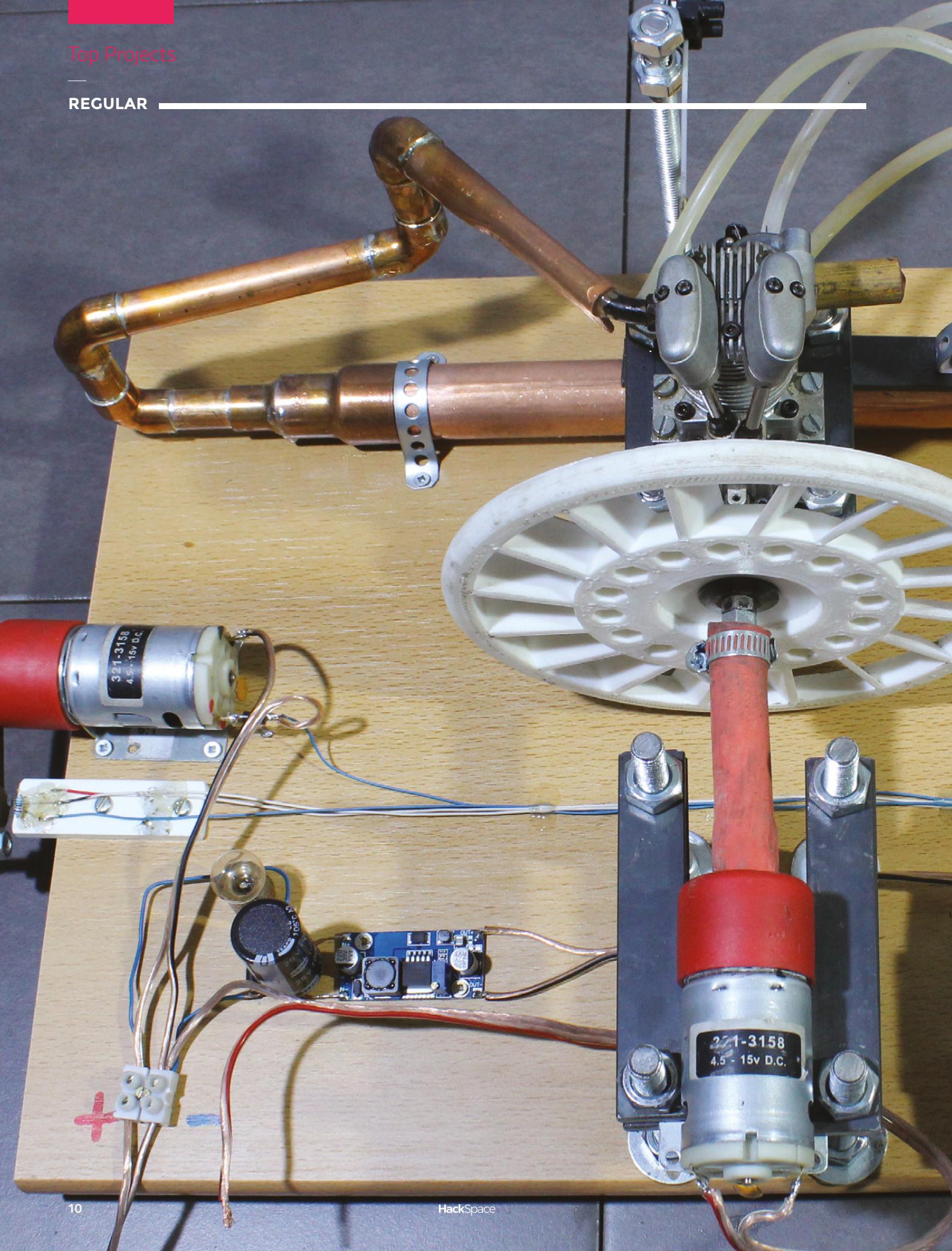
The two parts of the installation use battery power to increase the range and the possibilities of placement.

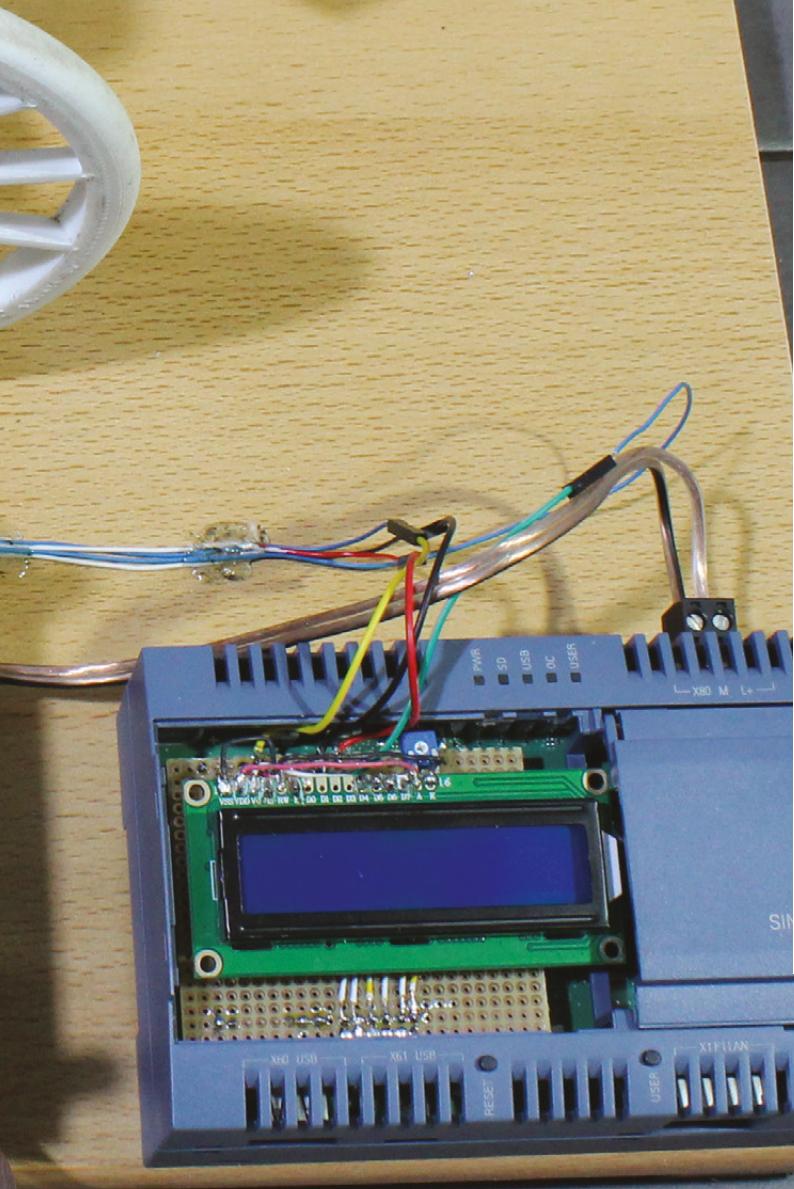
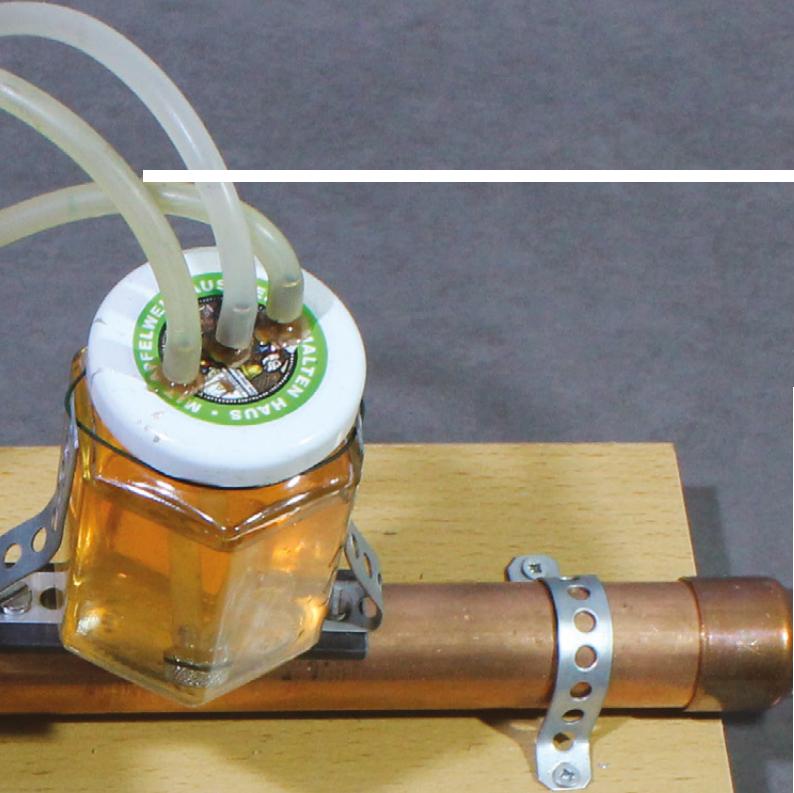
You may notice that the camera lacks a nice case. I'm still prototyping the hardware so I hope to print a case soon. Here in Curitiba [Brazil], we don't have any makerspace or fab lab, so it can be kind of hard to find a 3D printer. □



Left ♦

The wooden totem was used first to show results of an architectural photography workshop that Felipe gave last year





Methanol-powered computing

By Norbert Heinz

 HomoFaciens.de

T

his contraption was built by Norbert Heinz to illustrate how much energy your computer uses.

By turning a handle (or, more awesomely, by burning methanol in a four-stroke engine), the user gets a feel for how much power the computer demands (Norbert has also made the system run using an Arduino):

"I started diving into electronics during my apprenticeship as a physics laboratory assistant. Learning how semiconductors work was a great thing to me. I bought my first computer in those days, which was an Amiga 500 – I'm no longer a youngster – and started learning how to do coding. I scraped the parallel port of my Amiga during my early hardware experiments..."

"The idea behind my internet pages is to share knowledge with mankind, as well as getting a deeper understanding of all the stuff for myself by simply writing it down. HomoFaciens is a made-up word for making humans (in short: makers), derived from Homo sapiens, which means 'the wise man'. Creating animations is a great tool to share knowledge, so I started a YouTube channel a couple of years ago. The four-stroke generator is a less academic project – learning physics should be fun as well. My RoboSpatium is another fun part of my project. You can drive my camera-equipped robots in my attic and my garage for free."

"With the hand-cranked computers, I wanted to give people a better understanding of electric power and losses or efficiency. It's a nice demonstration; young as well as old kids enjoy powering a computer through a geared DC motor – at least for half a minute."

"The motors were left over from previous projects, and after a short test I saw that it worked. I added the four-stroke engine because I had one and I simply like the sound of those machines." □

Left

"I added a four-stroke engine because I had one." There's no better reason!

3D contraptions

By Colin Catchpole

 Bigbangcollectables

I started making in 2016, after my father brought me my first printer, a CTC Makerbot clone that really didn't work very well! I soon discovered Thingiverse, and was amazed at the amount of cool ideas people were putting on there for free. I started downloading and printing models that I liked, but the printer wasn't very good and I really wanted to design my own things and contribute to the community. I was going to have to teach myself 3D modelling.

I soon learned that it is possible to modify and remix STL files, and I began experimenting with the walking tanks that generally consist of existing models that are mashed together to give me what I want. A big thanks to all the designers that let me use their parts for things they never imagined.

I use A123D, Meshmixer, Netfabb, and Microsoft 3D Builder for remixing and repairing mesh files.

I have just started a Facebook page, named Bigbangcollectables, with the aim of one day opening a shop for people that want to buy finished models without the effort of making and painting them. Check it out, and please like and share or just drop by for a chat: I'm always happy to give advice or shoot the breeze.

I've come a long way in just over two years. I'm improving all the time and thoroughly enjoying the modelling and building experience. I'd like to thank my friends and family for the support, and especially my dad for buying me that first printer and generally putting up with my crazy ideas.

I love 3D printing and modelling because it is the only medium available to Joe Average where you can go from concept to object within hours. It's awesome! □

Right □
Colin's original purpose for the printer was to make parts for his scale radio-controlled aeroplanes, but he had no idea how to design them. After months of trial and error and YouTube tutorials, he had enough knowledge to design a model





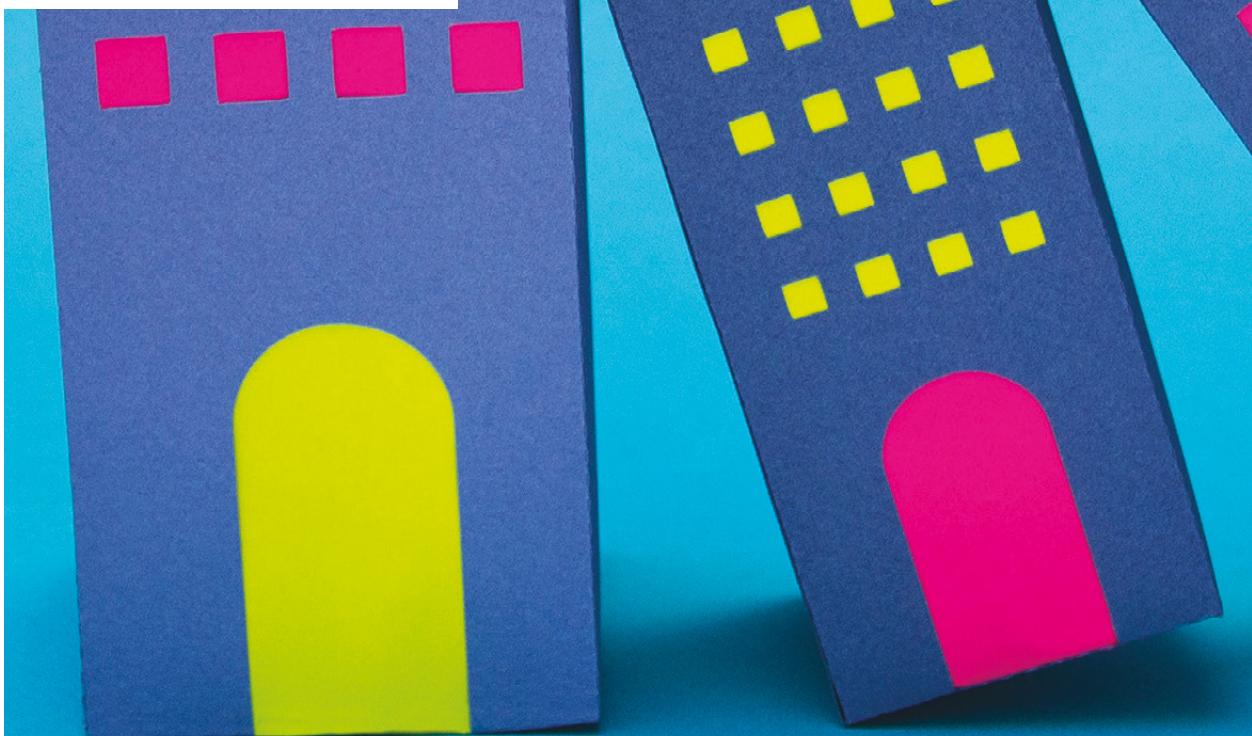
Gatos Furiosos

By Estudio Guardabosques  guardabosqu.es

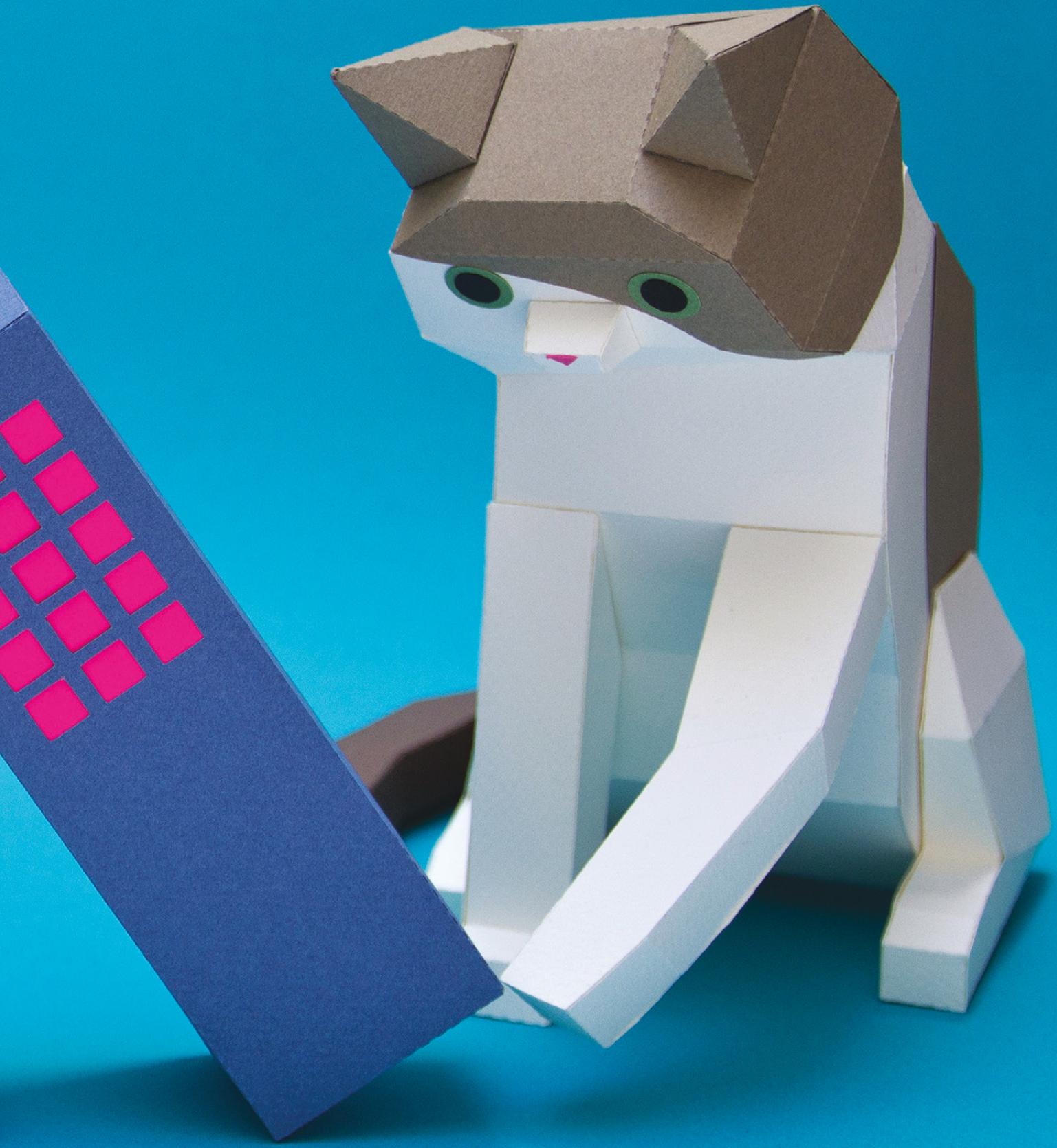
If you've been inspired by the papercraft that we crammed into issue 6, and are hankering after more of the same, here's a whimsical take on the art form from Argentina. We're no good with foreign languages, so we reckon Gatos Furiosos probably means 'horrible little allergy-causing creatures – basically the poor man's dog'. Whatever it means, the idea and execution are brilliant, and if you can fold a piece of paper, you can have a go yourself. Here's Juan Elizalde to explain more:

"Guardabosques is a Buenos Aires-based studio, formed by Carolina Silvero and Juan Elizalde, working in design and illustration, mainly with papercraft.

"Their personal work has a base in nature: they enjoy investigating and learning about the species they translate to paper, and to spread that curiosity to everybody else. They like the idea of being able to use their tools as designers to introduce all these species, from a different place, to the people." □



Right □
You can't reason with giant cats. They destroy cities out of boredom!



Objet 3d'art

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

A stereographic projection is a way in which a round shape (such as the earth) gets mapped onto a flat one (such as a map). As things that are curved become flat you get distortions, which over the centuries have given map-makers cause to squash and stretch different countries of the globe to either over-emphasise the importance of the global west or to make navigation easier, depending on your viewpoint.

This stereographic projection, by Thingiverse user Henryseg, goes in the opposite direction: it takes a grid of squares and deforms them into a basket, revealing the original shape in shadows when a light is placed directly overhead. Clever. □

↗ hsmag.cc/aYpBtk

**3D
PRINTING**

Supplied by
3D HUBS

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





On page 64 we talk to Helen Steer, who's making all sorts of things to show kids, and curious adults, that real things don't have to be complicated to be beautiful and functional. Our favourite of these is this programmable unicorn. It's a simplified kit to help kids build and program their first robot (to the extent that it uses continuous servos rather than motors, so kids don't have to learn what an h-bridge is).

Like real unicorns, this one has a 3D horn printed out of reactive filament which changes colour from white to purple when it's exposed to sunlight. □

↗ hsmag.cc/doxokO

Meet The Maker: Spencer Owen

Forwards to the past!

Right
“I thought this would be great for a bit of beer money for something on the side to help cover my own costs, but the Tindie sales were going up and up...”

Below
An RC2014 in a laser-cut wooden case

Young ‘uns today might not remember it, but there was a time before touchscreen phones and internet-connected devices.

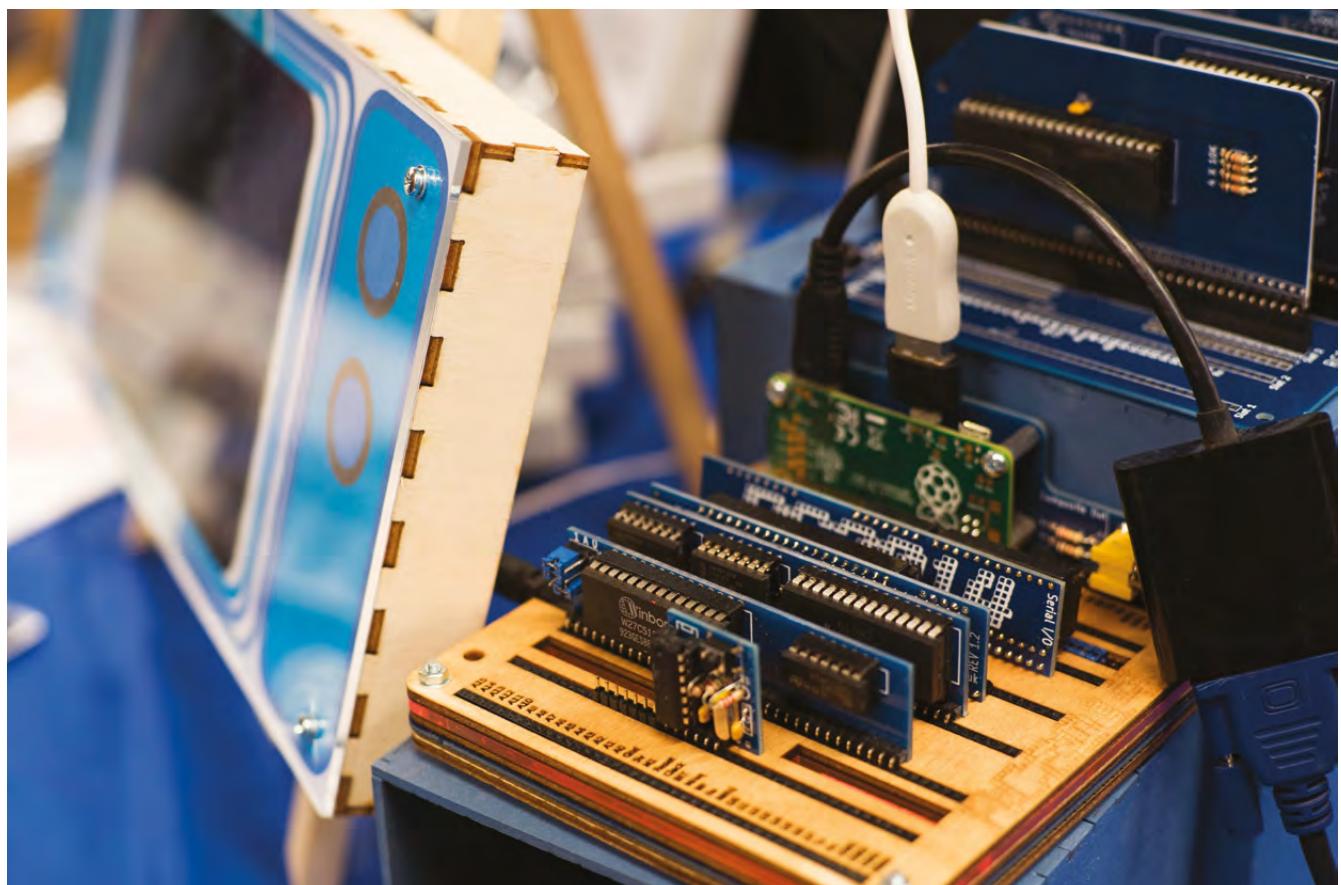
Screens were curved, games came on cassettes, and two processors fought for dominance: the Z80 and the 6502. Which one was best? Well, the answer depends on who you ask, but here at HackSpace magazine, we’re big fans of the Z80. We caught up with Spencer Owen, whose



RC2014 Z80 computer kits are bringing people back to this processor. Take it away, Spencer...

DAYS OF EARLY SPENCER

Originally, it was my own personal project to see if I was able to design a computer, the sort of thing they had back in the 1980s. It turns out that, yeah, it was possible. I made up a machine on a breadboard – Z80-based because that was the chip I had lying about, a donation to Nottingham Hackspace. I





wondered if I could put these together, and get something that worked. Yeah, I can run BASIC! I wondered, can I make a PCB out of this? I designed some PCBs and they worked! Great! I was happy.

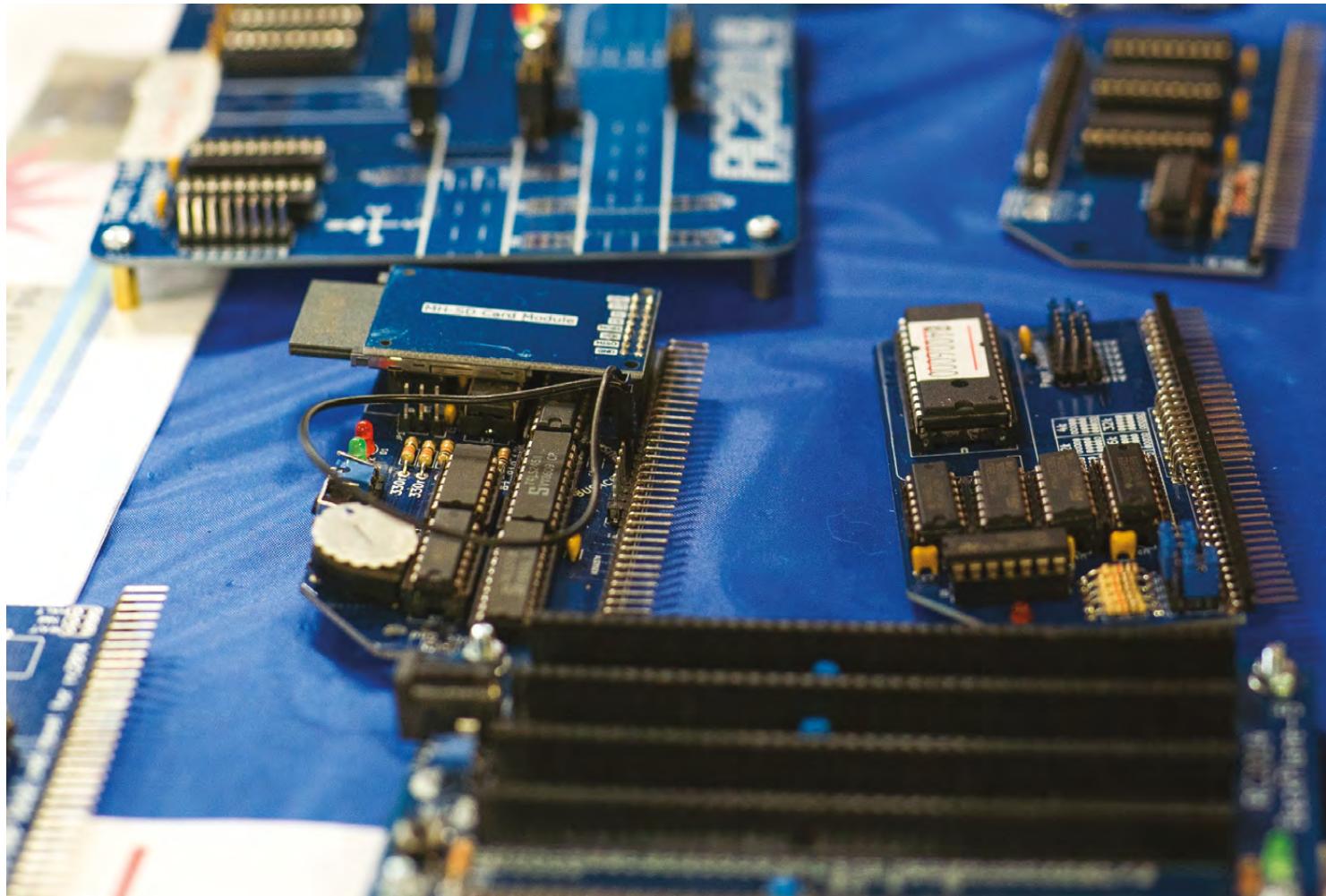
A couple of people said, "I like your computer, have you got any spare boards?" I had a spare set, so I gave them to them free of charge. Someone else wanted a set, so there you go. I'd run out, but more people wanted them. I thought I'd get some more printed and list them on Tindie to see if anyone wanted to buy them. As long as I made my money back, I'd be happy – and I sold out those ten in no time at all. I had to get some more, and people wanted the components to go with it, they wanted some other options – better versions of various modules. Before I knew it, I was working almost full-time on that, even though I had a full-time job, so I got to the stage where I quit the day job. I was a network engineer and server engineer for 16 years. I quit the day job and the RC2014 has been paying the mortgage for the last couple of years now.

"A couple of people said, "I like your computer, have you got any spare boards?" I had a spare set, so I gave them to them free of charge

Because I started selling through Tindie, I [initially] thought that this would be great for a bit of beer money for something on the side to help cover my own costs, but the Tindie sales were going up and up and up, and in the job I was in at the time, my enjoyment was going down and down and down. After working a full eight-hour day at work, and driving, then working for another six to eight hours, something had got to give. There were one or two things that happened at work that made things a bit easier, so I thought, 'I'm out of here'. My honest expectation was that if I sold RC2014s full-time, I'd still need some sort of back-up job – [e.g.] work in Sainsbury's stocking shelves – just so I've got ➤

Above All good computers need a rainbow ribbon cable or two

REGULAR



Above The RC2014 is modular and made up of a series of PCBs mounted on a backplane

Right The RC2014 brings modern protocols, such as HDMI and WiFi, to the Z80 (via a Raspberry Pi Zero and ESP8266)





some regular income. If I made enough money just selling RC2014s, then great. As it happens, I've not yet taken up a job in Sainsbury's, and I'm still selling more RC2014s week-by-week.

Every couple of months or so, there's a new module coming out to either enhance what's already there, or add to the range. There's a nice little traffic-light board coming out soon, so you can test your programming skills by controlling some traffic lights. It was a classic type of a test project when I first learned computing, back in the 1980s.

THE VIRTUES OF THE Z80

It's a really good, solid chip. Until recently, NASA still used them in their satellites, because it was the most advanced chip that they could analyse down to the transistor level. Everything is tried and tested with the Z80. There are a lot of people who grew up with it, their first programming experience was with it – there's a lot of knowledge out there which helps too, as well as the fact that, for a lot of people, their first computer was their first love,

BUILD IT YOURSELF

If you want to follow in Spencer's footsteps and build your own Z80 on a breadboard, take a look at our tutorial on page 86. Alternatively, head to Spencer's Tindie store at hsmag.cc/xnkettv to get yourself an RC2014 kit.

and if you had a BBC as your first computer, you're probably not going to be a Z80 fan and vice versa. I'm not a 6502 fan because I set out with a Sinclair Spectrum – which has actually been proven to be the best computer in the world ever, by the way, just to slip that in.

Raw Z80 code will run on the RC2014. If you've got some code for, say, the Spectrum that relies on the Spectrum hardware, then that won't work – it just won't have the hardware to talk to. Although conversely, things that were written in BASIC back in the day almost certainly will run, because the RC2014 runs BASIC. The RC2014 Pro runs CP/M, and almost all of the CP/M software runs on it. If there's a hardware-dependent graphics card or something, that won't work, but as long as it can talk through the serial port, almost anything that's CP/M will run on the RC2014.

There's a software project for the Z88, which was Sir Clive Sinclair's portable computer after he left Sinclair. The Z88 is still thriving, and there's a software group that's sprung up around that. That can now export C code directly onto the RC2014, so there is quite an ecosystem there, and people writing their own software to do that.

ALMOST OPEN-SOURCE

I don't describe it as open-source, because I don't share everything. I don't share the Gerber files, so only I can make the PCBs; however, all of the schematics are open. I do encourage other people to make their own modules. All of the source code is as open as possible, so technically it's not completely open-source, but it's as open as it can be.

The community that has sprung up around the RC2014 has probably got about a dozen people making boards for the RC2014, so I'm now not the only person making RC2014 boards. There are now other people making boards designed for the RC2014 with either expansion boards or a different shape and size backplane, because not everyone wants that form factor. I'm more than happy for anyone to make their own board designed for the RC2014! □

Networking for makers

Maker meetups can seem intimidating, but they don't have to be



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

Networking. Does that word make you want to run away and hide in a corner? It used to have that effect on me, but, nevertheless, there are still times when a maker needs to meet new people.

I thought it implied, 'talk to people to find out how you can use them for something' – or more to the point 'be quizzed by people until they realise you are of no use to them'.

It always seemed insincere and rude. Those who are good at small talk make it look effortless, while I would often stumble and blush my way through what felt like hell.

But now I really appreciate the opportunity to get to know some people better. I still am uncomfortable starting a conversation with someone new. I still find 'peopling' exhausting. I still would rather not do it at all.

However, the reward of making friends with interesting new people completely outweighs all this.

Here are the two main tricks I use to help me get over that awkward initial meeting. These involve using Twitter, but other social media platforms would also work. I have made most of my 'Real Life' friends this way.

- When I find someone interesting on Twitter, I start to chat with them by responding to something they have tweeted. I take care not to overdo it or

spam them, and try to add something constructive to the conversation. I know that if someone replies to every one of my tweets, they are likely to soon get muted, so I respect that others may feel the same. Then, if we are at the same event or in a similar location, when we do meet in real life (IRL) it's not 'cold' – we already have some conversation history.

- At a conference, I see who else is tweeting by following the event hashtag. I then tweet to them, and arrange to meet during one of the breaks. Twitter is then an easy subject to start talking about. From their Twitter bio, I also (usually) have their name – so I don't have to remember that either!

Those who are good at small talk make it look effortless, while I would often stumble and blush my way through what felt like hell

had either contributed, or lurked, on #makershour (Wednesday evenings, 8pm UK time on Twitter). This meant most of us knew of each other. As Guild member, Sue Archer, said later, "The launch was brilliant. Just the right mix of social, speakers, 'networking', and fun. I say 'networking' because it seems most of us all really enjoyed the launch. Even those of us who don't always 'people' so easily."

I'm not a people person. But I do like to meet great people. Using Twitter, I have re-defined 'networking' in my head to 'making new friends'. □

The Donkey Kong Transistor

Silicon brains go under the knife



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.

A

paper from 2017 entitled ‘Could a Neuroscientist Understand a Microprocessor?’, by Eric Jonas and Konrad Paul Kording

[hsmag.cc/ivOQfi], is a brilliant example of the kinds of insights that come from interdisciplinary collaborations.

The paper applies neuroscience techniques to analyse Visual 6502 [hsmag.cc/CIQGCQ], a JavaScript-based transistor-level graphical simulation of a 6502 processor.

The latter is subject to a panel of neurobiological assays, such as connectomics, single-lesion studies, single-device tuning studies, local field potential analysis, dimensionality reduction, and

more, sometimes with insightful and other times humorous results. My favourite is the single-lesion study, where the authors studied three different 6502 ‘behaviours’: rendering the Donkey Kong, Pitfall, and Space Invaders video games, while individually removing single transistors and observing if the behaviour would succeed or fail. A biologist friend of mine felt the experimental setup was also typical of the studies often used in genetics, disease, and metabolic investigations.

The results of the experiment were fascinating: while lesioning roughly half the transistors resulted in a total failure to function and the other half seemed to have

no impact, a handful of transistors were found to be specific to each game. Certain transistors, when removed, would inhibit the Donkey Kong behaviour, but not impact the Pitfall or Space Invaders behaviour. This could lead a neurobiologist to hypothesise that perhaps there is a ‘Donkey Kong’ transistor: a single device responsible for the ‘memory’ of Donkey Kong.

Other studies from the paper were able to draw more insightful conclusions, such as the clock being two-phase, that the read/write signal is highly significant, that

the registers affect the accumulator, or that decoders affect the status bits. However, no study was quite able to thoroughly grasp the higher-level structure and organisation of a microprocessor.

Certain transistors, when removed, would inhibit the Donkey Kong behaviour, but not impact the Pitfall or Space Invaders behaviour

The paper’s thrust was to advance the evaluation and refinement of neurobiology methods and neuroinformatic tools. Perhaps well-characterised devices, like the 6502, could be *in silico* ‘model organisms’ for refining techniques prior to deployment in expensive *in vivo* experiments. My intuition tells me that there is much low-hanging fruit to be harvested from further collaborations between hardware reverse engineers and neurobiologists: the premises and techniques used to analyse black-box silicon or undocumented instruction sets might be adaptable to neuroinformatics, thus expanding the toolbox available for decoding the function of our own brains. □

Letters

ATTENTION ALL MAKERS!

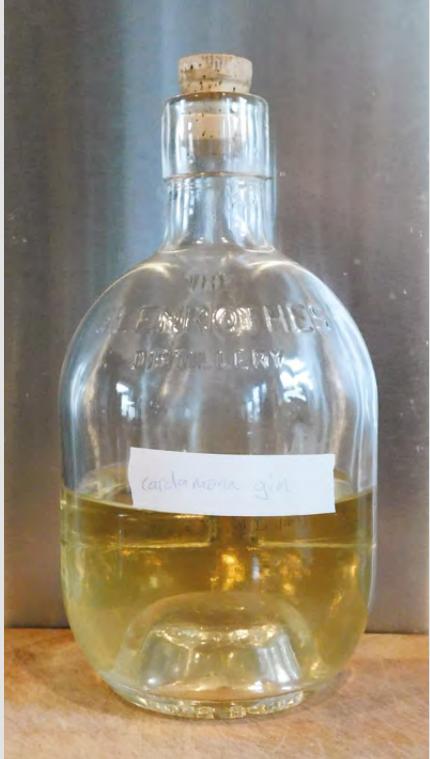
If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at
hsmag.cc/hello

GINNY PIG

I tried making gin after reading issue 6; it didn't work. So I went out and bought a bottle of Tanqueray instead, which very much did work. Thanks for putting the thought in my head!

Richard

Milton Keynes



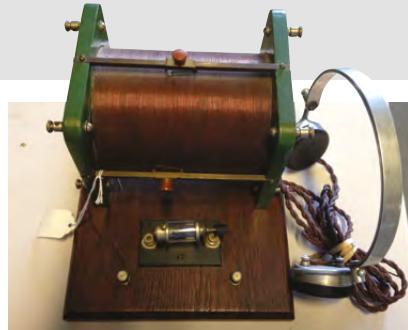
RADIO RADIO

How nice to see the humble radio get a full six pages of geekery devoted to it! It's not been cutting edge for a while, but there is something really satisfying about buying a few off-the-shelf components and recreating a technology that revolutionised the world not long ago. I think Mr Marconi would be pleased to learn that people are still tinkering with his creation all these years later.

James Turnbull

Brighton

Ben says Building a radio from scratch should be within the reach of most people reading this, but as devices get



more computer-based, we're losing that sense of immediacy you get when you know that you could build it yourself. That's why I'm impressed that two of the projects in this issue, the Arith-Matic S1-AU Mk1 on page 25 and Spencer Owen's Z80 on page 18, keep hold of that home-brewed ethos. We're still in a golden age!

METAL MACHINE MUSIC

Reading your interview with Alec Steele [HS06] reminded me of Zen and the Art of Motorcycle Maintenance. For those who haven't read it, it doesn't have a great deal to do with motorcycles; it's more about doing the best you can, finding pleasure in your work, and living in the moment. Alec's right about the magic of bending the world to suit you, rather than allowing your work to be shaped by the materials. Plus the YouTube channel is seriously, seriously impressive. Fire is, ironically, cool.

Rachel Fitzwilliam

New York





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

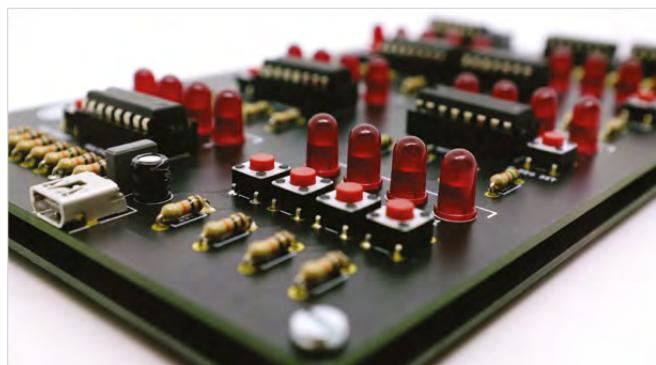
Arith-Matic S1-AU Mk1

A DIY adder to build at home

From £44 | arith-matic.com | Delivery: 2018

How does a computer work? That's the question that Arith-Matic are aiming to answer. They're launching a beautifully crafted 4-bit adder board built out of 7400 series logic chips. Four input switches allow you to load data into two registers, and display the result. This result can then be loaded back into one of the registers, allowing rudimentary multiplication. All the processing takes place in 7400 series chips (flip-flops, multiplexers, adders, logic gates, etc.) – there are no programmable parts on the board. Don't be fooled by the USB cable – there's no data uploading, it just provides power.

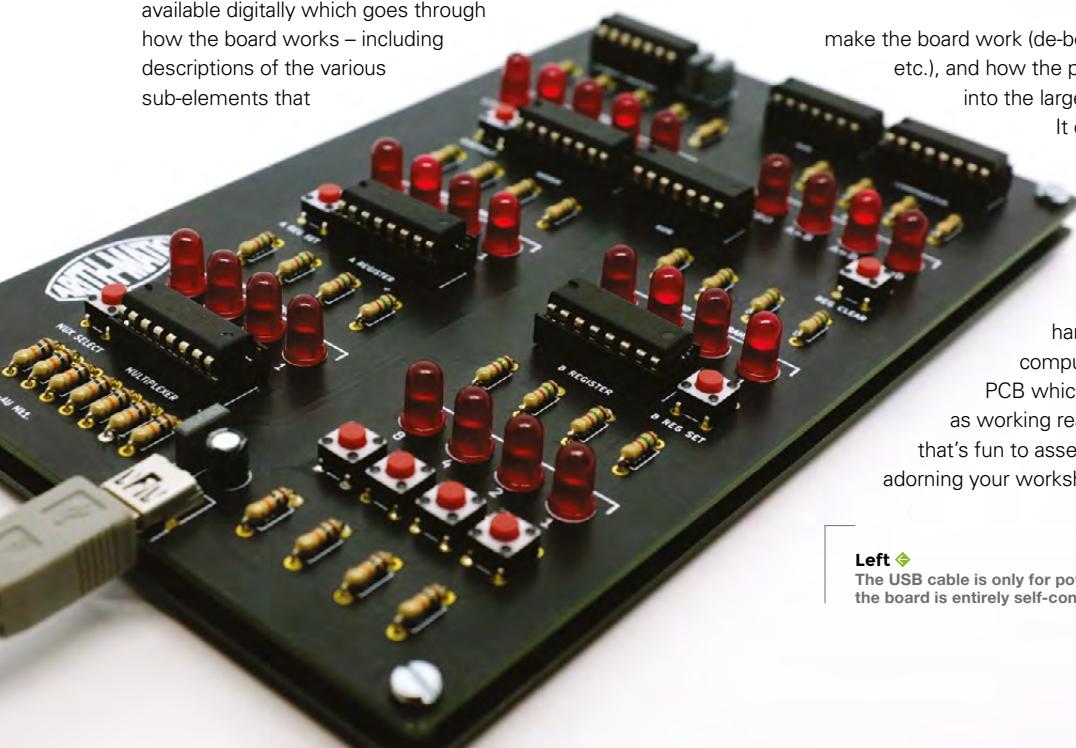
The board comes in kit form, so you will need to know how to solder it together, but that's the only knowledge you need to use it. There's a comprehensive manual available digitally which goes through how the board works – including descriptions of the various sub-elements that



Above Push-buttons get data into the board and LEDs display the results of the computation

make the board work (de-bouncing circuits, current-limiting resistors, etc.), and how the processes demonstrated in the board fit into the larger picture of computing.

It could be a good tool to help children understand what goes on inside computers, but it is more than this. It's the combination of form and function which particularly impresses us. The well-spaced through-hole components with large (5 mm) LEDs hark back to the older days of homebrew computing. They look great on a slate-grey PCB which is designed for its aesthetics as well as working really well. It's the sort of circuit board that's fun to assemble and play with, but also looks great adorning your workshop. □



Left The USB cable is only for power – the board is entirely self-contained

Space of the month: Urban Hax



Urban Hax

Urbanhax.com
[@UrbnHax](https://twitter.com/UrbnHax)

W

e spoke to Geoff Henderson about Urban Hax, a community interest company (CIC) makerspace in the West Midlands.

HOW DID YOU GET STARTED?

I hate rules. It was the same when Fab Labs came out. You had to buy this equipment, and you had to portray the Fab Lab in this particular way. About five or six years ago we were going to set up a – it wasn't called a makerspace then – we were going to set up a Fab Lab, and you had to spend £100k on equipment. So you had to have a shop-bought router, you had to have a laser cutter and so on, before you could use the Fab Lab brand. The idea, the principle of the fabrication lab is great, but it didn't suit our direction of travel (and we didn't have £100k to spend, which was kind of a stumbling block). So we started small instead.

WHEREABOUTS ARE YOU BASED?

Walsall, West Midlands. We're in what used to be an old corn mill – a Victorian mill that was an old corn mill. In fact, it's one of the few remaining Victorian mills in Walsall, an old industrial building. We still have the stable block, and in our stable block are the maker bays. One stable = one maker.

Mark, who's one of our residents, he's a guitar maker, a luthier, he rents a stable. We have Carl who's a carpenter, he rents a stable. Upstairs where you have a cleaner environment, makers rent a desk. We have a jeweller, a fashion designer, a 3D modeller, we've got a web designer, and the whole idea of our model is to build a creative community where they can get on and do their own thing, or if they so choose they can talk to their fellow creators and come up with new ideas.

WHAT SORT OF EQUIPMENT ARE YOU WORKING WITH?

Upstairs we've got the photography studio, backdrops, cameras of course, we've got a bank of 3D printers, high-resolution printer, high-resolution desktop laser scanner, large-format colour printer, high-spec laptops and 24-inch monitors so people can do CAD design

and graphic design. And there are sewing machines and so on, and it just builds and builds and builds. If the residents make out a case for the next piece of equipment, we will prioritise their wishes.

Walsall's a leather town; there are a lot of leather companies in Walsall, and we made contact with one. I asked them if they did leather offcuts, and they rang me the other day with an offer of maybe two or three thousand pounds' worth of high-quality stuff with minimal defects. That's a resource now for the jewellery makers, for the fashion designers. I want to go on a course now to learn how to work with the stuff.

WHAT SORT OF THINGS ARE YOUR RESIDENTS WORKING ON?

Mark, being a luthier, he's obviously working on guitars, and right now he's working on a Telecaster-style custom model for a customer, which he's building out of swamp ash – I never knew what swamp ash was. We have Carl, who's a traditional carpenter. He's now upcycling musical instruments to make lamps. We're working with a guy called Lee, who's looking to build a car to break the British land speed record by dropping a couple of jet engines into a car, so we're doing all the prototype modelling of the car, the computational fluid dynamics to analyse the airflow. We're working on a hydroponic system for growing plants indoors with someone setting up the first zero-waste supermarket in Birmingham. He likes the idea of hydroponics for herbs, so customers will be able to come in and just pick their own fresh herbs grown in the store.

We're designing a flatpack loom; we've got a passionate textiles group and that's a multinational group of Iranian, German, and English makers working together on fashion and textile ideas. They're working on the idea of a flatpack loom, using traditional woodcraft but also designing some of the parts using a 3D printer – it's a real mix, and we think that's the way it should be.

We took our inspiration from MIT and Fab Lab. There's one in Massachusetts called Artisan's Asylum, which started off as a small makerspace. They now have 40000 square feet. Welding bays, woodworking areas, spray booths, and you rent space and storage there. That's where we want to get to. ➔



Above

The West Midlands has a long history of jewellery making, and this lives on in Urban Hax

Space of the month

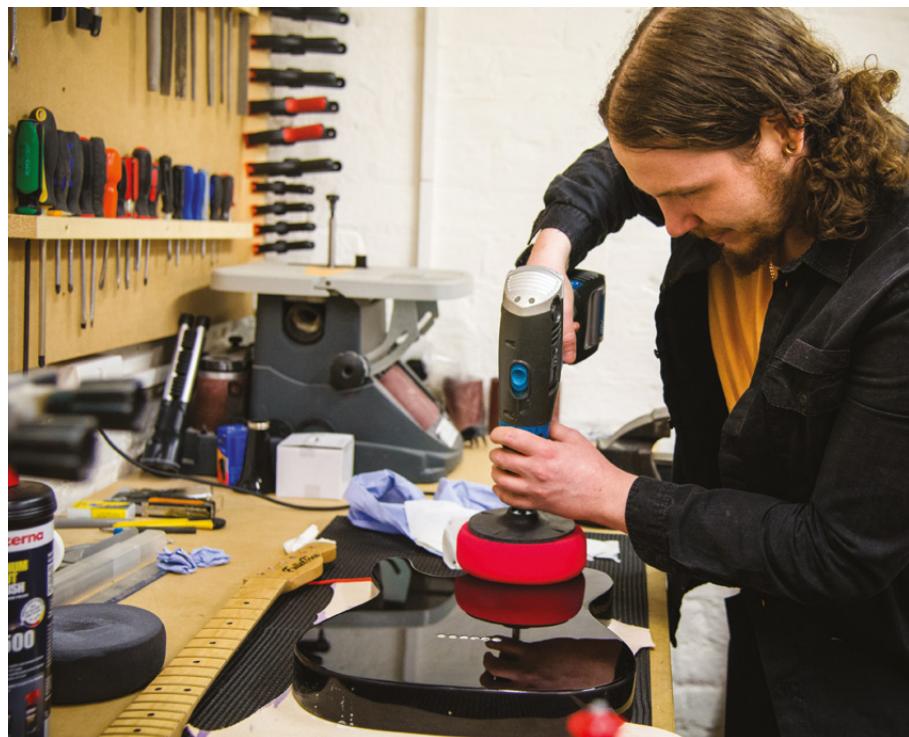
REGULAR



Above ♦

Photography is a key part of the professional maker experience. Better images lead to more exposure and more sales





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.

Below ■
The Black Country was once known as the Workshop of the World, and still has a lot to offer the global maker scene



WIN!

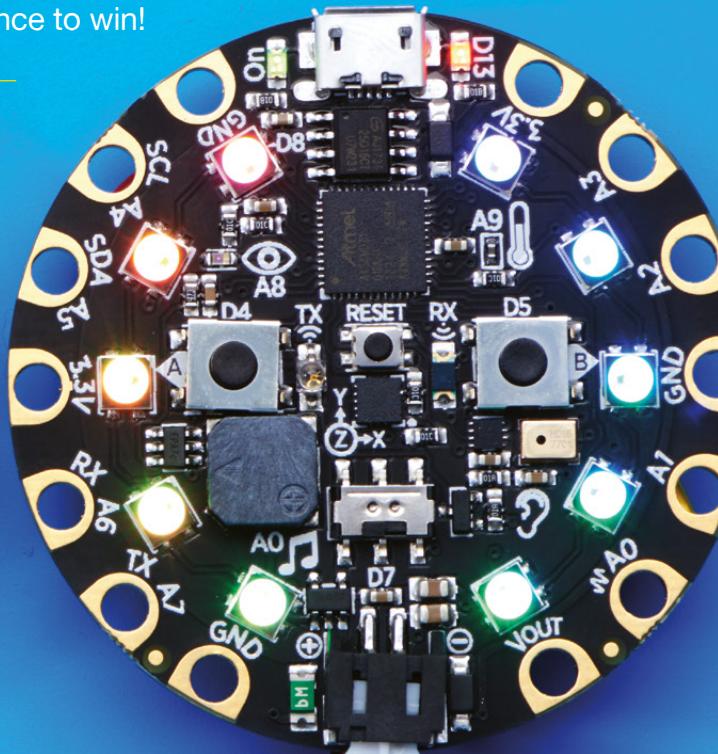
1 of 10
Adafruit Circuit
Playground Expresses

WORTH
£25

signed by Limor 'LadyAda' Fried

Head to **hsmag.cc/win**

Enter by 30 June
for your chance to win!



Terms & Conditions

Competition opens on 21 May 2018 and closes on 30 June 2018. 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.



LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG
48



ELECTRIC LEGOLAND

How one reader hacked his Lego set to include WiFi lighting

PG
52

DIGITAL BLACKSMITHS

3D printing from waste materials is bringing modern making to Africa

PG
58

MORE MEMBERS

Bring in people from a wider range of backgrounds and grow your hackspace

PG
32

INTERNET OF EVERYTHING

Network your makes and build them into a hive mind

PG
64



INTERVIEW HELEN STEER

Everyone needs to create their own light-up unicorn

PG
72

PENCIL ERASERS

They're everywhere, but surprisingly hard to hack with

INTERNET OF THINGS

Everything you ever wanted to know about building your own IoT device

You are in your favourite cafe, engrossed in the latest issue of HackSpace. You reach for your cup of coffee by impulse, when suddenly your wristband vibrates to let you know that the coffee is still too hot for you. You get back to the magazine and about a minute later there's a notification on your watch telling you that the beverage is at the temperature you like. That's the kind of interconnectedness that's possible thanks to the Internet of Things, popularly known as IoT. While you can't buy this smart cup (which knows how hot you like your coffee from the NFC in your smart wristband) off the shelf just yet, innovative makers are building all kinds of IoT devices to change the way we live. For you, IoT might be a security system that monitors your house for any intrusions

and alerts you, and the security agencies, when it senses a break-in. Or, a wristband with the ability to monitor the vitals of your parents, which alerts you when it finds anything out of the ordinary.

Think of the IoT as a network of connected devices. It can include everything from phones and wearable devices to everyday appliances like washing machines and coffee makers and home automation systems. What makes them smart is that they are connected to a network like the internet, and are aware of other connected devices around them. But IoT is much more than things connected to the internet. IoT is about making dumb things smarter by giving them the ability to sense, communicate, and respond.

The internet is just an efficient means for these smart devices to communicate with us and each other. We respond to our environment based on the inputs received through our five senses. Similarly, if you add sensors to devices, they can do the same too. While you are at it, throw in a bunch of actuators

NGS

as well to get these IoT devices to do something like open and lock doors, switch on the lights, or turn up the heater, and you've got a smart device that can respond to its environment based on sensory input.

BUILDING THE FUTURE

Not surprisingly, it is the maker movement that's fuelling the growth of IoT. Modern-day makers have the same ethos as the homebrew computing club hobbyists of the 1970s. They have a knack for solving problems creatively, and the ability to work around limitations is woven into their DNA. The proliferation of wireless connectivity and cloud computing together with low-cost microcontrollers, as well as the ubiquity of advanced sensors, enables makers to prototype all kinds of IoT devices at a very low cost.

There's no doubt that the rapid growth of IoT is due to the explosion in the availability of small, inexpensive computing hardware. There's a staggering number of IoT prototyping kits and development boards that combine microcontrollers and processors with wireless chips and other components in a pre-assembled, ready-to-program package, and we'll look at some of the most popular

ones in this feature. Whatever the needs of your project, there's sure to be a board that meets your requirements. Similarly on the software side, we'll look at some development tools that'll help you quickly program your project into existence. The feature will also showcase some of the interesting IoT devices that makers have built using some of this hardware and software.

In 2017, for the first time, the total number of connected devices outnumbered the world's population (according to Gartner), and they estimate that this number will more than double in the next couple of years. Smart consumer devices will continue to be the main driver, and account for a majority of the IoT peripherals. A quick gaze into our crystal ball tells us that the maker movement will continue to push IoT devices from their garages to the shelves of a store near you. That's because online fundraising platforms, like Kickstarter and Indiegogo, have helped democratise entrepreneurship, and in fact some of the most popular IoT devices and smart wearables have been crowdfunded using these platforms. Combined with the influx of cheap yet powerful hardware, this trend is only going to get much stronger. →

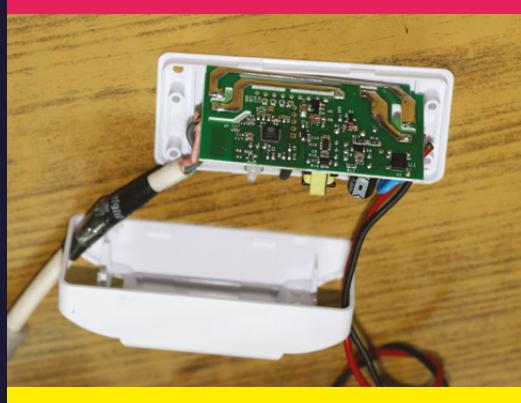
DEVELOPMENT BOARDS

Kick-start your IoT project by selecting the right hardware

If we had to pick one reason for the growth of DIY IoT projects, it has to be the availability of small, inexpensive development boards. They combine microcontrollers, processors, wireless chips, and other components, and come in various configurations – from tiny battery-powered chips to credit card-sized single-board computers. Here we've rounded up ten of the most popular boards that you can use to get your IoT project off the ground.

ESP8266

This board caught the fancy of makers from the moment it was launched in 2014. The tiny 24.75 mm × 14.5 mm board is essentially a WiFi module that allows microcontrollers to connect to a wireless network. The reason for its popularity is that it is far cheaper than other WiFi devices, which is why you'll find it in a wide range of other development boards and IoT kits. There are several variants or versions of the module; some have external antennas while others feature multiple GPIO ports. If you are new to IoT and making, you should avoid directly using this board since it takes some effort to safely power and program it.



Right ♦
Pry open an IoT device and there's a good chance you'll find an ESP8266 inside it, such as in this Sonoff WiFi switch

Arduino MKR1000

The MKR1000 (pronounced 'Maker'), is based on the Atmel ATSAMW25 SoC that combines a WiFi shield with an Arduino Zero-like 32-bit ARM microcontroller, and also includes an encryption chip. It has a LiPo charging circuit and battery management that allows it to run either with batteries or external 5 V power that can be supplied via the built-in USB port. The board measures 61.5 mm × 25 mm, which makes it ideal for energy-efficient, remote-powered projects needing WiFi capabilities in a compact form factor. You can program it using the Arduino IDE.

Particle Photon

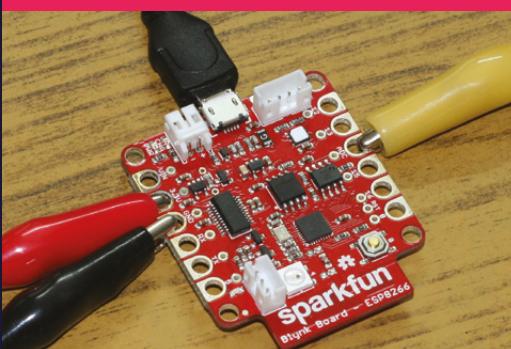
One of the smallest prototyping boards on the market, the Photon is powered by a 120MHz ARM Cortex M3 microcontroller, and includes a Broadcom WiFi chip in a tiny thumbnail-sized module called P-zero. Photon has 1MB flash, 128kB RAM, and an on-board RGB status LED. It has numerous pins for connecting sensors and actuators, and is available in two physical forms – with and without headers. You can power it via the on-board micro USB connector or via the VIN pin. Photon plugs directly into standard breadboards, and can also be mounted on a PCB. You can program it using Particle web or local IDE, and use ParticleJS to build web and mobile apps.

SparkFun ESP8266 Thing

This is a simple board that adds some extra features to the ESP8266 WiFi module. SparkFun has named it Thing because it's intended to form a perfect foundation for IoT projects. The board has breakout pins in two parallel breadboard-compatible rows. There are also some LEDs to indicate power, charge, and status of the IC. The ESP8266 Thing also has a battery charger circuit and a JST connector for plugging in a LiPo battery. It's also got an on/off switch and a USB port, and you can program it in the Arduino IDE by installing the ESP8266 Arduino add-on.

SparkFun Blynk Board

If you are new to IoT, this board will help you get started in no time. It's based on the ESP8266 framework, and has a dozen crocodile-clip-compatible pin connectors to ease first-timers into making. The Blynk board also has an on-board humidity sensor and can be powered either via LiPo battery or micro USB. You can program the board without writing a single line of code via the Blynk app that's available for both iOS and Android. The app has several widgets that are designed specifically for the Blynk board.



SparkFun ESP32 Thing

When Espressif released the supercharged update to the ESP8266, called ESP32, SparkFun gave it the same treatment as the older module. This Thing has all the goodies of the ESP32, which uses a dual-core microcontroller along with 128kB of ROM and 416kB SRAM, and includes Bluetooth Low Energy (BLE) and about 30 new GPIO pins. The ESP32 Thing has an FTDI USB-to-serial bridge for programming the microcontroller, along with an integrated LiPo battery charger. SparkFun has again broken out the I/O pins into breadboard-friendly headers and you can still program the EST32 Thing using the Arduino IDE.

Adafruit Feather

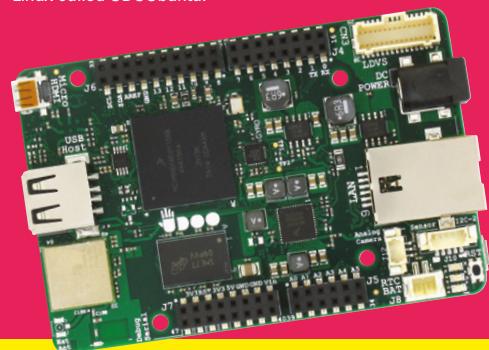
The Adafruit Feather is a range of standalone and stackable boards that are tiny and portable. You can power them with LiPo batteries or micro USB. The Feather line includes basic boards, data loggers, HUZZAH ESP8266, HUZZAH ESP32, and Arduino-compatible boards. You can also extend the capabilities of the boards by stacking accessories, dubbed FeatherWings, for functions such as real-time clocks, DC motors and steppers, GPS, and more. All the Feather boards can be programmed with Arduino IDE, and some can even be controlled by CircuitPython, which is a Python variant designed to help beginners program IoT boards.

Raspberry Pi 3B+

The Raspberry Pi is perhaps the best-known of all the IoT development boards. The third generation of the credit card-sized single-board computer (SBC) is based on a Broadcom BCM2837 SoC, with a 1.4GHz 64-bit quad-core ARM Cortex-A53 processor and 1GB RAM. It's also equipped with Bluetooth 4.2, WiFi, an Ethernet port, an HDMI port, four USB ports, and a microSD card slot. It features 40 GPIO pins and you can hook up third-party breakout boards. The board runs a customised version of Debian called Raspbian, on which you can install a variety of development packages like the LAMP stack, Node.js, Python, and more.

UDOO Neo

This crowdfunded SBC is often pitched as the combination of the Linux and Arduino. It has two cores on the same processor – a 1GHz ARM Cortex-A9, which can run Android, and an ARM Cortex-M4 for Arduino. The board has the same pinouts as the Arduino Uno and includes a WiFi and Bluetooth 4.0 module. Adding to its USP are the nine axis motion sensors. There's also a USB port, micro HDMI, and a micro USB that you can use to power the device in addition to the DC plug. You can program it with the Arduino IDE or in any other language since it can run a customised version of Linux called UDOObuntu.



BeagleBone Blue

BeagleBoard makes several credit card-sized SBCs, and the BeagleBone Blue is a wonderful choice for IoT and robotics projects. It features a 1GHz ARM Cortex-A8 processor with 512MB of DDR3, 4GB flash storage, WiFi, and Bluetooth Low Energy. You can power it with a two-cell LiPo battery. For makers, the board has two user-configurable buttons, eleven user-configurable LED indicators, eight 6V servo out and four DC motor out, along with a nine-axis IMU and barometer and an assortment of GPIO and serial protocol connectors. The board is compatible with Debian, ROS, and ArduPilot software, and the LabVIEW and Cloud9 IDEs. →

Left ↲
The Blynk Board comes with a guide to help first-time makers build a dozen projects without writing a single line of code

Below ↴
Besides UDOObuntu, you can also download a ported version of Android Marshmallow for UDOO Neo from its website

DEVELOPER TOOLS

Make your microcontrollers do your bidding with these developer tools

Once you've zeroed in on a development board for your IoT project, the next challenge is to find the right tool to program your project into existence. There are a whole gamut of open-source tools at your disposal and we've rounded up some of the best that'll help you get off the ground in no time.

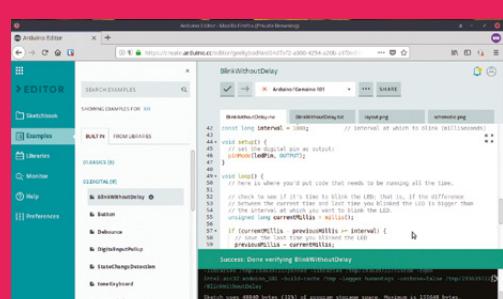
Arduino Create

Arduino Create is an all-in-one online platform for makers to write code, configure boards, and share projects. The platform allows you to browse a collection of projects that you can then use for your own projects and write code using the web editor. When you're done, you can share your builds, along with step-by-step guides, schematics, and references. The web IDE lets you write code and save it to the cloud, from where you can access it from any device. When you visit the web service for the first time, you'll be asked to create an account and then install the Arduino Web Editor plug-in to your browser. The plug-in is available for all major web browsers across Windows, macOS, and Linux. With the plug-in installed, the web IDE will recognise your Arduino and Genuino boards connected to the PC and allow you to configure them as well. You'll also be able to upload sketches from the web editor to your board via USB, using the serial monitor that's part of the IDE. Although the Arduino web IDE is intuitive, there's no dearth of textual and video documentation to hand-hold you through its features. There's also the Arduino Cloud platform (reviewed last issue) that includes a service for passing data backwards and forwards from your device to other internet-connected services.

Right ➔
Arduino Create works well with the official Arduino/Genuino boards, but if you have others you'll be better served by Arduino IDE

PlatformIO

While the availability of different microcontroller and development boards is a good thing, programming multiple ones can be quite a hassle since a majority have their own development tools. If you are working with different boards, PlatformIO will help you streamline the development process. It supports over 400 boards and more than ten frameworks, including Arduino, ESP8266 SDK, mbed, WiringPi, and Simba. PlatformIO is a cross-platform development environment for building IoT devices that includes an IDE, a build system, a serial port monitor, and a library manager. To use PlatformIO you'll have to add it as a plug-in or extension to popular IDEs such as Atom, VSCode, Sublime Text, CodeBlocks, Emacs, vim, and more. There are several editions of the editor, but the freely available Community version should work for most non-enterprise users. When you fire up PlatformIO from inside your favourite IDE, you'll have to first select a target board and a directory for the project, inside which it'll download all the required libraries you'll need. Since PlatformIO runs inside the IDE you are already familiar with, it shouldn't throw any unexpected surprises. Still, new users are advised to refer to the project's extensive documentation to familiarise themselves with its nuances.

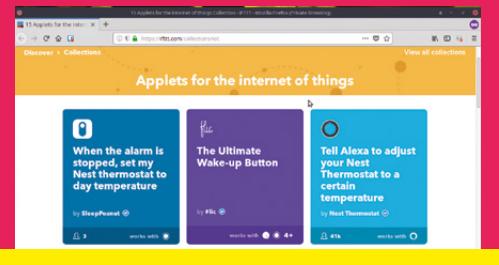


Node-RED

Node-RED uses a visual programming approach to allow developers to connect blocks of code, known as nodes, together to perform a task. Node-RED was developed by IBM and it allows developers to connect devices, services, and APIs together using a browser-based flow editor. You still need to write code, but the amount of code required is significantly reduced. A Node-RED flow is made up of nodes, each of which encapsulates a reusable piece of code that performs a certain task. The various nodes offer different functionality, such as the Raspberry Pi node which allows you to read and write to the GPIO pins of the Pi. Besides desktops, Node-RED can run on the Raspberry Pi and BeagleBone Black. Node-RED runs a Node.js-based web service that presents you with a graphical user interface to control data flow between modules. Based on JavaScript, modules can comprise all I/O and processing steps. Node-RED comes with a large library of complementary modules, for interacting with everything from the Raspberry Pi's GPIO to social media. The project has ample documentation to help you get started.

IFTTT

The popular IFTTT tool, which hooks several web services together, has been helping users automate mundane tasks since 2010. You can combine the relative simplicity of the web service with your IoT devices to weave together a smart home. Using IFTTT, which stands for 'If This, Then That', you can connect and control a host of internet-enabled devices. The service supports a wide array of smart devices including Belkin Wemo, Philips Hue, and Nest thermostat. The Maker service (now called webhooks) allows you to connect IFTTT to your personal IoT projects. You can use it to connect an applet to any device or service that can make or receive a web request. So for example, you can use IFTTT to turn on the smart lamps automatically when you get home, or use Google Assistant to water your plants. IFTTT supports hundreds of services that you can use as triggers. In addition to a trigger, you'll need to define a corresponding action. Together, the trigger and action constitute an event, or an applet as they are known in IFTTT parlance. You can create applets from a web browser or using the IFTTT app for iOS and Android.



Adafruit IO

Adafruit IO isn't unlike the other development platforms we've introduced. It is, in fact, a web service that logs data from your IoT devices and visualises them. Although you can use the service via a graphical interface, you'll still need basic programming to work with Adafruit IO. Adafruit provides client libraries to work with several programming languages, including Ruby, Python, and Node.js. To log data from your IoT to the service, you'll first have to sign up for a free Adafruit account. You can send data from your IoT device via the MQTT messaging protocol to the Adafruit IO service using your unique AIO key. The service will log raw data that'll show up in the feeds page. You can then use Adafruit IO to build a custom dashboard to visualise these values. The service supports all kinds of visualisations such as a gauge and a line chart. These are called blocks, and can be added as a component on a dashboard to show the data. In addition to logging data, you can control your IoT devices from Adafruit IO.



Eclipse IoT

Rather than being a standalone tool, Eclipse IoT is a collection of over two dozen open-source tools that can be used for developing IoT projects. Over the years, the Eclipse project has grown into a language-agnostic foundation that works across a wide range of software projects. The Eclipse IoT Working Group co-ordinates the efforts of various individual and corporate developers that work across several companies and share an interest in IoT. One of the major Eclipse IoT projects is Eclipse Kura, which is an easy-to-use gateway system that acts between sensor devices and a back end. It enables remote management of these gateways and provides a wide range of APIs that allow developers to write and deploy their own IoT apps. Kura runs on top of the Java Virtual Machine (JVM) and leverages OSGi, which is a dynamic component system for Java, to simplify the process of writing reusable software building blocks. There's also Mosquitto, which is a message broker for the MQTT protocol, and Eclipse Paho which is a set of libraries for different programming languages. It also hosts frameworks, such as Eclipse SmartHome that works on building solutions for the smart home. →

Left ↳
You'll need to install the Adafruit_MQTT library before sending data from your devices to the adafruit.io service

Left ↳
You can use webhooks to control custom IoT devices using IFTTT

SMART DEVICES

Makers are wiring smartness into all kinds of appliances

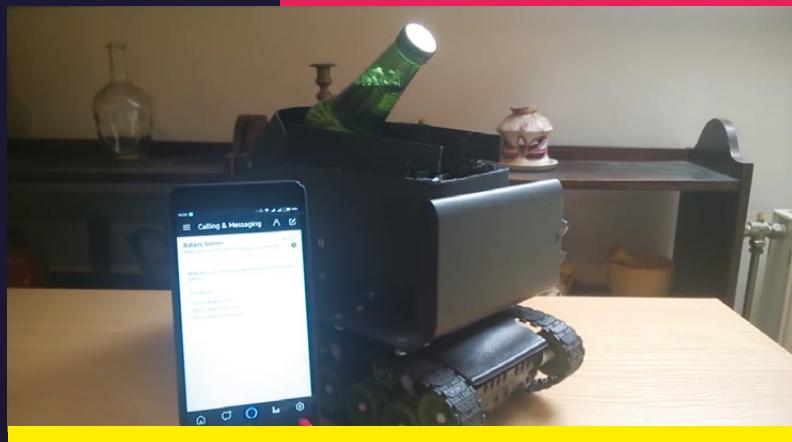
C

onnecting everyday appliances to the internet is perhaps the first step towards a smart home. But when we say smart, we aren't talking smart-TV-smart – devices that just connect to the internet to fetch content.

Instead, we are talking *Jetsons*-smart. Devices that use the web to function efficiently and outgrow their design goals.

Khairul Alam is a serial maker who loves to build IoT projects, several of which have won contests on Instructables over the years. He's used the ESP8266 to build a pill bottle (hsmag.cc/ERiWrc) that keeps track of the pills it holds, and also sends messages to remind users to take their medication on time. The device fits inside the cap of the bottle and is powered by a coin battery that can last for several months. Khairul uses the commercial Ubidots platform to turn the data from the device into meaningful information. Users can feed in their medication schedule and track the status of the bottle via an Android app.

Below ■
No IoT-enabled smart home is complete without this voice-controlled, beer serving, autonomous, follow-me-around tank



Khairul has also built a smart pantry (hsmag.cc/KceBXV) that automatically replenishes stock in your kitchen using Amazon's Dash Replenishment Service (DRS). He's used a force-sensitive resistor to measure the weight of his customised pantry shelf; this is passed on to an Arduino UNO, which interfaces with the DRS.

Stephen Harrison had other worries besides keeping a well-stocked kitchen. His Roll Pole project (hsmag.cc/ajFdjD) uses a Particle Photon to build a smart toilet-paper holder that ensures you don't find yourself in an embarrassing situation. The custom holder uses reflective infrared sensors to detect the number of rolls. If the number falls too low, IFTTT adds toilet rolls to your shopping cart.

Talking of toilets, Igor Fonseca has made an IoT-ised air freshener (hsmag.cc/mvYqBH) that can spray perfume, based on a timer, or upon receiving an email, or whenever you enter your home, or manually by pressing a button. Igor 3D-printed a case for his NodeMCU LoLin-powered air purifier, and used IFTTT to create triggers for the different actions. In every case, the service calls upon Adafruit IO to log the commands. It'll also communicate with the device to move a servo motor and release the perfume.

And how can you have a smart home without a smart butler? We're stretching the definitions of IoT a bit here, but it'd be a shame not to include Balázs Simon's autonomous beer-serving tank (hsmag.cc/fbYkdp). Balázs has used an Arduino MKR1000 and a Raspberry Pi to control the tank, using a Walabot to navigate. You can operate the tank in the Follow Me mode, where it just follows you around, serving you beer when you ask it to via Alexa. Checkout the demo videos on Balázs's project page for inspiration on easy beer service.

INTELLIGENT INTERFACES

With your IoT appliances in place, it's time to smart-up the interfaces

An interface in computing is the boundary between the different components. When we talk about interfaces in the IoT-enabled smart home, we mean devices on the fringes, which includes things like doors and mailboxes.

Frank Burke-Olson gave his old door buzzer the IoT treatment by fusing it with a Raspberry Pi Zero W (hsmag.cc/axdpLE). Whenever somebody presses the buzzer, an Android phone in his living room snaps a picture that's emailed to Frank, who then uses the Pi to signal to a relay circuit to unlock the door. The system works when he is connected to his home WiFi, so Frank used a port forwarding service so that he is able to buzz someone in from across the planet.

Members of the Windows IoT team used a Raspberry Pi to create a Facial Recognition Door (hsmag.cc/GeqaKI) that unlocks itself when it recognises the person requesting entry. The project uses a basic webcam, and the door does facial recognition using a Microsoft Cognitive Services API running in the Microsoft cloud.

The makers at Naran Inc have a different approach to the problem of letting in visitors. Instead of building facial recognition into their door, they have made a smart doorbell (hsmag.cc/hRzvsS). When a visitor knocks at an earmarked spot on the door, they are picked up by a sound sensor that activates and snaps a picture of the visitor. The picture is then sent to the owner via the Telegram messaging service, who can then ask Alexa to remotely open the door using MicroBot Push, Naran's wireless robotic button-pusher. All the logic in the project is controlled via IFTTT and Prota OS, which is a home automation platform for the Raspberry Pi.

Now from opening doors to locking them. Dave Astolfo has lots of stuff in his garage that he sometimes forgets to lock. So he used an Arduino Nano to monitor the garage doors (hsmag.cc/dVUtHb) and can lock them remotely using the Blynk app. Dave's setup also involves a fingerprint scanner, mounted outside the garage door, to authenticate and open the doors. In addition to the build instructions, Dave shares the issues he faced when implementing the project in the real world, as well as a couple of useful tips.

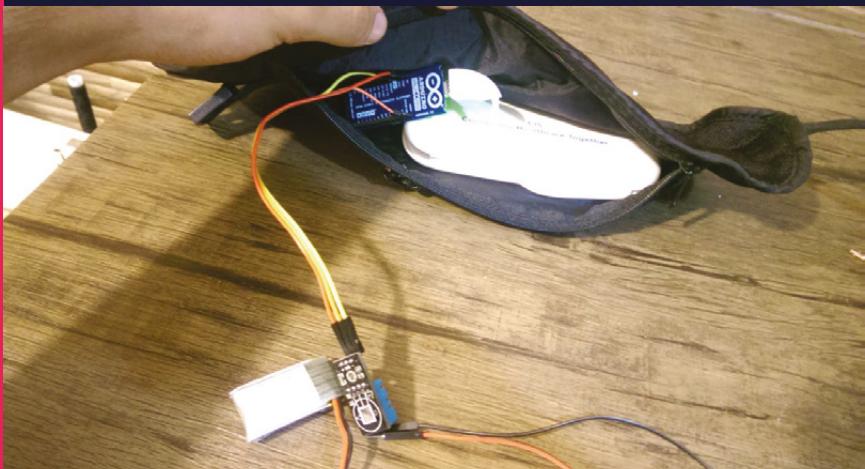
Talking of notifications, John McAlpine decided to build a smart mailbox (hsmag.cc/cpENTx) in order to receive notifications whenever he received snail mail. He used a Particle Photon because it could be put to sleep when not in use. John used a reed switch to detect when the mailbox is opened, to wake up the Photon, and send an alert using IFTTT. If you are an amateur in electronics, let Mike Diamond hand-hold you through the process of making a smart mailbox (hsmag.cc/fEEbof). Mike pairs a reed switch with an ESP8266, and has interesting insights into powering the project, which is useful for anyone interested in making ESP8266-based remote-powered projects. →



Above ↑
The components of the original smart mailbox were mounted on a Perfboard, but one of Mike's friends has since fabricated a custom circuit board for him

MONITOR AND TRACK

Use IoT to become omnipresent



Above Mahdi's temperature monitor fits inside a bumbag, and all you need to do is just attach the DHT sensor with medical tape

If convenience is the primary advantage of IoT, then peace of mind has to be a close second. An indirect benefit of internet-enabling your possessions is that you can keep an eye on them from anywhere in the world.

That's all the excuse Kelsey Breseman wanted to strap a Tessel 2 on to her Pomeranian (hsmag.cc/TccOqB), along with an accelerometer to track how long it slept when Kelsey wasn't around. She used Twilio to send SMS alerts whenever the accelerometer values changed after being the same for five minutes.

Staying with the theme of pets, the CircuitIO team made a Pet Feeder (hsmag.cc/rQYiQV) using an Arduino Uno to dispense treats in two modes. In one mode, the feeder lets out a beep and triggers the servo to dish out some food only when the PIR motion sensor can see the dog approaching. In the other mode, the servo can also be triggered remotely from a smartphone.

Scott Bennett doesn't have a pet to track – instead he uses an Arduino Uno to track his bike (hsmag.cc/zZOZAf). The whole system fits inside a bag that is hidden under the saddle. The project uses an Arduino GSM shield, together with an Adafruit GPS breakout, to send GPS co-ordinates of the bike via text messages that you can plot on Google Maps.

Instead of tracking individual devices, Naran Inc uses a Raspberry Pi to keep an eye on their house (hsmag.cc/PMULza) while its residents are away on a vacation. It's a rather straightforward project that repurposes a webcam to take on-demand snapshots and also warn you when it detects and captures motion. It uses a sound sensor to detect any unusual sounds. There's also an old smartphone that live-streams video and is pointed at the entrance door, so it can play an alert message to warn the neighbours in case of a break-in. Naran's system also uses the MicroBot Push to remotely switch on the HVAC to come home to a warm house during the winter, or a chilled one during summer. The entire thing is powered by their Prota OS distribution and makes ample use of IFTTT to define the different tasks.

On a more personal level, arguably one of the best uses of IoT is in the field of medicine. Mahdi Hajebi has innovatively merged microcontrollers with medical instruments to make smart medical devices. The first device uses two Arduino MKR1000s to monitor and control a patient's temperature (hsmag.cc/BjRqGh). One Arduino is part of a wearable that uses a temperature and humidity sensor and sends them to Samsung's Artik cloud. When the patient's temperature gets too high, the device alerts the doctor and signals the second Arduino to switch on a cooler.

HOME AUTOMATION

Use IoT to give your house an upgrade

Once you're done turning your everyday appliances into smart ones, it's time to give your digs the same treatment.

When Giovanni Gentile had to install new lights in one of his rooms, he decided to take the IoT route and make the lights (hsmag.cc/QkJFrB) reflect his mood. Giovanni's project involved lining strips of RGB LEDs behind a big wardrobe in the room, which were hooked to an Adafruit PCA9685 and a Raspberry Pi Zero. He uses the Cayenne app to manage the setup from his phone.

Adithya TG wanted something more comprehensive. He uses an Arduino Uno, a couple of ESP8266s, a LinkIt Smart 7688 Duo, and other devices to control and monitor his house (hsmag.cc/rUOday). The system is voice-controlled via Alexa. Adithya can keep tabs on the temperature and humidity inside the house, control lights and fans, and check on the status of these devices. The system also sends notifications when the temperature rises above a particular threshold (such as in case of fire). He also gets a notification when someone rings the doorbell, along with the picture of the person at the door. Another webcam takes on-demand snapshots of the living room.

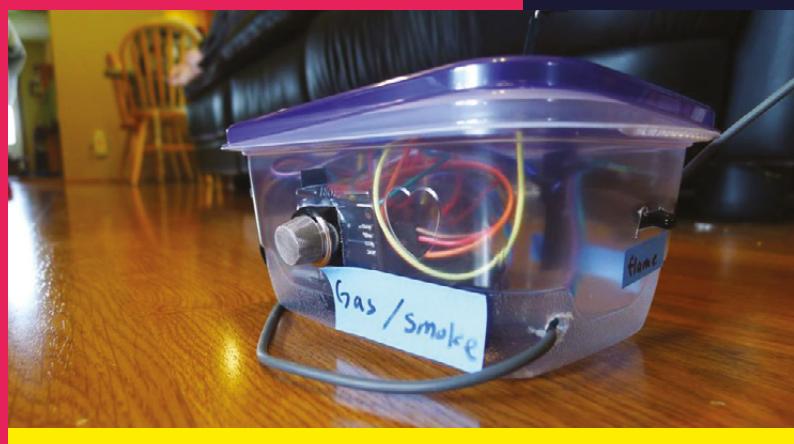
David Liberda has a very presentable take on home automation. His project, dubbed Walle, (hsmag.cc/IWqdyYP) uses a 15.4" laptop screen and a capacitive touchscreen that you can use to control the lights and heating, and doubles up as a smart calendar. You can also talk to Walle via Alexa. A Raspberry Pi 3 controls the touchscreen, while two SparkFun Arduino Pro Mini 328s are in charge of the lights and heating.

If you want something cooler than a voice to control your IoT devices, take a look at Tamas

Imets's project that uses Jedi Force gestures to control lights and his TV (hsmag.cc/OYBbFV). Tamas has created a smart plug, powered by a Particle Photon, that controls any device plugged into the circuit. His system uses the values of the accelerometers in a Pebble smartwatch and translates them into actions. In the video, Tamas demonstrates using the Force pull gesture to switch a lamp on and off, a Force push to turn on the TV, and he can flick through the channels with a Jedi mind trick gesture.

For a comprehensive solution to IoT-enabled home automation, you can follow Eric Tsai's modestly titled Uber home automation Instructable (hsmag.cc/vshQrY). Eric equips his house with various Arduino-controlled sensors that mimic four of the five human senses to detect and respond to over ten different activities. His system senses his dog's bark, tracks it when it runs away, and even maps all the places it poops. There's also an intrusion detection sensor, a washer-dryer sensor, a water leak sensor, a garage door monitor, a sensor to check if lights are left on, and more. →

Below ↓
All of Eric's sensors feed data into his OpenHAB installation, running on a Raspberry Pi



DIY: IOT-ENABLED PLANT

Get your plant to send you an email when it feels thirsty



Who doesn't love plants?

These little portable bundles of nature make our concrete jungles habitable. But they are also very delicate, and need constant tendering and

nourishment. To ensure we don't lose any because of neglect, we decided to IoT-ise ours to send us an email whenever it's feeling thirsty.

For this project we'll use a soil moisture sensor, hooked up to a SparkFun Blynk Board that'll send us an email whenever the soil in the plant pot gets dry. As we've mentioned earlier in the feature, the Blynk Board uses the ESP8266 WiFi chip, and is a wonderful platform for anyone new to IoT. The Blynk Board interfaces with the outside world using GPIO pins that you can use to control

Below Place the Blynk Board in an enclosure of some sort, away from the plant, to make sure it's protected from water



motors and LEDs and read in values from all types of sensors. The board also has an app called Blynk that makes it relatively straightforward for everyone to control the board.

PROVISION THE BLYNK BOARD

To get started, first download the Blynk app into your Android or iOS device. Then open the app and sign up for a new account. The board comes with a card that has a QR code that you can now scan with the app. Make sure the board is powered. The app will detect and connect to your board.

This procedure doesn't always work, and so you might have to provision the board manually. For this process you'll need a Blynk auth token. To get this token, fire up the Blynk app on your mobile device, log into the account, and tap the button to create a new project. Give the project any name and select the SparkFun Blynk Board from the list of hardware. When you tap the Create button, the app will email you a copy of the auth token. Keep this handy.

When it isn't provisioned, the board acts as a WiFi access point. So head to a computer and change the WiFi network to connect to the BlynkMe-XXXX network, where XXXX is the colour code flashing on the board. When you're connected, fire up a web browser and point it to **192.168.4.1**. This should bring up the setup page where you can enter the auth token and details of your home WiFi network. When you hit Apply, the board will first blink blue and then green as it tries to connect to the Blynk server via your home WiFi. When it's done, the board will 'breathe' by fading the green LED in and out.

MONITOR THE SOIL

With the Blynk Board set up and ready, let's hook up the soil moisture sensor to the board. The SparkFun sensor we used for this project (SEN-13322) had a screw terminal installed for ease. If yours doesn't have one, you can solder a screw terminal on to the sensor yourself. You can also solder the wires directly to the sensor.

In any case, take three different-coloured jumper wires and plug them into the VCC, SIG, and GND terminals on the sensor. Then take three crocodile clip cables (preferably of the same three colours) and connect them to the other end of the jumper cables. Finally, clamp the other end of the crocodile cables into the 3.3V (VCC), ADC (SIG), and GND pins on the Blynk board. Double-check the connections to make sure you connect the wires correctly. The VCC jumper cable from the sensor should connect to the 3.3V pin on the board and the SIG jumper wire from the sensor should be hooked up to the ADC pin on the board. The GND cable runs from the GND terminal on the sensor to the GND pin on the board.

The ADC pin measures the voltage at the pin and turns that into a digital value. The ADC on the Blynk Board produces a value between 0 and 1023, with 0 being 0V/LOW/OFF and 1023 being 3.3V/HIGH/ON.

PREPARE THE IDE

The Blynk Board can be programmed using the cross-platform Arduino IDE that you can fetch from arduino.cc. You'll then have to install the Blynk Board add-on into the Arduino IDE. Launch the IDE, head to File > Preferences and paste the URL https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json into the Additional Boards Manager URLs text-box. Then head to Tools > Boards > Boards Manager, and search for Blynk in the window that pops up. Select the sole entry that comes up and click the Install button to download the add-on. When it's done, scroll down the Tools > Boards menu and select the entry for the SparkFun Blynk Board.

You'll also need to enable or install drivers for the FTDI chip that converts USB data into serial data before uploading your code to the board. The drivers are already installed in Linux and all you need to do is give your Linux user permission to access it. Head to hsmag.cc/dRvACO for

step-by-step instructions for Linux, Windows, and macOS.

SET UP BLYNK

For our project we'll use a combination of the Blynk app and raw code. The Blynk app helps us create an interface for the project that we'll hone using the Arduino IDE. To get started, fire up the Blynk app and open the project you used to provision the board.

Each Blynk project has a different auth token. So, if you create a new project, it'll generate a new auth token that you must use to reprovision the board. If the project already has widgets you can tap on them to open their settings and scroll down to the Delete button to remove them. Do this for all the widgets until you have a clean project space.

Tap anywhere in the blank project space, which should bring up a toolbox from the right side that will list all the available widgets. Scroll down the list and add a Value Display widget that we'll use to monitor the ADC pin. You can tap and hold the widget to resize and drag it anywhere within the project space.

After placing the widget, tap on it to bring up its settings page. Use the text-box at the top to name the widget as Moisture. Then tap on the Inputs button to make sure it points to the Analog ADC pin. When you're done, use the back arrow at the top of the screen to head back to the project space. Tap to add another widget and select the Email notification widget. You don't really need to change the settings of this widget, but you can review them by tapping on the widget.

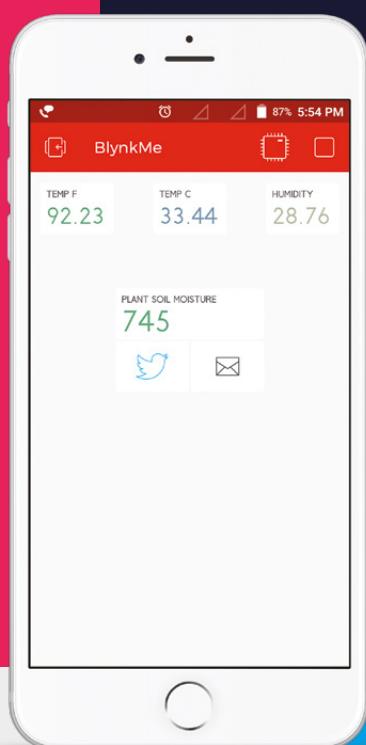
Once you've added the widgets, insert the soil moisture sensor into the plant pot and tap the Play icon in the Blynk app. The Moisture widget should display the current moisture reading – the higher it is, the wetter the soil.

SEND AN EMAIL

When you are done, fire up the Arduino IDE and write some code. First, include the necessary headers:

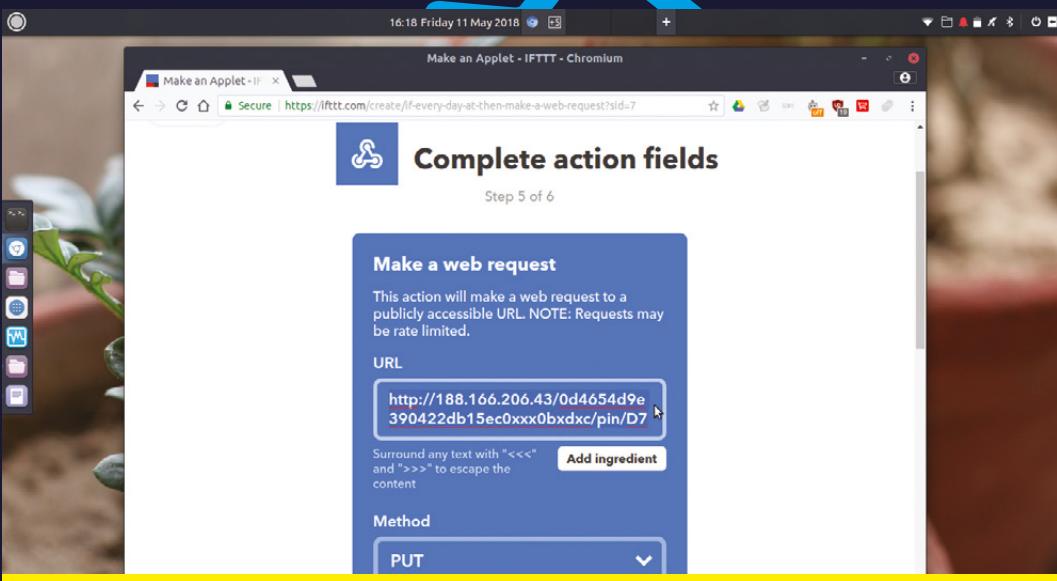
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

Below ↓
Set up a Twitter account for your plant, and use the Twitter widget in the Blynk app to get your plants to tweet their moisture levels



FEATURE

Right ↘
 The IFTTT webhooks URL to control your Blynk board is in the form:
`http://YOUR_COUNTRY_BLYNK_IP/YOUR_TOKEN_HERE/pin/V1`. Replace V1 with the pin you wish to access and ping `blynk-cloud.com` to find Blynk's IP from your country



The first one prints the built-in Blynk status prints that helps you debug issues. Next come the Blynk settings:

```
char BlynkAuth[] = "YOUR-AUTH-CODE";
char WiFiNetwork[] = "SSID-NAME-OF-YOUR-NETWORK";
char WiFiPassword[] = "PASSWORD";
```

Make sure you change the text within the double quotes to values from your setup/network. Next up, we'll define some variables:

```
int moistureThreshold = 500;
int moisture = 0;
bool sentNotice;
BlynkTimer timer;
```

We've arrived at the moisture threshold value after observing the Moisture values in the Blynk app. The plant will send us an email when the soil moisture level falls below this threshold. We'll use the `sentNotice` Boolean variable to ensure we don't flood the Blynk servers with email requests. The timer is a special type of variable that'll help us ensure that we check moisture levels at regular intervals.

```
void isPlantThirsty() {
    moisture = analogRead(A0);
    Serial.println("Moisture reading is (" + String(moisture) + ").");

    if (moisture < moistureThreshold & !sentNotice)
    {
        Serial.println("Plant is thirsty."); // This
```

can be seen in the Serial Monitor
`Blynk.email("geekybodhi@gmail.com", "Thirsty plant", "The plant in the windowsill is dry.");`
`sentNotice = 1;`
`} else if (moisture > moistureThreshold) {`
`sentNotice = 0;`
`}`
`}`



Right ↗
 Luis Ortiz has configured the Blynk app to monitor data from the various sensors, and the status of his custom irrigation system, and also control some parameters like the moisture threshold level and the watering interval

This is where the magic happens. We first read the current moisture value from the sensor into the `moisture` variable and also print it to the serial monitor. We then compare this to the value of the threshold we've defined earlier. If the current reading is less, we print a message ('Plant is thirsty') on the serial monitor and then send an email using the `Blynk.email()` function. The first value is the email address we wish to send an email to, followed by the subject of the email and then the body.

After sending the email, we change the value of the Boolean variable to make sure we don't send out another. Next time around, even if we haven't yet watered the plants, we won't be sent an email, because the Boolean is set. Only after we water the plants and the `moisture` variable measures greater than the threshold does the Boolean variable reset.

```
void setup()
{
  Serial.begin(9600);
  Blynk.begin(BlynkAuth, WiFiNetwork,
  WiFiPassword);
  while (Blynk.connect() == false) {
  }
  timer.setInterval (20000L, isPlantThirsty);
}
```

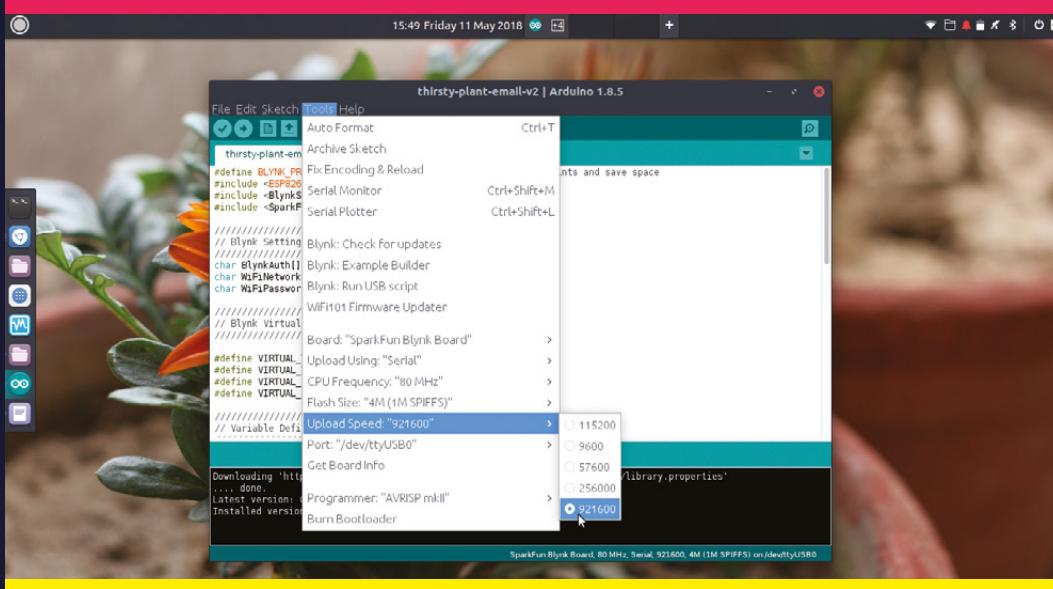
In this block of code we initialise the Blynk board using the values specific to our network and

then wait 20 seconds before calling on the function to check the moisture levels. And finally we get the ball rolling with this:

```
void loop()
{
  Blynk.run();
  timer.run();
}
```

After writing the program, press the checkbox icon to make sure there are no syntax errors in the code. Then, ensure the Blynk Board is connected to the computer before you click on the right-pointing arrow icon to upload the code to the board. When it's done uploading, click on the magnifying glass icon in the extreme right corner to bring up the serial monitor. You'll notice the status messages as the Blynk Board connects to your WiFi network and then the Blynk cloud server. You'll then see the values from the moisture sensor scrolling across the serial monitor. After a while, pluck the sensor from the pot, which will drop its value drastically. Confirm the drop on the monitor and then check your inbox. There should be an email asking you to water your plant.

You can easily extend the project by employing the IFTTT service to take further actions after receiving the email. You can also follow Luis Ortiz's project (hsmag.cc/lhuvVW) and hook up a water reservoir to automatically water the plants in addition to monitoring them. □



Left ↶
Before uploading your code to the Blynk Board, head to Tools > Upload Speed and increase the upload rate to 921600

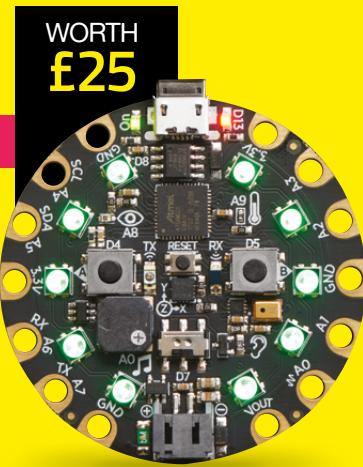


CIRCUIT PLAYGROUND EXPRESS

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

Visit: hsmag.cc/subscribe

SUBSCRIBER BENEFITS ↓

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

OTHER WAYS TO SUBSCRIBE

Rolling subscription from £4 a month:

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

Digital subscription from £2.29 a month:

- Direct to your mobile
- For both Android & iPhone
- No delivery fees
- Back issues available



Visit: hsmag.cc/subscribe

How I Made

ELECTRIC LEGO MINI COOPER

By Peter Kent

The Lego Mini Cooper is an awesome thing, capturing the very essence of the real thing, while preserving the integrity of Lego itself. This is especially true if you have any fondness for the original Mini Cooper or even its BMW-owned reincarnation. And no, nobody here is being paid by Lego, or BMW, to say that. Although that would be nice.

When my (grown-up) daughter completed the Lego Mini that I gave her for Christmas, we immediately deemed it a thing of perfection and took it on an extended photo-shoot with her real Mini Cooper. Except, nothing is quite perfect, is it? Or at least, the human condition is to view what was once seen to be perfect as vaguely unsatisfactory. This is why we need landfills and divorce lawyers.

Happily, it is also a truth universally acknowledged that something that is almost perfect is merely in want of LEDs. Indeed, there are any number of ready-made lighting kits you can add to the Lego Mini if you are in possession of a good fortune. In this project, we will save our money and install a custom lighting kit with web-based control from your phone. All for around \$10!



Above ◀
The finished Mini showing off its WiFi-controlled lighting

Before getting started, a word of caution. The Lego Mini is fragile, and modifying it to include the lights means that you will be constantly teetering on the edge of causing it to revert to its original factory condition. While this would be an almost magical property in a real car, it is less desirable when 'factory condition' is a jumble of 1077 mysterious choking-hazard pieces. If you ever doubted that entropy is the defining characteristic of the universe, look no further than Lego. During exceptionally tricky bits, a line from Yeats's *The Second Coming* may come to mind.

BUILDING BLOCKS

You'll need a Lego Mini Cooper, obviously. Or, you could substitute the equally marvellous Lego VW 'Splitty' bus and use much the same code and overall approach, though the wiring would of course be a little different. The only other component of consequence is an ESP8266-based NodeMCU. About \$6. Then all you need are a bunch of 3mm LEDs, a few resistors (including an optional LDR/photoresistor), a couple of NPN transistors, thin hook-up wire, and heat-shrink tubing.

You can use other ESP8266 development boards, but the NodeMCU represents

a good compromise between size, pin availability, Arduino IDE support, and price.

On the tools side you'll need a soldering iron and precision drill press with mint-condition bits (3mm and down). I didn't have the latter so I used a hulking great DeWalt with my usual chewed-up bits, which worked fine. A variety of smallish hand tools will also come in useful. I found a pair of pointy tweezers especially helpful.

WIRING INSTALLATION

The challenging aspect of this project is deciding how to get wires to all the lights. Each of the sets of lights (headlights, fog lights, turn signals, interior lights) needs a slightly different approach. In general, however, wiring to all lights except the fog lights can be installed by drilling a small hole from the centre of the light straight back into the next available cavity, then routing the wire inconspicuously to the underside of the Mini. For the front fog lights, the wire must enter the rear of the reflector slightly off-centre. The pictures give you a better idea of the approach I used for the exterior lights.

The RGB interior LEDs are easy to install by drilling a hole straight up through the underside of the car so it emerges between and slightly to the rear of the front seats. I used three separate lights because I didn't have any combined ones. Judicious application of hot glue keeps them together and has the additional benefit of providing some diffusion.

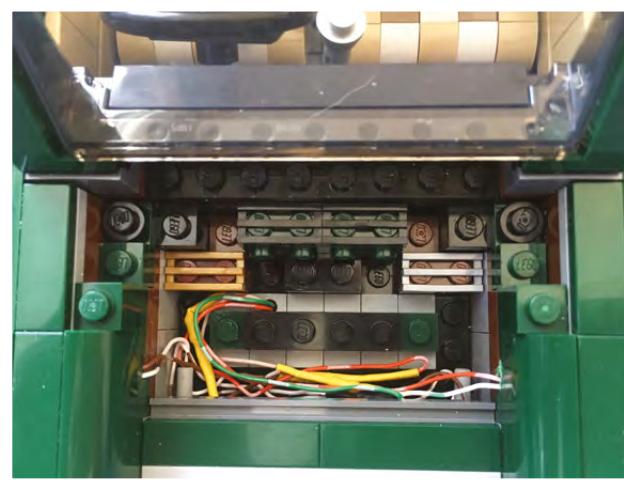
Features

- Individual control of exterior and interior lighting
- Lighting selections mirrored on the phone display
- Auto lights feature dependent on light level
- Internet-based timekeeping, so your Mini can flash the hour with its lights
- User-configurable time-zone selection
- A near infinite variety of manually selectable interior lighting colours and brightness levels
- An automatic 'Groovy' mode to alternate interior lighting for that true sixties vibe. Yeah, baby!
- Multicast DNS (no need to remember pesky IP addresses)
- OTA (over the air) firmware upgrades to easily add even more features
- WiFi Manager so the Mini can be moved to new networks without re-coding SSID and passwords
- Tesla-like autopilot. No, not really

SOME HARD-WON TIPS

- To keep the hole sizes as small as possible, you might want to file off the flanges of the LEDs.
- The installation requires drilling through several small bricks that are normally hidden. The drill will want to separate Lego bricks rather than going straight through them. Try to identify when this is likely and clamp these bricks as best you can while you drill.

- The thinness and flexibility of the hook-up wire is crucial to success. I used single-strand conductors from an old telephone cable, which worked OK but was vulnerable to breaking. Very fine multi-strand wire would probably be a better choice, although most breadboard jumper wire is too thick to be ideal.
- Some intentional disassembly of the Mini is inevitably required. At the very least, you'll need to remove →



Left ♦
The engine block keeps the wiring tidy and obscured from view

Far left ♦
The raw wiring under the bonnet

How I Made: Electric Lego Mini Cooper

FEATURE



Left ↳
The front wheel arch with a minor amount of surgery

the wheels, roof, bonnet, boot, and engine block. Some unintentional disassembly is also inevitable. Be at one with it.

- A question that might occur to you is whether it might be easier to do the wiring installation before building the Mini. I'd say no, unless you can

maintain spatial awareness of how 1000+ pieces will fit together.

When you are done, all wires should meet in the large space between frame members under the Mini. This is where the NodeMCU will be installed and the wiring terminated on the appropriate pin. If you plan on powering the NodeMCU via

the micro USB, you should consider which orientation hides the cord best, given your intended location for displaying the car. Providing 5V power to the VIN pin may allow more flexibility in this regard.

THE CIRCUIT

While planning out the project, I made the questionable decision to include the current-limiting resistors for the LEDs, transistors, and LDR in the wiring runs. I did this because I was originally planning to mount the NodeMCU directly on the Mini's chassis. In the end, I mounted the thing on Perfboard so I could have easily consolidated all the electronics in one place. Not doing so will make replacement of components slightly more challenging. I also wish I had used LED resistor values greater than 220Ω , since replacing LEDs will be a pain and the perceived brightness for these LEDs isn't much affected below 470Ω . Oh well.

I used transistors on the front and rear turn signals/hazards to save pins, an approach you could extend to any lights that will always be switched together.

Another tip: the NodeMCU is a great development board for the price, but it is a bit more quirky than standard Arduinos, and available documentation is less complete and occasionally contradictory. In particular, it wasn't clear to me if pins D9 (RX) and D10 (TX) are available as GPIOs if there is no serial traffic. Based on my experience, I don't think they are. Also, pins D3, D4, and D8 determine boot mode, so if you suddenly find your code isn't loading or executing, try moving stuff around on the pins.

THE CODE

The code (Arduino, HTML/CSS, and JavaScript) is available from hsmag.cc/SzCNxV. It leans heavily on an example included in a great ESP8266



Left ↳
Wiring to the rear wheels comes via the spare wheel compartment



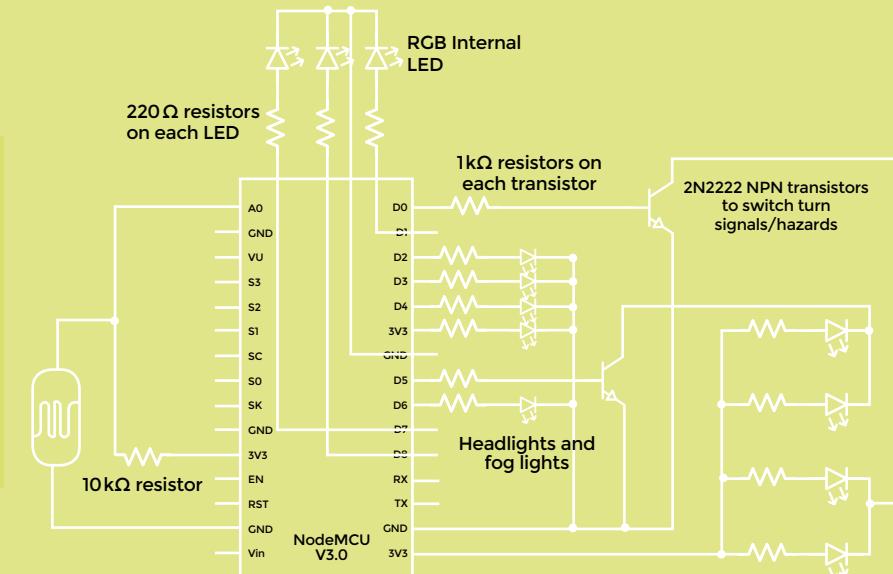
Above You can see the lights in action in a video at hsmag.cc/mLzbub

resource called 'A Beginner's Guide to the ESP8266', the title of which modestly understates its ambition and utility.

Rather than using HTML forms or AJAX requests to send data to the NodeMCU, the code uses WebSocket communication, which allows low-latency commands to be sent without refreshing the page. This provides a much more 'app-like' feel to the user experience. The code includes all the images I used, so it works out of the box – including some images for adding a customised launch button to your iPhone's home screen.

Using WebSocket communication also makes passing data to the NodeMCU easier to implement on the Arduino side, even with only a rudimentary knowledge of JavaScript. In fact, adding additional functionality is so easy, there is a danger of 'feature creep' that probably should be resisted, in our experience.

The problem is not on the software side, it's just that any additional features will likely have to be implemented on the



hardware side in the Lego Mini. At some point you will run out of hiding places for additional wiring, sensors, and suchlike

since these features require no additional modification of the Mini.

All the data for the webpages (in the folder helpfully named 'data') is stored in the NodeMCU's SPIFFS area (check that Beginner's Guide again if you are not familiar with how to work with SPIFFS). There is plenty of room for some decent-quality images and the ESP8266 is more than

and I think it is important not to change the Mini's outward appearance.

That said, since making the YouTube video, I've tweaked the GUI and added a slider control for transition speed of the RGB interior 'Groovy Mode' lights

capable of serving them up very quickly.

Using the WiFi Manager library means your SSID and password do not need to be hard-coded and you can take your Lego Mini around to your best friend's house when you go on a play date. □



Right Unfortunately, improving the lighting won't actually help the traction in the snow

DIGITAL BLACKSMITHS NETWORK

Changing the future of manufacturing in the developing world



Cameron Norris

[@cameronnorris](#)

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

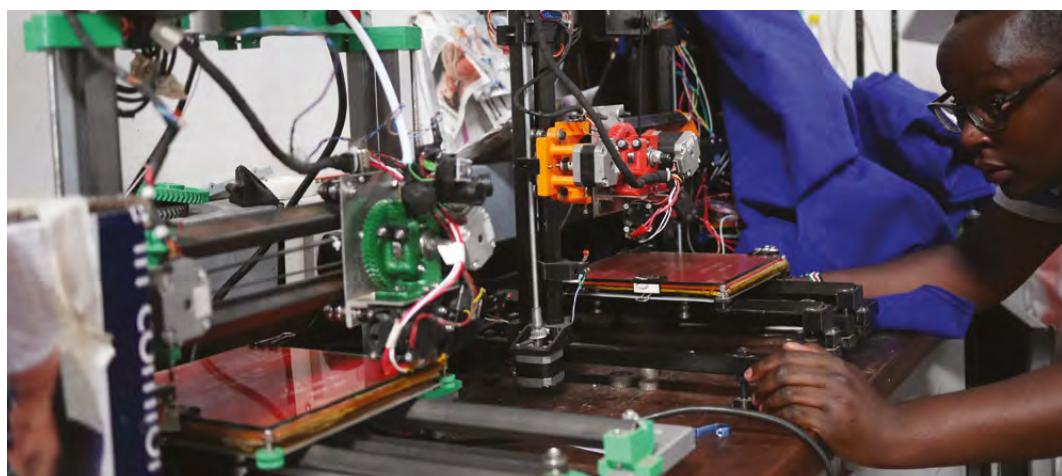
Through TechforTrade, thousands of tonnes of e-waste and discarded plastic that would otherwise end up in landfill is being transformed into a commercial and design opportunity for people in developing economies using open-source 3D printing technology.

TechforTrade is building a global 'Digital Blacksmiths Network' with the goal of alleviating poverty and pushing for economic growth through open-source product development, while ensuring that network members have access to the technology and training they need to provide 3D-printed products to their local communities.

TechforTrade's CEO William Hoyle has been working to alleviate poverty in developing countries through technology and entrepreneurship. TechforTrade is a leading UK charity specifically focused on bridging the divide between innovative new technologies, global trade, and economic development. Based in London but with impact in communities around the globe, TechforTrade has initiated a number of projects using 3D printing for international development.

DIGITAL FABRICATION

William and the TechforTrade team believe that global access to 3D printing technology is a vital step



Right ◆
Putting the Retr3D printers to work

towards the democratisation of manufacturing. It has the potential to eliminate the need for distribution by transporting physical products 'electronically', as digital designs. By lowering costs and barriers to entry, William believes 3D printing has the potential to create opportunities for micro-businesses to enter new markets by producing 3D-printed products.

TechforTrade launched the Retr3D project with the goal of providing affordable 3D printing equipment for developing economies using the 41 million metric tonnes of e-waste that is discarded globally each year.

The open-source Retr3D software enables anyone to build 3D printers using different components from discarded inkjet printers and photocopiers, and inputting the dimensions of procured e-waste components to generate a unique 3D printer design.

Matt Rogge, TechforTrade's technical director, believes 3D printing can be as transformative in developing economies as the mobile phone, and that by introducing 3D printing at a community level, jobs can be created through local manufacturing: "With a 3D printer people can gain access to a wide variety of educational, medical, and mechanical materials that would otherwise have been inaccessible."

Matt first became involved in 3D printing after working as a volunteer building rural water systems in Panama; here he first heard about 3D printing and saw an opportunity to help the local community.

However, Matt soon discovered that the number of specialised or proprietary parts required to keep conventional 3D printers functioning in rural communities was far too high: the price of imported parts and printers was often double or even triple the original cost due to shipping and customs charges.

While working in Mexico, Matt had the idea of creating 3D printer design software that could



Matt Rogge

"A retread is a type of tyre made so that the surface can be replaced when it wears out. William Hoyle came up with the name Retr3D because the printers are a way of giving new life to the still usable components found in obsolete electronics"

automate the design changes and modifications required to make and maintain a 3D printer using the lowest cost, local source of parts possible: e-waste.

Initial testing of the designs generated by his Retr3D software showed that many printers were sensitive to transportation on rough roads. Loss of calibration and damage was commonplace. However, by adopting a design with a welded steel frame, which could be outsourced to local welding shops, the 3D printers proved to be much more durable.

Also, by ensuring that everything could be built locally, replacement parts could be easily obtained and someone with the skills needed to make repairs was never far away. By releasing the software and designs as open-source, TechforTrade enabled anyone wanting to build a Retr3D printer to access the information needed to keep it running and make improvements.

Open-source electronics have helped to reduce the printer's cost and improve its maintainability. The popularity of RAMPS 1.4 controller boards, commonly used with RepRap-style printers, has helped to make them low cost and widely available. The electronics on a Retr3D printer cost less than £25 and, thanks to the modular design of RAMPS, damaged components can be replaced for far less. The total cost of making a Retr3D printer is only around £100, which includes the labour cost of welding the steel frame.

"The best part of this project has been the opportunity to work with such amazing people," says Matt. "I have seen nothing but awesome creativity and dedication to making something great happen. I love to see people who have become involved in the project talk about their goals and how they see 3D printing as a way to achieve them." →

Above ▷
Matt Rogge (right) and the Digital Blacksmiths team at the University of Nairobi, Kenya. Image: TechforTrade

Below □
The assembly line of Retr3D printers at AB3D



RECLAMATION

The idea for 'e-waste 3D printers' was pioneered by West African inventor Kodjo Afate Gnikou, who built the first 3D printer produced from e-waste in his workshop in Togo.

THE THUNDERHEAD FILAMENT EXTRUDER

In developing countries without established transport or supply networks, the local cost of 1 kg of 3D printer filament can go up to as much as £60. In turn, this prevents those interested in using 3D printers from accessing the supplies necessary to utilise their machines at a reasonable price. To tackle this problem, TechforTrade developed the open-source Thunderhead filament extruder, a small-scale extruder designed to recycle PET plastic bottle flake directly into 3D printing filament. Nearly all low-cost 3D printers are fused-filament fabrication machines, which use 3 mm or 1.75 mm diameter plastic filament

Below The Chopper deconstructing a plastic bottle into even flake

as their raw material. This filament is typically produced on an industrial scale before it is imported into developing countries where the market is quite small. This importation can be costly and time-consuming and often leads to unpredictability in filament supply, which reduces the viability of running a successful 3D printing-based business in these areas.

The Thunderhead Filament Extruder provides a low-cost, small-scale means of locally producing 3D printer filament from post-consumer polyethylene terephthalate (PET) bottle flake. Waste PET is available in nearly every location on earth, which not only helps to reduce the challenges of import duties and unpredictable supply chains, but also allows for value to be added to the waste plastic close to the site of collection. PET is an ideal plastic for 3D printing, as it has a low amount of shrink like PLA, and yet is tough like ABS.

PET SOUNDS

Furthermore, small communities in developing economies often lack access to recycling facilities, and without a local means for size reduction or compaction, it is frequently too costly to transport high-volume, low-mass plastic bottles to a centralised processing centre. Instead, a low-cost, small-scale recycling system such as the Thunderhead could be a viable means for small communities to handle problematic plastic waste, while also gaining access to 3D printer filament.

In many developing countries, recycling systems are mostly informal, with vast numbers of waste collectors working in often dangerous conditions to make as little as £1.50 per day selling collected plastic waste at £0.10/kg. The capital investments required for setting up recycling facilities are generally high,





requiring expensive equipment and lots of space in centralised locations. As with many commodities, the route to profitability is through the processing of large volumes.

Developing low-cost equipment for the small-scale production of recycled 3D printer filament keeps the value-added steps within the country of origin, and by reducing the necessary capital investment required to get started, it becomes possible for those closer to the bottom of the supply chain, including waste pickers, to benefit from the value addition process.

TechforTrade is currently working with NGOs in Kenya to employ refuse pickers in the Dandora slum, a waste site a few miles from central Nairobi, to collect, clean, and shred waste plastic ready to turn into filament. As part of this initiative, they have established the Ethical Filament Foundation, hoping to replicate for 3D printer filament what fair trade has done for coffee.

The aim is to partner with organisations that encourage the manufacture of 'ethically produced' 3D printing filament, made from recycled plastic waste, as a viable alternative to the standard virgin plastic spools. The ethical element extends not just to feeding recycled waste plastic back into the 3D printer ecosystem, but to provide a sustainable income for vulnerable waste pickers. The Ethical Filament Foundation is now working to develop a standard for ethical filaments that can be used to certify producers, who will be able to license and display its accreditation mark.

A RUBBISH INNOVATION

"After realising a gap in the market for 3D printer filament made from recycled plastic, we immediately

FROM TRASH TO TREASURE

The value chain from waste PET to 3D-printed products:

- Waste plastic bottles are valued at around £0.10/kg
- When converted to clean flake, the value rises to around £0.50/kg
- 3D printer filament produced from clean flake sells for around £20/kg
- 3D-printed parts made from recycled filament can reach £400/kg

recognised the opportunity this presents to the developing world where plastic waste is in abundance," explains William. "By making the first move into ethical filament, we hope to raise awareness about the importance of this technology and the benefits it can provide to some of the poorest people in the world."

To overcome the technical challenges in producing PET filament of sufficient quality, TechforTrade is developing a bottle washing and de-labelling station and a chopper for producing flake, to support their filament extruder, all of which are open-source. The total materials cost for the whole system is less than £1500.

The Thunderhead filament extruder can produce 5–10 kg of filament per day, which is sufficient to supply 20–40 Retr3D 3D printers for full-time production. Based on the sale price of £20 per kg for recycled PET filament, the extruder is capable of producing £100–£200 worth of filament per day. →

Left Three forms of the same plastic, from waste to product

Below The Thunderhead, ready to convert plastic flake into filament



TREASURE MADE FROM TRASH

Right ♦
The 3D-printed microscopes are ready for a trial in Kenyan schools



Roy Mwangi is the founder of AB3D (African Born 3D Printing), a social enterprise that manufactures Retr3D printers and 3D printer filament using the Thunderhead Filament Extruder in Kenya. Roy's first social impact project, 'Happy Feet', provides bespoke 3D-printed shoes for people with foot deformities caused by jigger sand flea infection.

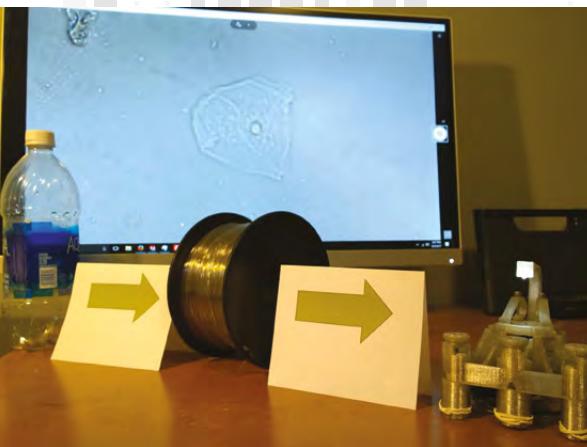
"The sand flea is a tiny insect that feeds on mammals, including humans. It burrows into the skin and destroys the surrounding tissue. People develop foot deformities, which leads to amputations or even death," Roy explains. "Many of those affected by the sand flea infection cannot wear normal shoes because of their deformities."

By creating specially made shoes, Roy aims to help sufferers remain mobile, while also lessening the chance of re-infection. Over a 20-month period in

Kenya, around 50 000 students dropped out of school due to jigger infections alone, highlighting the frightening severity.

3D printing technology is widely viewed as unreachable by many Kenyans due to the pricing of the machines, but TechforTrade partnered with Roy and other social entrepreneurs in Kenya to teach them how to build e-waste 3D printers, while also providing financial support for their micro business ideas, including the Happy Feet project.

"For as long as we use petroleum products, we will always have plastic as a by-product; for now, the best thing to do is to find positive uses for it," says Roy. Alongside Happy Feet, AB3D has also begun fabricating and distributing open-source 3D-printed microscopes to help Kenyans identify bacterial contamination in places that lack access to clean water. Using a Raspberry Pi and camera module, the OpenFlexure Microscope can take high-resolution



Above From waste to educational tools in two easy steps

time-lapse images of bacterial growth within a water sample to test for dangerous pathogens including E. coli, Listeria, and Legionella. In addition to the electronics, the design requires only around 100g of plastic and a few nuts and bolts.

The design uses thin flexible hinges, rather than the mechanical joints that are machined from metal in conventional microscopes. This reduces the friction and vibration of the 3D-printed parts, which in turn enables the microscope to achieve steps below 100 nanometres – which is 1000 times smaller than the thickness of a human hair – when driven with miniature stepper motors.

LOOKING CLOSER

"We're on the brink of launching a microscope trial, with an aim to donate 100 3D-printed microscopes to Kenyan schools, made from recycled plastic collected locally," reveals Selam Zeru, TechforTrade's Entrepreneur in Residence and Business Lead. "Just this month, we closed a crowdfunding campaign raising almost double our target to fund this trial."

Besides checking for bacterial growth in water samples, the microscopes can also be used as an aid for interactive STEM teaching in Kenyan schools to enhance the learning experience. Selam believes that if the trial is successful, it will stimulate interest within the education sector, which will justify scaling up local production of educational tools such as microscopes and other interactive learning aids from within the Digital Blacksmiths Network, creating additional opportunity for entrepreneurs like Roy Mwangi.

LEARNING EXPERIENCE

And things are already looking promising as the Kenyan government announced in 2017 that it would shift its policy by putting more resources into

Katherine Hughes
Fundraising and Communications
Office at TechforTrade

"We are thrilled at the enthusiasm these machines have received from around the world, but particularly in East Africa. As with so much 'new' technology, including mobile phones and mobile money, entrepreneurs in developing countries are enthusiastically adopting and adapting 3D printing technology for use in their own communities"

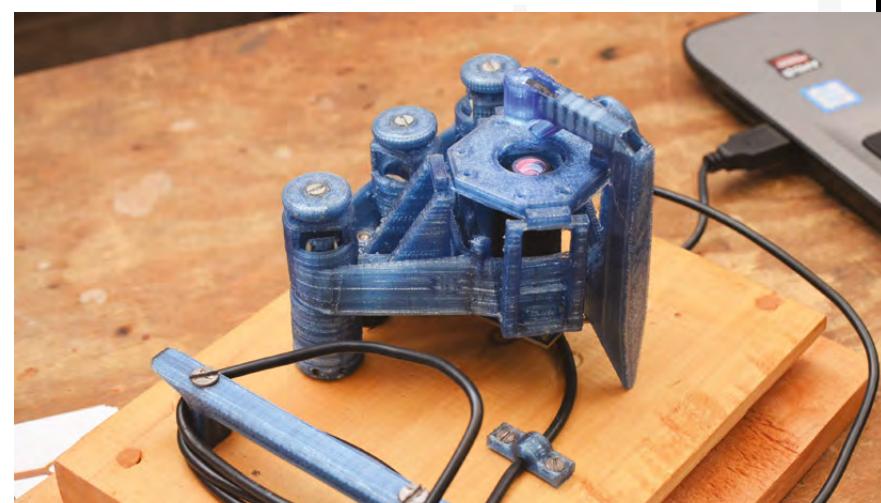
promoting STEM education to increase the number of students pursuing STEM courses in higher education.

The OpenFlexure Microscopes are now being tested in Kenya by Farm Africa's SIDAI group of veterinary technicians, while a team of supporters based at the University of Nairobi work on developing curriculum-relevant content to include the microscopes in STEM education. Dr Richard Ayah, Director of the Science and Technology Park at the University of Nairobi comments, "This project seems to inspire everyone who comes across it, so we're excited to see how we can get more support to push things further and faster. The potential impact of being able to print really useful items at a fraction of the cost is massive."

Matt Rogge adds, "It's taken us nearly four years to get to the stage where we can go all the way from bottle to printed product, so we've finally hit the most exciting time for Digital Blacksmiths – Now we need to focus on how we get as many of these products into the hands of people who need them as we can."

To find out more about the Digital Blacksmiths Network visit digitalblacksmiths.org.

Below A 3D-printed microscope made from recycled waste



DIVERSITY

MORE MEMBERS MORE SKILLS + MORE INSPIRATION

How to grow a hackspace by attracting members from a wide variety of backgrounds

If you want to lose the attention of a group of people, one possible way is to start talking to them about diversity.

Turgid corporate training days will come to mind, in which consultants lectured them on self-evident things, and they will turn off and think about football, or what they had for dinner last night. Diversity training is an annoying experience because people are either being told again what they think they already know, or they are being confronted with things that might be slightly uncomfortable and which they would prefer not to hear. There really are few things more boring or demotivating than a company diversity training day.

But diversity doesn't always have to be about lectures, and about making people feel as though they are somehow transgressors, when in fact they are simply required to be there as part of their employment or education. Instead, diversity should be about opportunity, about making the most of the resources a diverse range of people represent, and about benefit rather than boredom.

A typical hackspace can present an inspiring and interesting membership with an astonishing variety of skills, but it is undeniable that those members will

often look and sound very similar. This is not in itself a bad thing, but it does hide an unintended consequence. For people with different backgrounds and life experiences, such a homogeneous gathering can be somewhat intimidating. The opportunity to enrich and increase the membership is lost, because potential members who come through the door are often put off and never return.

DIRECT DIRECTORS

Those of us who have been involved in the running of hackspace will be very familiar with this. We will all have seen potential members interested in using the laser cutter or the 3D printer, perhaps women in their twenties slightly on edge having found their way to what is usually a slightly insalubrious location due to cost constraints, being put off by well-meaning but socially inept members. The aim here isn't to separate groups of people, but to ease a new member gently into the social aspect of life as a hackspace member.

Less obviously problematic to observers, but every bit as off-putting for new female members, is when through being polite and friendly they accidentally acquire the attention of an overly keen helper. That is, a member who will pop up constantly every time she is



Jenny List

@Jenny_Alt0

Jenny is the creator of the @LanguageSpy electronics kit and a key member of Oxford Hackspace.

**Left**

We hope American airmen and women have better diversity training sessions than we've had

Credit

US Air Force. (CC0)

in, wanting to accompany her in her hacking, and asking to keep in contact. Though this member often has the best of intentions, this amount of attention can make the new member feel uncomfortable. Since it doesn't appear to cross any boundaries, there is rarely any intervention, but do not doubt that she will soon make her polite excuses and leave.

SOLUTIONS NOT PROBLEMS

The answer to this problem of how to retain rather than repel these members is very easy to say but extremely difficult to achieve: ensure that the space is a welcoming environment for them. The verification of this goal lies in how many members from diverse backgrounds you are attracting and retaining, and any answers should come from the impression those people have of your space rather than the self-impression of the dominant group within it.

In the first instance there is an easy win when it comes to attracting and retaining members: how welcoming does your space look? What does a visitor see when they approach your front door? It's true that most spaces have to take whatever they can get and they are often in the cheaper parts of town, but that notwithstanding, are you doing everything you can to make your entrance look welcoming? Walk up to it both in daylight and after dark; is it clean, bright, and, most importantly, well lit? And finally, is it well signed?

Once a prospective member crosses the threshold, their first impression is crucial. Therefore it's most important to pay attention to how new arrivals are handled by the first people they encounter. Our community attracts people from a significantly technical background, and it is fair to say that among the members of a typical hackspace are likely to be people with all levels of social skill. They may all be well-meaning, but in situations where it is the first impression that matters, there will be members who you would be happy to greet new arrivals, and those who perhaps you wouldn't. If you are running a hackspace, it is important that the job of meeting and greeting new faces falls upon someone with the right selection of skills to put them at their ease. It is a job that requires charm and diplomacy, something which if you are a hackspace director you may have to employ

yourself when selecting a member of your team for this crucial task.

So you've attracted your visitor and they're excited about being shown the laser cutter; can you now pull a Club-Mate from the fridge and enjoy your reward? Not just yet, because things can still go wrong. Until the newbie has got to know the members a little and learned that they are nothing to be afraid of, there is still a chance that you could lose them. If your resident Fidel Castro launches into an hour-long lecture on the benefits of anarchist-hacker collectives and nobody rescues them, that may be it. A hackspace director's job is never done, and on social nights that means keeping an eye out and having a trusted team of helpers ready to extricate and shepherd new members should that be necessary. We've covered the aspect of increasing your space's diversity that involves making it a more welcoming environment for new arrivals who perhaps don't feel entirely at ease. But there is another completely different aspect to the issue, that of attracting those arrivals in the first place. For that you will need a much more fundamental diversity strategy, as well as a series of attractive and interesting events and activities. It's time now to take a look at a real-world example of a space that managed this. →

Below

A well-presented entrance makes South London Makerspace a place you'd want to enter

Credit

South London Makerspace



DIVERSITY + AT OXFORD HACKSPACE+



Oxford Hackspace (Oxhack) is a typical medium-sized makerspace in the ancient British university city. At the start of 2017 it had about 70 members, but, due in no small part to a well-executed diversity and events strategy, that admirable number rose 66% over the course of the next nine months. To find out how they approached the question of hackspace diversity, we interviewed long-standing Oxhack director Lauren Hutchinson, a leader in diversity within the UK makerspace and innovation community.

Below ■ Oxford Hackspace holds its own Sawdust Studio woodworking event

Credit Oxford Hackspace



HSMag How did you come to this, what gave you an interest in hackspace diversity?

Lauren H Diversity and inclusion in civil society, and fairness in accessing the means to building a good life, have been topics close to my heart for as long as I can remember. It sounds trite but this stuff really matters to me. Suffice it to say that I have witnessed, experienced, or helped others through enough exclusionary and/or difficult situations as a youth and young adult that, as an empathetic and ethical person, I have found that I can't easily let arbitrary exclusion lie. Experiences working in youth politics, Aboriginal legal aid, international development, with deaf people, EFL learners and offenders, and advocating for health patients definitely inform my thoughts. Like many in the maker community, I was a geeky child with trouble fitting in, and for many years as a young woman I had great trouble accessing the makery tools and upskilling opportunities that I would have liked to have done.

I also had, and still sometimes have, trouble being listened to when I speak about the latter challenges. So on my watch in the hackspace movement, I am quite keen for everyone who wants to grasp these democratised opportunities for learning and making to be able to bloom, without any arbitrary barriers getting in the way.

HSMag Did you prioritise targeting any particular group or section of society?

Lauren H No, not particularly. People with physical disabilities, people on the autistic spectrum, and women, minorities and LGBT+ were all people we tried to include, and there were multiple things I identified or theorised and experimented with that we



thought might help them gain access and feel welcome. One of the first things we did on moving into our new place was survey it from an accessibility perspective and remove ledges and obstructions to entering rooms. We conducted a risk assessment on the lift and found out that people in mobility chairs could get stuck if they entered the small space one way but not the other, so it was important to signpost from outside where the interior controls were located. Same process with the loo, though the building didn't implement all recommendations. We created some comfortable spaces centred around traditionally

itself into being, despite people being generally friendly and welcoming. As co-ordinator, it seemed to me that we weren't meeting our mission unless we made ourselves as physically, financially, and geographically accessible as possible, and actively encouraged a healthy diversity balance in the membership. The other directors were lovely and supportive of this. For some time I was loath to push on diversity for personal capacity and drama avoidance reasons, but once we achieved a much larger space I decided to champion the issue to hopefully help make us into a place to be proud of, one that could really improve some lives.

Some of our founding members were pretty passionate about inclusive principles being built into our model early on

'female' activities for new people to use as a safe hub, and we ran a number of events targeted at women, minorities, and LGBT+.

HSMag Why did Oxford Hackspace consider this to be important?

Lauren H Due perhaps to the divided nature of Oxford and life hardships, some of our founding members were pretty passionate about inclusive principles being built into our model early on. Accessibility was named in our mission as an important goal, and our minimum fee set to £10 for inclusiveness. Years passed but somehow, a diverse membership never magicked

HSMag Where did you see your greatest challenges at the start?

Lauren H There were a few challenges. One was getting many members to think that a diversity push was more than shruggingly important,

to buy into helping me run these events and implement environmental changes in an active way. On inquiry, people were mostly passively supportive of inclusiveness, but many often didn't go ahead and volunteer time towards things unless I approached them directly or unless they were given direct responsibility for supporting specific people. Then once they had specific actions requested of them personally, they tended to be great helps. Mind you, a small core of members, many with teaching, queer, and/or autism spectrum backgrounds, did actively contribute and offer enthusiasm and support, which was really lovely. A second issue was the question of holding targeted events for minorities, because our regular events →

Left◆
Attendees at one of the monthly Oxford Hackspace textile open afternoons



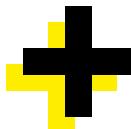
Above Attendees at one of the monthly Oxford Hackspace textile open afternoons

weren't drawing many people who were not youngish, technical, male, and white. We had some drama and discussions over whether targeted but non-exclusive events, or even mildly exclusive 'safe' targeted events at which non-attendees were asked not to linger, were OK to hold to get more people coming along and comfortable with us. There were a handful of objections on the basis of feeling excluded, and we even got a drive-by attack on Twitter, but our regular events were demonstrably just not drawing a diverse crowd, while tailored events did.

Put charismatic people with good personal social development and empathy in charge of greeting visitors

HSMag How did you decide on what individual projects your diversity strategy would contain?

Lauren H The short version is that I brainstormed a lot on it for months, visited other spaces, spoke to friends, asked for ideas from members, and in particular from minority visitors who turned up. I dug into my own experiences of alienation and anxiety in tech spaces, and invited my intuition to suggest incremental improvements to me over a long time. I also read various Nesta and PhD thesis-type documents about welcoming spaces that are floating about, and occasionally bounced ideas off sympathetic directors from other spaces.



My advice on developing a diversity strategy is to put someone from a minority background in charge of thinking it through gently over a few months, and tell them explicitly to listen to their intuition and that it's OK to experiment. Importantly, listen to event participants' feedback.

We have a few members who really get the importance of diversity, and having them take ownership of implementing parts was a great help. And finally, remember that it's important to take account of what resources you have and honestly don't have to work with currently: think about who you have with a high charisma rating from a minority background, who might be able to reach out or be an ambassador, who would be happy to teach a minority-targeted instance of a workshop. Even things like tea, snacks, and interior design skills can help make a place feel like home.

HSMag Which were the easy wins, and which were more difficult than you expected to implement?

Lauren H I've talked about a few above, but here are some easy wins:

- Put charismatic people with good personal social development and empathy in charge of greeting visitors – at the very least on open nights. Also, make sure you give them a backup person or two so that if several new visitors come in separately one after another, it's not like the tour guide is already busy and the second and third visitor are going to fall through the cracks.

- Create a dedicated textiles room or area, and

ideally an explicit crafts area that does not get overrun by all sorts of computers and cabling all the time. The goal is to not make it a requirement of crafting that someone needs to first be confident enough to ask other members to move.

- Think about getting someone female or from a minority background to write up your event descriptions in a welcoming way. Ideally someone empathetic. You'd be surprised at the difference this can make for readers when descriptions are not overly brief, technical, or mildly forbidding-feeling for people with imposter syndrome. Explicitly say what is

needed to bring and if all levels are welcome, say so. Stick these descriptions wherever you normally advertise events, and tweet the link with an inviting and targeted title at relevant communities or individuals in your town who might retweet you to their followers. In general, try to build good relationships with these groups, where you help each other out when possible.

- In terms of interior design, consider creating some hidey holes, so to speak. You want various different seating and working layout options, and also to provide some choice to members around noise and interaction levels required at any given seating position. This will help shy people, people who may be low-level uncertain about their belonging or safety, privacy advocates, and people on the autistic spectrum feel comfortable. There should always be a few seats with their backs to the wall so passers-by can't see screen contents – I can't emphasise this enough! Having snuggly places to sit that are behind bookshelves, protected by friendly plants, or in corners behind piles of fabric are good things to explicitly and intentionally build in. Not everyone will feel as OK sitting at a long table in the centre of the room with random people walking behind them – to some that feels too exposed.
- Consider doing what we did with our vehicle mechanics classes: schedule a workshop

twice on the same evening: one for general all-comers, and one targeted at women, minorities, and LGBT+ people 30 minutes to an hour later. Write them up exactly the same way, but specify the audience and clarify 'exclusion' guidelines (if any) explicitly in the latter. Tweet them out separately, mentioning potentially relevant local groups. Tell your members to follow and retweet the space Twitter stream too! And if you want to make a chain of workshops on the same topic, say

When developing a diversity strategy, put someone from a minority background in charge of thinking it through gently

four Fridays in a row, know that for whatever reason, we found that placing the diversity one earlier and the general one later maximised attendance for us. It seems a lot of women and minority women, at least, need or want to be at home earlier than other groups, often due to dinner or child duties, but for some at least because they don't like being out on their own so much after dark. In the winter, this may mean scheduling your workshops earlier! □



Left ■ Oxford Hackspace stages events where you can listen – or just keep crafting

Credit
Oxford Hackspace

HackSpace magazine meets...

HELEN STEER

Maker, teacher, author, musician...

Helen Steer has our dream job: she makes a living out of coming up with ideas. Whether that's in the classroom, teaching children that there's more to computing than spreadsheets and Hello World, exhibiting at the V&A and Science Museum, or collaborating with international musicians and origami artists, she summons things out of her brain and into the real world.

You'd expect someone like that to have a few ideas about making, and that's exactly what we found when we went to Hackney makerspace Machines Room for a cup of tea and a chat with Helen and her colleague Rehana Al-Soltane. →

Right ↗
Hackney in East London is full of techy startups (and graffiti)





HACKSPACE Thanks for having us here at Machines Room, which is home to an incredible range of makers. Let's kick off with something quite general: what does the phrase 'maker movement' mean to you?

HELEN Well, as someone who's worked in education my entire professional life, 'maker movement' really is a buzzword. Making is something that's been around for as long as humans have been around. All humans love making things. From craft to fish finger makers. I mention that because I watched a documentary on factories, on how things are made. In this fish finger factory was a woman talking about how proud she was when she saw a packet of her fish fingers on the shelves: she'd made them. I thought, you know what? Everybody feels that kind of pride when you make something.

And making encompasses more than robotics and 3D printing. It's a way of learning. It's a way of creating. It can be one of the most intensely creative activities that we can do as human beings. If it's anything, the maker movement is really the integration of new technologies into this human activity of making, whether that's robotics, Arduino, electronics (though electronics isn't an especially new technology at this point).

The thing that stands out about this idea of a maker movement is the idea of makerspaces. Makerspaces and open-source technology have been integral to the idea of 'making' as a movement apart from this kind of human activity of making, which has always existed. There's this idea of sharing and learning and being able to learn from others. There's this collectivism around the maker movement. So, while making is an eternal human activity, what the maker movement for me has done has emphasised the learning, the sharing, the teaching, and the integration of these technologies into what, in my view, is a completely human endeavour.

REHANA In essence we are all makers. My dad was a car mechanic, so when we grew up, we would make little controllable cars, fix them, adjust them, make them go faster, but I never identified as a maker because I never knew it was a thing. Everyone makes stuff, and everyone can be creative in their own projects.

Later on I became interested in tailoring and sewing. I started to make dresses, and my mum helped me, my grandma, my aunts, my cousins, we're all very creative. But I would never have called myself a maker at that time because I never thought it was a thing until I came here; it turns out there's a whole maker movement.

Makerspaces and open-source technology have been integral to the idea of 'making' as a movement apart from this kind of human activity of making, which has always existed

I think it's a community thing. The skills that I've learned are from my dad, from my grandma, from Helen, from other colleagues... Being a maker is a collective thing.

HACKSPACE It's interesting that you talk about dressmaking. It's usually hardware that people are talking about when they use the word – and fish fingers of course.

HELEN Certain types of making are put on a higher level than other types of making, which I don't feel is fair. Craft and craftsmanship are integral to what making should be. Especially when you're talking about young kids, you're talking about learning hand-eye co-ordination, problem-solving, lateral thinking – actually there's nothing wrong with teaching carpentry or needlework either, as well as or instead of electronics. They teach the same skills.

And also, I feel like there's a bit of a class thing. I come from a very working class background, and sometimes feel like the maker scene can be quite middle class. It totally is. I feel quite conscious of the fact that I'm getting all these high fives for doing electronics that my sparky cousin would laugh at. Or my builder uncle – his level of making is going to be far superior to mine.

It's not just a class thing; it's a gender thing. I'm lucky to be working in a makerspace where there are so many technical women. It's unusual. Most of the people I work with in my maker space are women, from different types of backgrounds. Some are incredible electronics engineers, some are really good with different materials, and I think it's a particularly welcoming space. There aren't many women in hardware, and this space is a hub for us.

Making and crafting in my view are the same thing. The worlds have a lot to learn from each other, and that we as makers should respect and raise up the people who have been doing those things for generations, and that means looking outside our own little bubbles.

HACKSPACE Is that why with your Do It Kits [which are aimed at teachers], you purposefully want to keep them not just for the IT classroom, but also the music room and science labs?

HELEN My ethos in education is that I like a diversity of factors. When I'm in the classroom, if I'm teaching electronics, I like to show artists' work as well as robotics work. Because it's really important to understand that what you're learning has a context outside of the technical. We can be inspired from all sorts of different directions.

For example, using an Arduino in your music class, or making an instrument in your physics classroom, can really expand your brain and make you think →



Above ↗
Soldering safety:
always solder in a
ventilated area and
using eye protection



Above Rehana Al-Soltane, left, has been working at Machines Room as part of her degree, learning first hand how to debug sewn circuits (among other things)

differently about the subject that you're passionate about. You see different avenues for your creativity. It's really important for a well-rounded education as well, to be able to have that. Especially when you get to GCSE and A level, you're often siphoned off down one little path. So, if you're doing physics, maths, and further maths A levels, being able to learn about some music theory and be able to listen to music, and about how beautiful and intricate resonance can be... it's a lot more exciting to learn about resonance and magnetic fields when you're doing it by making an electric guitar.

HACKSPACE I must admit that's the thing that caught my imagination.

HELEN It makes you more interested in physics, but also opens your mind to the applications of the thing that you're learning: it takes it off the page and into the real. That's why I really like physical computing rather than just coding. That's what really led me towards being a maker. I really love the tangible; I'd rather code a robot than code a website, 100%.

It can open up the world of technology by including something tangible, by including something unusual, and you can open up that world to children who would normally look at code on a screen and think 'oh, that's not for me'. It really works to try and get the unusual suspects into electronics and into computing and into STEM subjects.

HACKSPACE Mixing up your stimuli like that isn't just for kids, though. If grown-ups only thought in one way, our minds would stagnate.

HELEN Taking artists outside of their comfort zone and seeing how they'd work with some new technology is really interesting. I did a hack with a Japanese origami artist called Coco Sato. We made

interactive radio-controlled flirting devices that were based on this whole language, a nonverbal language that you can do with fans.

There are different signals you can send with the different fans. She made this beautiful paper origami bracelet and a giant origami fan, which we then attached a micro:bit to and coded and paired. Based on the gestures that you gave, it would tell the other person if you were interested or not. If you held the fan up in front of your face, the meaning is "I fancy you; come and talk to me". And it would send the other person a signal. If you wave the fan in front of your face it would send a signal, "I like you, you can come and talk to me but no funny business". And the last one, if you put the fan facing downwards it just sends a big red X. Move on, on your way.

When you combine two types of practices, the richness that you get from this combination is something special. When you put joy and art into technology or technology into an

For a kid who's not interested in coding, you say "Hey, you can make Hello World! Flash up in a screen" the response will be "Yeah, but we've got Facebook"

artist's practice, the combination of the things can lead to something much greater than the two things individually. Keeping ourselves in these little boxes throughout our adult life and throughout our childhood, we're not exposing ourselves to the full potential of our creativity and humanity.

HACKSPACE We've got massive brains. As a species we're so clever, and yet we just sit in our little cubicles and do the same thing every day. Until we retire, and we get to do what we really wanted to do for all those years.

HELEN I want to spend my life making education more integrated. To make science and technology fun and curious and exciting, enticing, and accessible. By demystifying these technologies, we're hopefully equipping young people with the skills they need to change their world. Or maybe just make their world a bit sillier; I'd settle for that. Kids think of robots as things that come in this perfect white casing, but it's not like that. By making imperfect things you are inviting collaboration.

HACKSPACE The physical computing thing; it's always been a bugbear of mine that the first thing we teach new learners to Arduino is how to make an LED flash, and the first thing you learn when you learn coding is 'Hello World'. Who wants to make a computer say Hello World? I certainly don't.

HELEN But who wants to make a robot unicorn? Of course I do. Who wants to learn how to solder while making an electric guitar? Obviously everyone does.

REHANA I do! I was very new to this, I took a coding course two months ago and the first thing we did was Hello World! And I was so happy that I was able to do this. All these skills, they all take time to learn and you have to start from somewhere, which is almost always from nothing.

The aim was to make your own website, and Hello World was the first step to doing that.

HELEN But you were self-motivated to learn. For a kid who's not interested in coding, you say "Hey, you can make Hello World! Flash up in a screen," the response will be "Yeah, but we've got Facebook". Physical computing makes a much more sensible entry point for most people.

I've been using the micro:bit a lot recently, either prototyping with artists or teaching people who've never even seen hardware before, and the micro:bit is a →



great platform for both of those groups of people. It doesn't assume that you're going to become an expert: you can just tinker with it, leave it to one side, and then come back to it. You don't have to learn epic amounts of things to be creative with it.

Something I'm really enjoying at the moment is using gesture control. There's something really tangible about making sound or making visuals with the movement of your hands. It's very expressive. I've been following the Mi.Mu project by Imogen Heap for a while, just because it seems so obvious to me, it seems [such a natural way] to create music.

I'm a musician as well, and the ability to use your hands to alter your voice as a kind of vocoder or sampler or loop effect really

appealed to me. And when I'm teaching, I love using real-life examples that inspire the kids. If you can say "Hey we're going to use a version of this thing that Ariana Grande used on her last world tour", they're going to be excited about that.

What I like to do is bring that sense of excitement and creativity into technology in the classroom. Or in the home; I do a lot of informal education as well. So I thought this would be a great thing to base a hack on. And then, through my music technology work, Imogen was made aware of me, we had a coffee, and I offered to build her a children's Mi.Mu glove in ten days before this big launch thing. And I made one, tested it, and wrote curriculum materials for the things in ten

Left ◇
Helen's robot unicorn kits introduce coding concepts to kids in an immediately accessible way

days. I like deadlines! It was a great piece of work, I really enjoyed doing it.

The aim was to get a musical instrument made by the kids, wired by the kids, and coded by the kids, and then played by the kids, so that the make isn't the end of it. Actually, after the make comes the creativity. So we had one little kid who made his glove into a sound effects glove. He wanted to play something when he punched forward and my favourite piece of functionality was that if it went into free fall, it would trigger a funeral march. So when you fall off a building it plays that. Children are hilarious, they're so macabre.

And of course if you give a gesture-controlled glove to an eleven-year-old girl, they're going to want to start to choreograph things, they'll think of *Shake It Off* by Taylor Swift. .

HACKSPACE I'd think Beyoncé and *Single Ladies*. Because there are only two or three moves in the whole dance.

HELEN It's not so much the dance as the music. So *Shake It Off* has only got four or five phrases of music in there, so it triggers ... They can choreograph what they're doing based on the melody.

Imogen was interested in the education potential of what is essentially a creative technology object.

You have to use block code, you have to use very simple wiring, you can't use any fancy pieces, you can't use anything that you have to add in; it has to be kind of like a closed loop system. You give a kid a few pieces of felt and actually having those constraints and simplifying technology, it can be really difficult to do, to plan a finished object for a child based on complex technology, [to] scale it right back so a small child can make, code, understand and play with it.

But there's also this idea of high or low. So you show a child a Mi.Mu glove and they think it's amazing, but there's no way they can do that for themselves. □



Above Hands up who wants to control robots by waving your hands at them?



ERASERS



Mayank Sharma

@geekybodhi

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

You've been using them to correct your mistakes, but here's how to use them creatively and lay down some rubber

A

long with the ability to write, humans have had to deal with another aspect that goes along with all self expression: making mistakes.

To err is human, and so is to erase. From tablets of wax to bits of rough stone or pumice, we've used just about everything with a coarse surface to erase our writing mistakes.

The modern-day erasers, or rubbers as they are commonly referred to here in the UK (stop sniggering in America), share their origins with the rubber band, since they are primarily made of the same material. In 1770 an English engineer,

Edward Nairne, accidentally discovered the erasing properties of natural rubber and started selling them for this purpose. But Nairne's eraser wasn't very successful. It was too sensitive to weather conditions and crumbled when used. Oh, and it smelled awful.

The next major development

in the history of the eraser came in 1839, with Charles Goodyear's vulcanisation process that made rubber more durable.

Most common erasers are in block form, or are placed at the end of the pencil for quick and easy use. Pencils with built-in erasers on the other end are largely an American phenomenon, and most pencils sold in Europe are eraser-less. The first patent for attaching an eraser to a pencil was issued in 1858 to Hymen Lipman of Philadelphia. Funnily enough, these little rubbers on pencil ends are not called erasers but plugs, and the small bands of metal that contain the plugs are called ferrules.

Erasers these days come in different shapes, sizes, and colours and are made from various materials, from synthetic rubber to flexible vinyl. Vinyl erasers are durable and have minimal crumbling. During the manufacturing process, a softener, usually vegetable oil, is added to make the erasers more flexible. An abrasive material like pumice is added, along with a dye to give the erasers a particular colour. Erasers work because of friction. When you rub the eraser on a pencil mark, the abrasives in the eraser scratch the surface fibres of the paper and this friction produces heat, which helps the rubber become sticky enough to hold onto the pencil's graphite particles. The softeners in the eraser prevent the paper from tearing. As the rubber grabs the graphite particles, small pieces of combined rubber and graphite get left behind on the paper, and that's the crumble you brush off when you've finished erasing.

ERASER FLASH DRIVE

Project Maker
ED LEWIS

Project Link
hsmag.cc/vOArdx

In his attempt to make sure his pen drives weren't pilfered, Ed Lewis decided to conceal one within a pink eraser. He simply sliced the eraser into two unequal portions, with the smaller one serving as the cap, and used a rotary tool to hollow out both pieces. He then pulled apart the plastic bits of flash drive to reveal its internals that he then snugly fitted inside the larger eraser piece, while the USB male plug went inside the smaller one to complete the conceal. "The good thing about this project is that I went into it completely blind about how easy it is to work with the eraser rubber," Ed remarks sarcastically. "It bends and compresses and moves. All of which makes it so frustrating to cut, Dremel, or file down to get a cleaner edge. The rubber gets into the sandpaper or the file. Everything smells awful, and when you're done the rubbery bits that are left on your workspace just wiggle about when you try and brush them away." Ed had just used the drive to transfer some files to his Raspberry Pi when we got in touch. □



Above Carving the eraser was a frustrating experience for Ed, who made just one of them before moving back to his forte – working with wood



ELECTRIC ERASER

Project Maker
ADAM BOWKER

Project Link
hsmag.cc/CDBSja

Adam Bowker knows that an electric eraser “sounds really lazy”, but that didn’t stop him from building one. He placed a 1.5–3V DC motor inside a small box measuring 4 × 2 × 1 inches, along with a 2 × AA battery holder and a 9V battery snap. He then drilled two holes in the box – one big enough for the motor’s shaft and the other for the mini momentary push-button. Adam then used hot glue to secure the connections from the battery pack to the motor instead of soldering them, which seems the right thing to do for someone lazy enough to build an electric eraser instead of expending kinetic energy to rub off their mistakes. After testing the connections, he poked a small hole in one end of an eraser that he had popped out from a pencil, and secured it to the motor’s shaft with hot glue. □

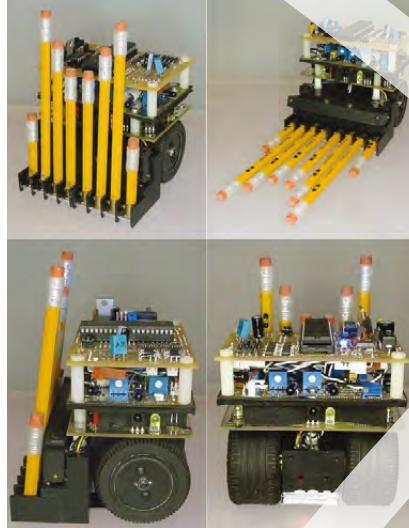


Above ♦
By his own admission, Adam's electric eraser is a fine example of a Chindogu: it works perfectly but is a hassle to carry and use



SUMO ROBOT

The robot sumo contest is just the kind of thing that catches the fancy of serial robot designer **David Cook**. The non-destructive, family-friendly contest pits two autonomous robots that try to push each other out of the ring. The No.2 robot is built on the lessons from the weaknesses of David's prior sumo robots. Unlike the earlier ones that feature wedges on the front to scoop their opponents, the No.2 robot uses Faber Castell No.2 pencils for attack. "Each pencil rides loosely in the robot's front comb," writes David. "Upon colliding with an opponent, the soft eraser tip slides up the competitor until it finds a nook or cranny to grab hold of. All it takes is one pencil to hold back the opposing robot, thus preventing the wedge from coming into play," he explains. Head to David's website for detailed instructions on how to build your own rubber-tipped sumo robot. □



Project Maker
DAVID COOK

Project Link
hsmag.cc/uhHJdD

Left David covers the ferrules on the pencils with clear shrink-wrap tubing to prevent accidental shorts to the opponent robot

MONOGRAMMED STAMPS

When she received a wedding invitation with a personalised monogrammed stamp, **SNLouise** was inspired to create her own using items many people will have in their stationery cabinet. She used a US quarter coin to trace a circle on an eraser that she then cut out with an X-Acto knife. You'll then have to sketch your design and transfer it onto the eraser cut-out. If you've printed your design with an inkjet printer, SNLouise suggests you place the design ink-side down onto the eraser and then dampen the paper. The ink from the paper will transfer to the eraser. Once the design



Project Maker
SNLOUISE

Project Link
hsmag.cc/wLtiSS

is on the paper, she uses a heated soldering iron to trace the design and emboss it on the eraser. Others in the comments and on the internet have used an X-Acto knife to carve the design. Once it's cooled, use a stamp pad to personalise your invitations. □

Left Remember to transfer your design in reverse to the eraser, to have it come out in the right direction when stamped

Over 10,000 Different Boards and Modules In Stock

NO ONE KNOWS BOARDS LIKE DIGI-KEY!

- Open Source Community Boards
- Evaluation and Development Boards
- Wireless Modules
- Single Board Computers
- Reference Designs
- Programmers, Emulators and Debuggers

**DESIGNS START HERE!
DIGIKEY.COM/BOARDS**

Digi-Key
ELECTRONICS



#MAKEWITHDIGIKEY

Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2018 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA



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
96

ESP32

Computing power,
connectivity, and low price
all in one board

PG
100

MAKE PCBs

Turn your garage into a
small-scale fabrication factory
for quick, cheap boards

PG
106



SPHERIFICATION

Unleash your kitchen
chemistry set to create
bursting bubbles of flavour

PG
78

SCHOOL OF MAKING

Start your journey to craftsmanship
with these essential skills

78 Arduino

84 Z80

90 Laser-cut gears

PG
110



LIGHT-UP SIGN

Use EL wire to create a
neon-effect sign without the
glass bending or inert gas

Arduino programming: Temperature, humidity, and libraries

Add temperature and humidity readings by summoning external expertise and knowledge, with just a few lines of your own code



Graham Morrison

@degville

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

In the previous tutorial, we applied our nascent C programming skills to extend what we'd learnt about seven-segment displays into controlling four seven-segment displays concurrently, all from the same Arduino Uno.

We finished with the display counting from 0 to 9999 over and over, like a 5 volt Sisyphus. This means we now have the ability to display a four-digit number. Or maybe two smaller ones side by side...

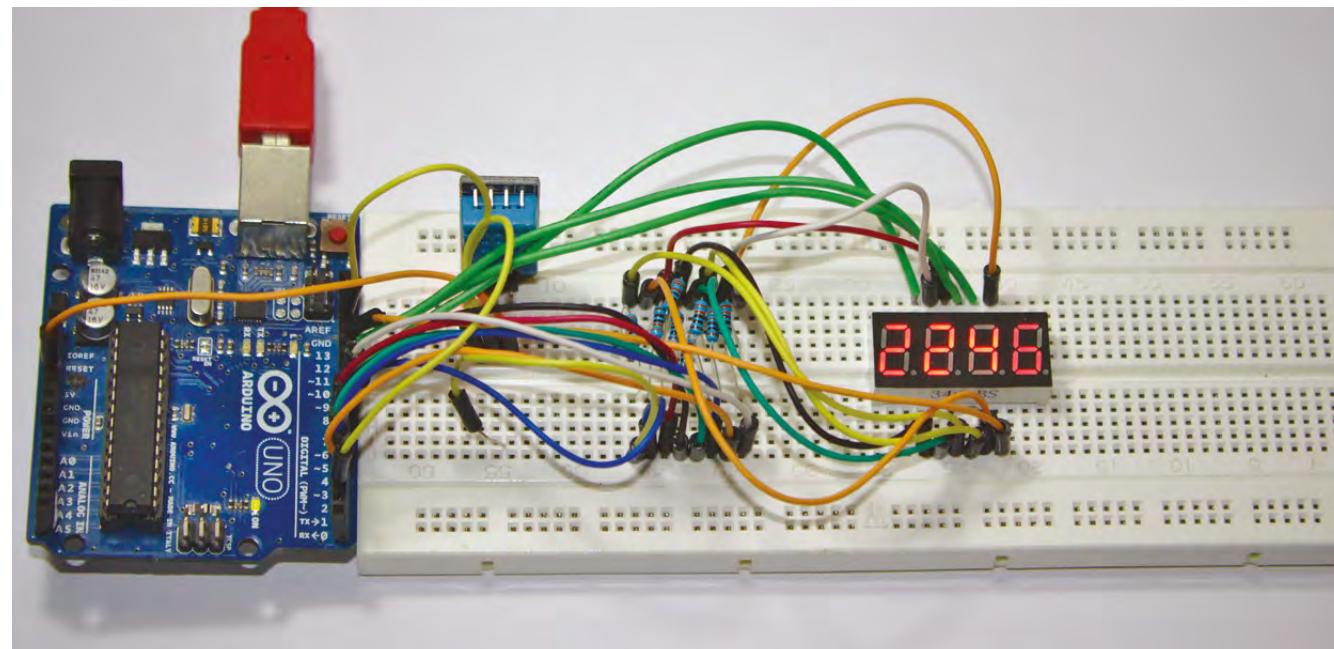
For the first time, rather than simply using the Arduino to manage a fancy set of LEDs, we're going to use it to measure something and then display the product of those measurements. And to do that,

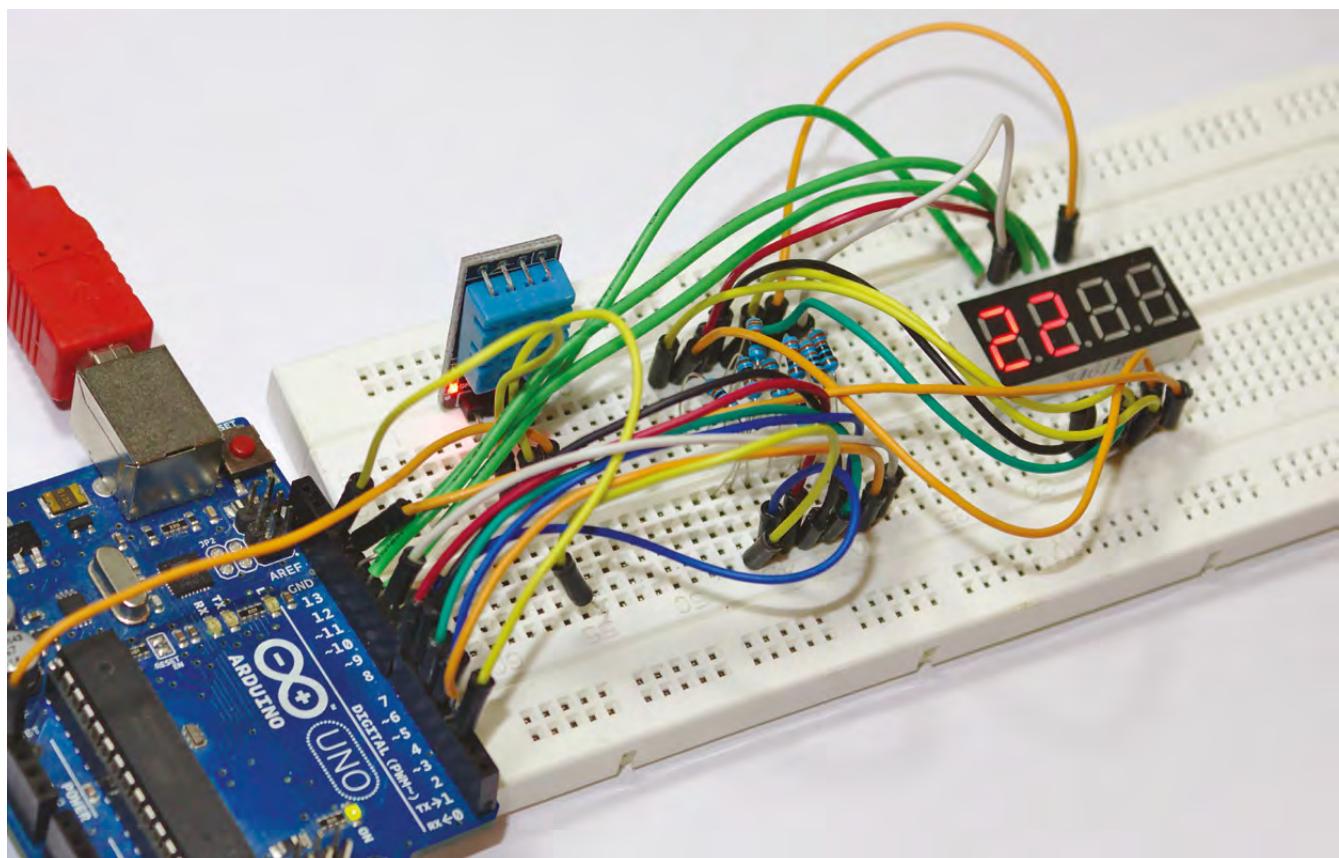
we're going to turn our seven-segment displays into both a temperature gauge and a humidity monitor. The hardware component we're building this project around is a DHT11 sensor module. These are cheap and readily available, and the concepts we use to interface one with the Arduino are almost universal.

The module itself combines both temperature and humidity sensors, and the great thing about this module is that it's incredibly easy to use. They're pre-calibrated, for instance, which means

Below ♦

The exact wiring will depend on the specification and pin configuration of your specific display





The module itself combines both temperature and humidity sensors, and the great thing about this module is that it's incredibly easy to use

we don't need to worry about the validity of the values we receive, and they send the values when asked correctly. Connecting one to your own projects couldn't be easier either, as there's only a single digital pin that needs to be connected to the Arduino. This pin handles all the communication between the modules and the Arduino, with the only other connections being 3V to 5V VCC for power and GND for a ground connection, both of which can be supplied by the Arduino. A four-pin variant is also available, but the extra pin can be ignored. The specification also recommends the use of a 5kΩ pull-up resistor connected to the data wire.

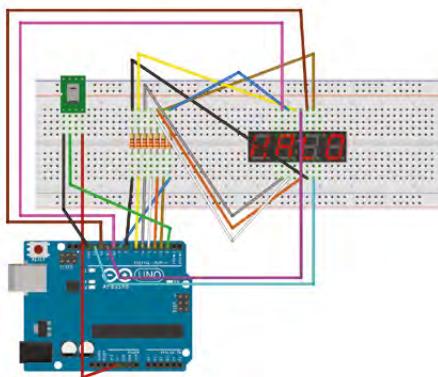
HARDWARE

The reason why we've chosen the DHT11 rather than one of its more capable siblings, such as the DHT22, is that the DHT11 returns only integer →

WIRING IT ALL UP

The data line to the sensor can be up to 20 metres long, which may be useful for garden readings!

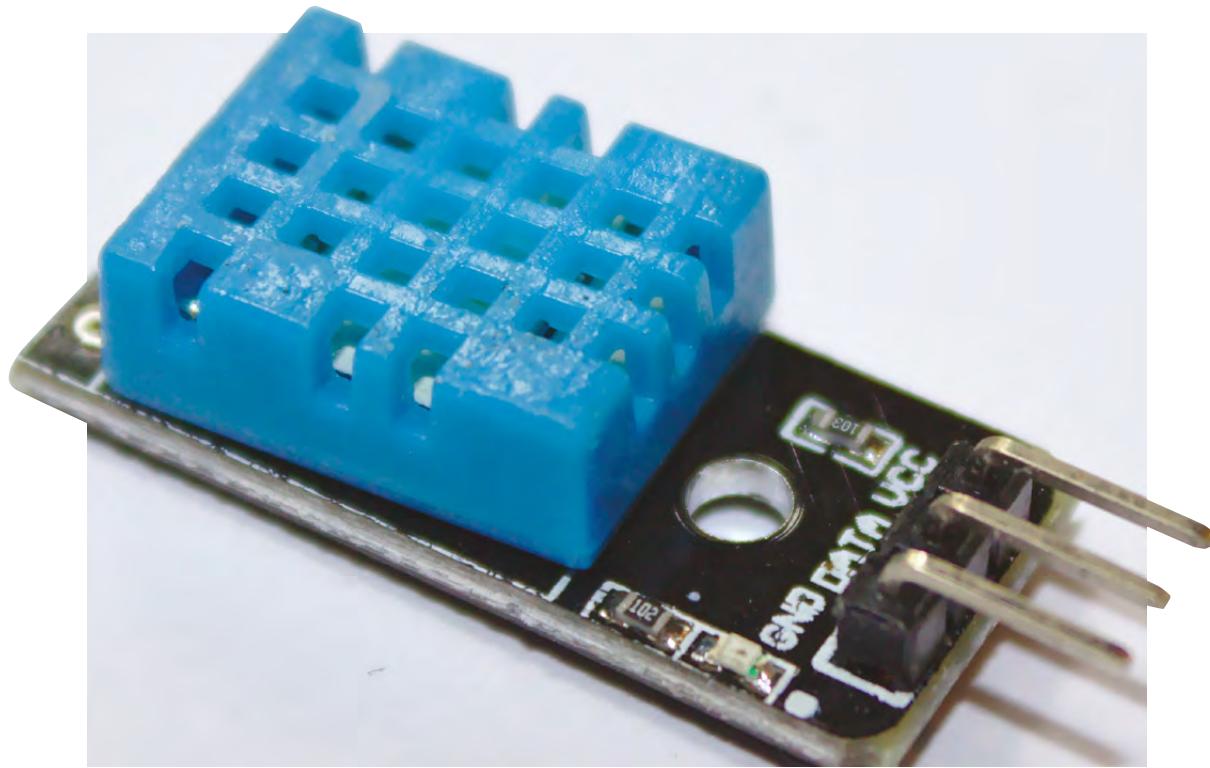
The circuit for this project builds on the one from the previous tutorial, adding a DHT11 temperature and humidity sensor. The three pins of this sensor need to connect to the 5V and GND provided by the Arduino, with the data pin connected to digital pin 2 on the Arduino. The specification for our sensor also needed a 5kΩ resistor between the 5V and data wires, with an optional 100nF capacitor between power and GND for power filtering.



Above

See the temperature and humidity changes in real time on your seven-segment display

SCHOOL OF MAKING



Right ♦

The DHT11 contains a calibrated temperature and humidity sensor, and can operate over 20 metres from your Arduino

temperature and humidity values. This would be limiting if you need a monitor with more precision, such as a temperature sensor attached to a beer fermenter, but it's accurate enough for our needs. In fact, as we're intending to connect the sensor output to four seven-segment displays, we only have four digits to play with – we'll put the temperature on one side and the humidity on the other. This makes the DHT11 perfect for our needs. But you can easily extend its capabilities by swapping out the sensor and just using the temperature value across four digits, or even setting up a line of LEDs to act as a bar graph for today's temperature.

We're now going to dive into the code to get this project to work. Normally, at this point, we'd need to deconstruct exactly what's happening with our circuit. In the previous tutorial, we covered multiplexing; for example, as a way to connect the multiple segments in the display with the limited number of digital pins on the Arduino. Similarly, we would ordinarily need to understand exactly how to communicate with the DHT11 and interpret any data the sensor sends back. This would itself be a complex job, even for a simple sensor like the DHT11. It uses a single pin – a 1-Wire data bus – to both receive signals and to send the data, which would require us to understand the protocol it uses.

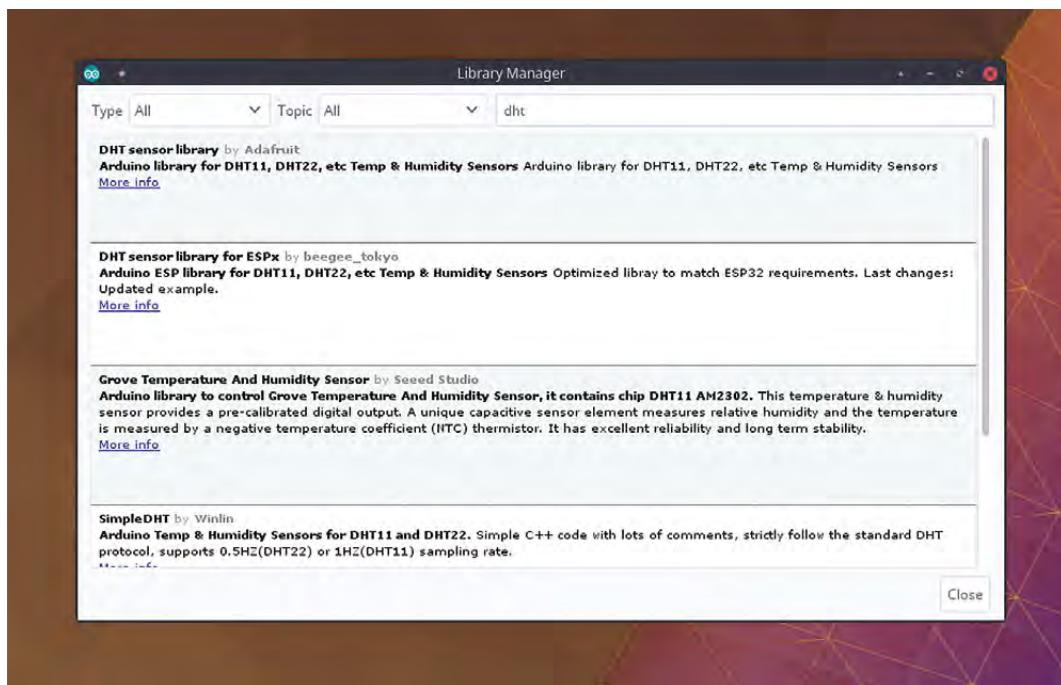
If you're lucky, the protocol is well defined and even supplied by the manufacturer, leaving you to implement the code in whatever way works best for you. But often, these protocols aren't documented and will need to be reverse-engineered, either through experimentation or by analysing the input and outputs of a sensor in a working configuration.

QUICK TIP

You will have come across libraries on your favourite operating system too, but these are typically the compiled object files generated from the source code that other tools can compile against to access their functions without having to be built against the library's source code.

Fortunately, most manufacturers of the DHT11 also provide a very informative datasheet

Fortunately, most manufacturers of the DHT11 also provide a very informative datasheet that not only covers the hardware specification and tolerances, but also the details of communicating with the sensor across the 1-Wire data bus. By reading this specification, you find out that the sensor needs a whole second with no signal to pass an initial 'unstable' status, and then you can send a signal to the bus lasting more than 18ms in order to instantiate a request. The response signal is a 40-bit



packet that contains both the relative humidity and the temperature. But we don't have to worry about any of this, thanks to what are known as libraries.

LIBRARIES

Previously, we've broken our code into functions that act as self-contained units that we call whenever convenient. We call function `displayNum` to show a number on the seven-segment display, for instance. We don't need to worry about how the LEDs are triggered, or how the numbers are displayed or sorted, or even how the delays are handled for multiplexing – we simply call the function with a single argument to pass the number we want shown. We could extrapolate this function into its own file by making sure that file contains all the information and variables it needs. We could then reuse the file in other projects, or share it with programmers who want the same functionality without wanting to constantly reinvent the wheel. You can see where this is going. C includes (hint) a way to import the contents of an external file so that you can access those external functions from within your code. And that's exactly what a library is. In fact, you add a library to your project using the special `#include` keyword, usually at the very top of your source file.

A library is usually a group of functions bundled with all the necessary definitions, structures, and variables to make those functions work as a self-contained piece of code. To keep these parts isolated from your own code, and to stop parts of your code falling into the scope of the library →

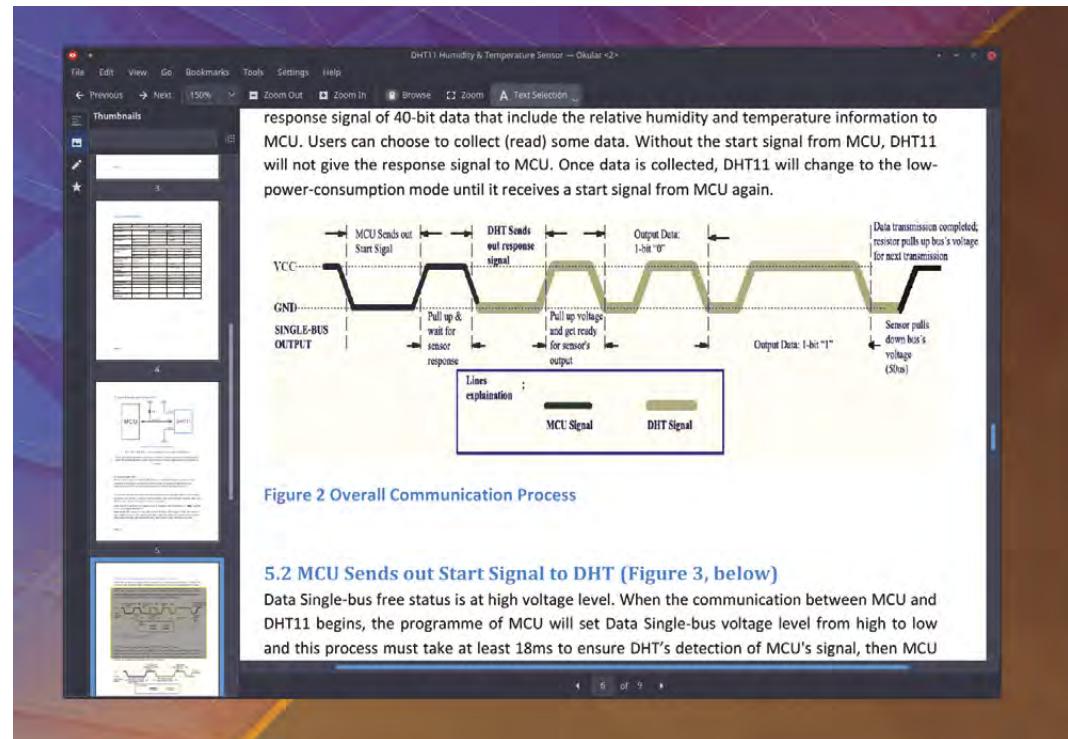
FLICKERING DISPLAY

One thing you may notice when running this code in this project, depending on your sensitivity, is that the seven-segment displays start to flicker. The cause of this is the processing and waiting delays of our code waiting for the data from the sensor, and it's an incredibly common problem. There's a direct trade-off in the number of jobs you ask your Arduino to perform and its ability to keep up a constant rate of updates with something like a display. This is why buffers were invented, so that the buffers can be filled in quiet periods and read from when the system is busy, and there are certainly designs that could update the displays from a buffer. But we can also do a lot in code, and while we'll look into more advanced options, such as using interrupts, in future tutorials, there is one area of our project that can be improved now, and it's the `delay()` call in the `displayDigit()` function. This delay was required to create enough persistence in the display for the characters to be easily visible, but as there's now more processing going on in the body of the code, the delay can be reduced. We've had best results by reducing this delay to 2 ms, so the code looks like the following:

```
void displayDigit(int digit, int number) {
    digitalWrite(digPin[digit], HIGH);
    for (int i = 0; i < 8; i++) {
        setSegment(segPin[i], segNum[number]
    }
    delay(2);
    digitalWrite(digPin[digit], LOW);
```

Left ♦
Libraries can be added manually or through the Arduino IDE. They're brilliant for making complex hardware easy to use

SCHOOL OF MAKING



Right ↗
The specification for the DHT11 includes analysis of the single-wire protocol. Checking this against the library implementation is a great way to learn new skills

and vice versa – a library is split into two files. The code that does the work is written in the '.cpp' file, analogous to the '.ino' files created by the Arduino IDE. But the part you import into your own project using the `#include` keyword is known as the interface, and is written in a 'header' file that has the '.h' suffix. It's called the header because the include command basically pastes its contents into wherever the command is implanted, which is nearly always in the header of your source file. The header itself doesn't include any functionality at all, but it does include template definitions for the structure and any functions and variables that you're going to use. This is how the compiler is taught about their existence and capabilities without including the functionality in your own code. The '.cpp' will also 'include' its own header file as it fleshes out the template with the implementation.

QUICK TIP

Another advantage of the 1-Wire data bus used by the DHT11, apart from cost, is that it can run across huge distances, with even 20 metres being feasible. Great for outside applications.

You can browse and download libraries automatically with the Arduino IDE. This feature is accessed by selecting Use Library > Manage Libraries from the Sketch menu, and you search for what you're looking for, such as DHT11, and click Install on the result. However, we feel it's worth doing this manually the first time so you can see how it works. Due to the popularity of the hardware, there are several libraries that make accessing the

DHT11 easy. The one we're going to use is called 'DHT Library', and it has the advantage of being compatible with the DHT11, 21, 22, 33, and 44 sensors, so you can upgrade the hardware in your project without having to change large chunks of the code logic.

Grab the latest `dht.cpp` and `dht.h` files from the GitHub repository: [git.io/vpudX](https://github.com/adafruit/DHT-sensor-library).

There are numerous ways to include these files in your project. You can, for example, create your own interface and implementation files and place them in the same directory as your project file. You can then use the `include` command with double quotes to add the library from the current location:

```
#include "dht.h"
```

The brilliant thing about using a library like this is that now have 'ourDHT' created via the definition in the library

The Arduino build environment will also look in the **libraries** folder just under where your projects live, and this is where you'll find any libraries installed via the Arduino IDE. This is also where we've put our downloaded **dht.cpp** and **dht.h** files, tucked away within a folder called **DHT**. As this location is part of the build environment path, you can include any libraries stored within the **libraries** system folder using the greater than/less than symbols around the library name, and this is what we're going to do with our project, adding the following to the end results of the code from our previous Arduino tutorial:

```
#include <dht.h>
```

OBJECTS

Just as you look at hardware specifications to understand how to use your components, you can use a header file to understand the capabilities of a library and how its features have been implemented. In particular, **dht.h** places almost everything within a 'class'. We've yet to discuss classes like these in our programming adventure, but we have discussed all the various components that go into them to make them useful. A class is a set of functions and variables all grouped together into something that operates a lot like its own type. Unlike a header file, a class is created to be directly assignable within your own code, allowing values to be set within its type and operations to be run against its state without the scope of your own code affecting that of the class. To add this type to our own project, we need to add the following:

```
dht ourDHT;
```

If you look at the header file for the library, you'll see the name given to the class is **dht**, which we use in our own code just as we might **int** or **float**. The brilliant thing about using a library like this is that now have **ourDHT** created via the definition in the library, we can very nearly start using our sensor. All that's needed, if you read the library documentation, is a **#define** statement to tell the class which pin we're using for the data line:

```
#define DHT11_PIN 2
```

As we've discussed previously, a 'define' statement is really just a global definition to replace the string with the value assigned to it, which made digital pin 2 on the Arduino Uno. That definition will percolate into the functions with the class so that everything works correctly. If you've got a brilliant memory, you'll have already noticed that the same

pin is already being used to drive our seven-segment display. In fact, we only have one pin free, and that's pin 9. We could simply attach the DHT11 data pin to this and change the define, but we found it easier to move the wire from pin 2 to pin 9 of the seven-segment display, followed by updating our array of pins to reflect this change:

```
const byte segPin[8] = {9, 3, 4, 5, 6, 7, 8};
```

A class is a set of
functions and variables
all grouped together into
something that operates a
lot like its own type

We now get to the part where we deal with all the complexity of working out temperature variations and connection protocols. Except we don't. All we need to do is wait for a ready signal from the sensor before reading the temperature and humidity values from the class that's dealing with all the complexity on our behalf. To get the temperature, for example, you could simply use **float newtemp = ourDHT.temperature**; . The dot after our class name means that **temperature** is a member of the class, as described by the header. We don't need to worry about how this value was placed in **ourDHT**. **temperature**, just that it was assigned to **temperature** which we're now assigning to **newtemp**. That's what's so brilliant about using libraries. But we don't even need to do this, because if we multiply the temperature by 100 to move it left by two digits, and then add this to the humidity reading, we can perform the entire step in the same command that sends the values to the display.

This means our entire **loop()** function need only be two lines long:

```
void loop()
{
    int chk = ourDHT.read11(DHT11_PIN);
    displayNum((ourDHT.temperature*100)+ourDHT.
humidity);
}
```

And that's it. As usual, and for brevity, we've omitted any code to error-check the sensor, but this should really be added as homework. Otherwise, send the code to your Arduino and we're done. You can download the updated source for this project from git.io/vpzvg.

QUICK TIP

A class will be composed of public and private elements. As their names suggest, public elements can be manipulated by your code, while private elements are intended only for the internal workings of the class.

Getting retro with the Z80: hardware

Relive the glory days of homebrew computing



Dave Astels

@dastels

Dave's career started in the 8-bit days, with the Z80 and 6502, and he's been working with computers ever since. He does some writing at daveastels.com and learn.adafruit.com.

A

Are you curious about retro 8-bit embedded computing? Curious about what goes on inside that tiny microcontroller chip on your Arduino board?

In this article we'll have a look under the hood by building a system around the Z80 (**Figure 1**), one of the most popular 8-bit CPUs.

The Z80 is a Central Processing Unit (CPU), not a Microcontroller Unit (MCU). That means it's not a complete system; it doesn't contain any memory or peripherals. MCUs, on the other hand, combine a CPU with memory (flash, RAM, and often EEPROM) and various peripherals on one piece of silicon. For example, the ATSAMD21 series of MCUs (used in Arduino Zero and Adafruit's 'M0' branded boards)

contains a 32-bit ARM Cortex-M0+ CPU, and the ATmega328 (used on the Arduino Uno, pro-mini, etc.) contains an 8-bit AVR CPU.

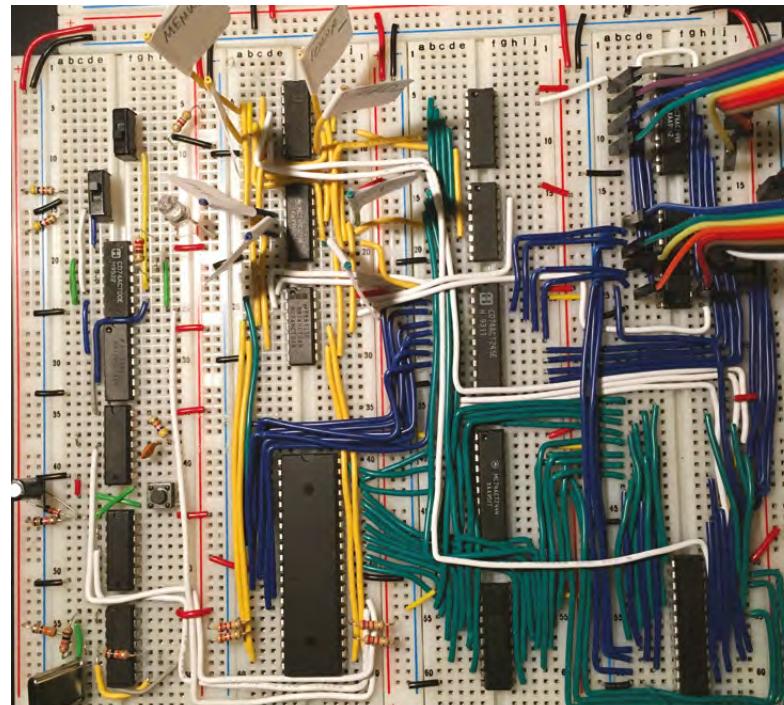
The Z80 was one of the more advanced 8-bit CPUs of the time (late-1970s to mid-1980s). It was released by Zilog in mid-1976 and had the brilliant feature that it could run code written for Intel's 8080 CPU without change. Up until this time, the 8080 was the most popular CPU. The Z80 would be uninteresting if it were just an 8080 clone, but it added many improvements and requires simpler support hardware. If you've ever played original arcade versions of Pac-Man, Frogger, Galaxian, and the like, you've benefited from the Z80's popularity in the video arcade industry.

Externally, the Z80 was very similar to the other CPUs of the time. As such, it had four types of connections:

1. CPU control and support such as clock, interrupts, and reset.
2. System control including read/write signals, memory and I/O selection; signals that tell the rest of the system what the Z80 is doing.
3. The address bus, which the Z80 uses to select the memory location or I/O port it wants to communicate with.
4. The data bus, which is used to transfer data between the Z80 and memory or I/O.

The practical effect of the Z80 being a CPU, as opposed to an MCU, is that we can't do anything with just a Z80 chip: we need to add a non-trivial amount of other circuitry. As with any CPU, there are five types of external circuitry needed:

1. What we call housekeeping: reset pulse generator, clock generator, etc.



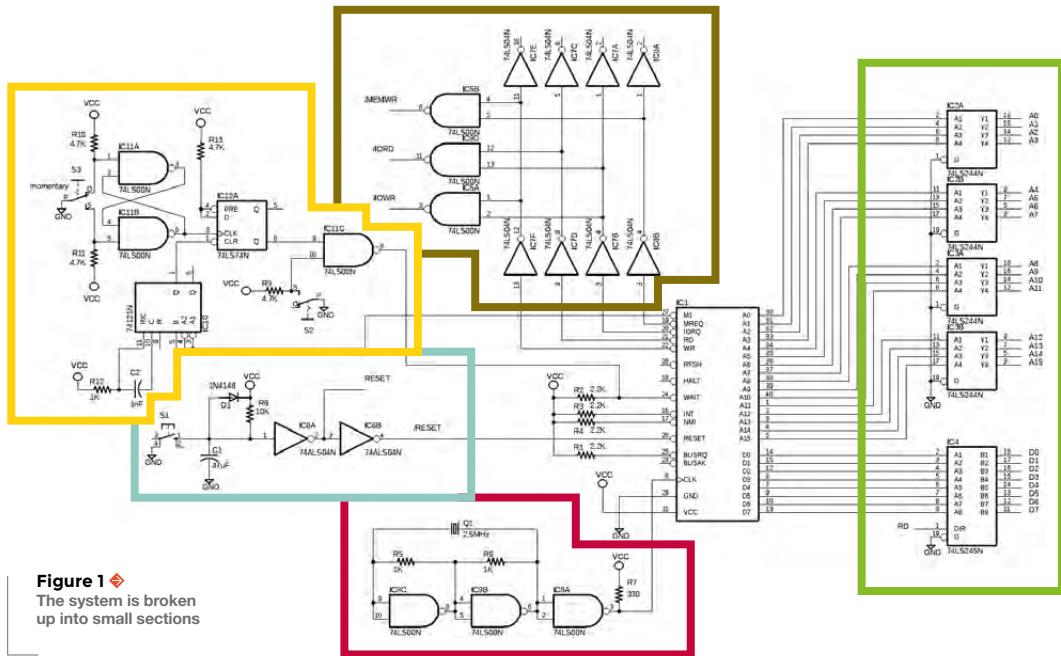


Figure 1 ◊
The system is broken up into small sections

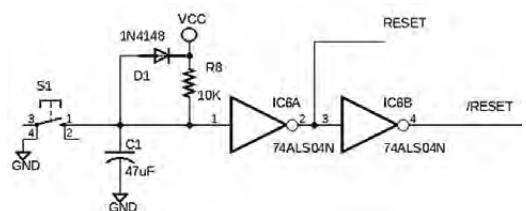
2. Buffers that let you connect multiple circuits to the buses.
3. Addressing logic that selects the various memory and I/O circuits based on the address being requested.
4. Memory, which will be some mix of read-only and read-write.
5. I/O interfaces of various forms.

Let's take a look at how this all works.

RESET

When the Z80 receives a logic low on its reset pin, it stops whatever it's doing, resets its internal state, and pauses until its reset pin returns to a high level. Then it starts executing instructions at the address 0x0000. The code stored there should reset various aspects of the system and put it in a usable state. There are

Figure 2 ◊
The reset circuitry brings the system back into a usable state when things go wrong



two times that we want this to happen. The first is when you power up the system; it should come up in a reasonable, and ultimately usable, way. The other time is when things go horribly wrong and you want to get the system back under control and running properly. **Figure 2** shows a reasonable reset circuit that provides for both use cases.

This reset circuit is relatively straightforward. C1 takes a brief bit of time to charge when the power is switched on. While it's charging, the RESET line is high, and /RESET is low. Once it reaches a charge equivalent to a logic high, RESET changes to low, and /RESET to high. The system will then start running at address 0x0000. That will take care of resetting the system on power up. Later, if S1 (a momentary push-button) is pushed, C1 is emptied of charge, and we are in the same state as when power was initially switched on. When S1 is released, C1 charges again as before. →

EPROMS

These days, MCUs typically have flash memory and RAM. Back in the day, when you were prototyping (or being a maker before it was cool), you used EPROM (erasable programmable read-only memory). Flash makes reprogramming easy and is a welcome change. EPROMs needed special hardware to program, and a UV 'eraser' that's like a mini tanning bed for memory chips; but imagine the chaos if you forgot everything each time you went to a tanning salon. You can identify EPROMS by the clear window in the top of the chip that lets the UV light in.

YOU'LL NEED

- ◆ 1 × S1 momentary SPST push-button
- ◆ 1 × S2 SPDT slide or toggle switch
- ◆ 1 × S3 momentary SPDT push-button
- ◆ 2 × 1×8 header JPI, JP2 pin header
- ◆ 1 × Z80 IC1 CPU
- ◆ 3 × 74LS244N IC2, IC3, IC17 octal buffer and line driver, 3-state
- ◆ 1 × 74LS245N IC4 octal bus transceiver, 3-state
- ◆ 3 × 74LS00N IC5, IC9, IC11 quad 2-input NAND gate
- ◆ 2 × 74ALS04N IC6, IC7, IC8, IC19 hex inverter
- ◆ 1 × 7412IN IC10 monostable multivibrator
- ◆ 1 × 74LS74N IC12 dual D-type flip-flop
- ◆ 1 × 6116P IC13 RAM memory
- ◆ 2 × 74LS138N IC14, IC15 3-line to 8-line decoder/demultiplexer
- ◆ 1 × 28C16AP IC16 EEPROM 16 K (2 K × 8)
- ◆ 1 × 74LS373N IC18 octal D type transparent latch
- ◆ 4 × 2.2 kΩ R1, R2, R3, R4 resistor
- ◆ 3 × 1 kΩ R5, R6, R12 resistor
- ◆ 1 × 330 Ω R7 resistor
- ◆ 1 × 10 kΩ R8 resistor
- ◆ 4 × 4.7 kΩ R9, R10, R11, R13 resistor
- ◆ 1 × 1N4148 D1 diode
- ◆ 1 × 2.5MHz Q1 crystal
- ◆ 1 × 47 μF C1 capacitor
- ◆ 1 × 1 nF C2 capacitor

Getting retro with the Z80: hardware

SCHOOL OF MAKING

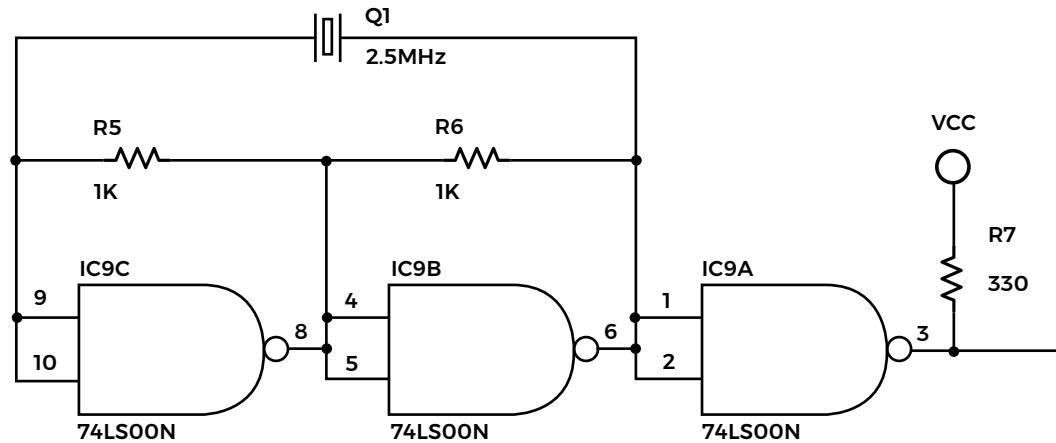


Figure 3 The clock keeps everything progressing in the right order

QUICK TIP

We denote an active low signal by prefixing the name with a slash.

CLOCK

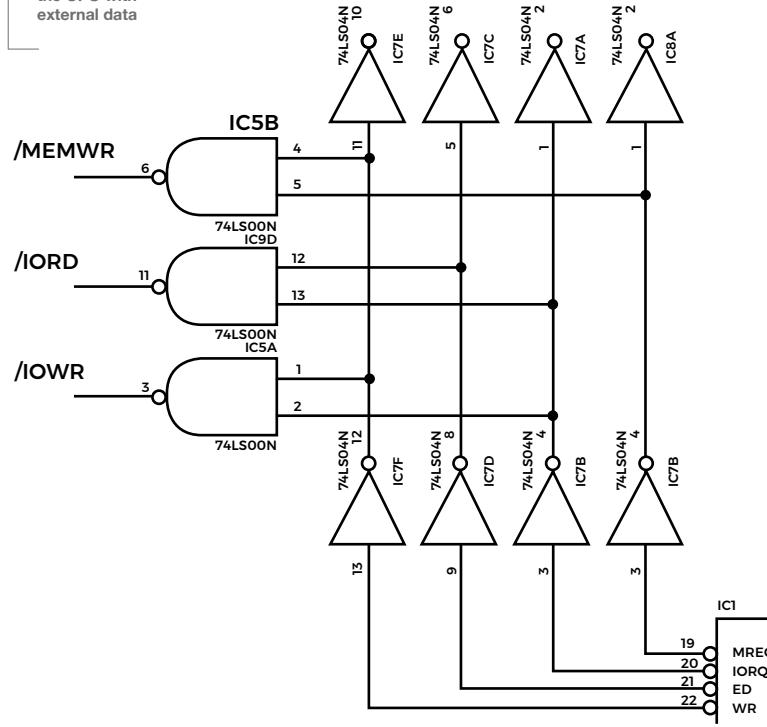
Every CPU needs a clock that steps it through its various states; e.g. fetch an instruction, decode it, load data, perform the appropriate operation, store the result, etc. All CPUs, and most MCUs, require an external clock generator, typically crystal-based. Using a crystal-based oscillator provides a stable signal that regularly alternates between high and low. The Z80 makes this easy because it requires a simple, single-phase clock. Some processors need multi-phase clocks that require somewhat more circuitry to generate. **Figure 3** shows a typical clock generator circuit.

MISC

There is always a bit of miscellaneous control circuitry to make interfacing with devices easier.

Figure 4 shows some standard circuitry of this sort used to combine interfacing signals that select I/O or memory and those that select read and write. We combine them to have the signals read-memory, write-memory, read-input-port, write-output-port.

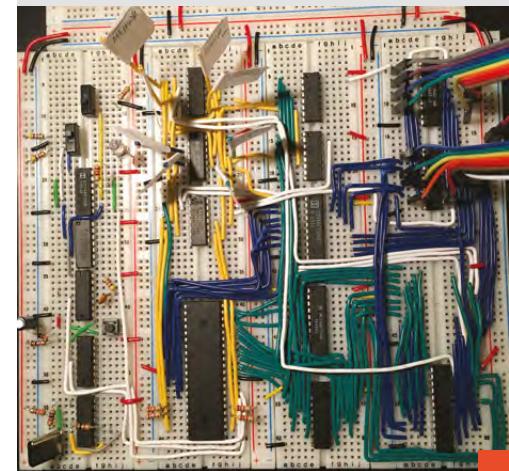
Figure 4 A bit of circuitry needed to interface the CPU with external data

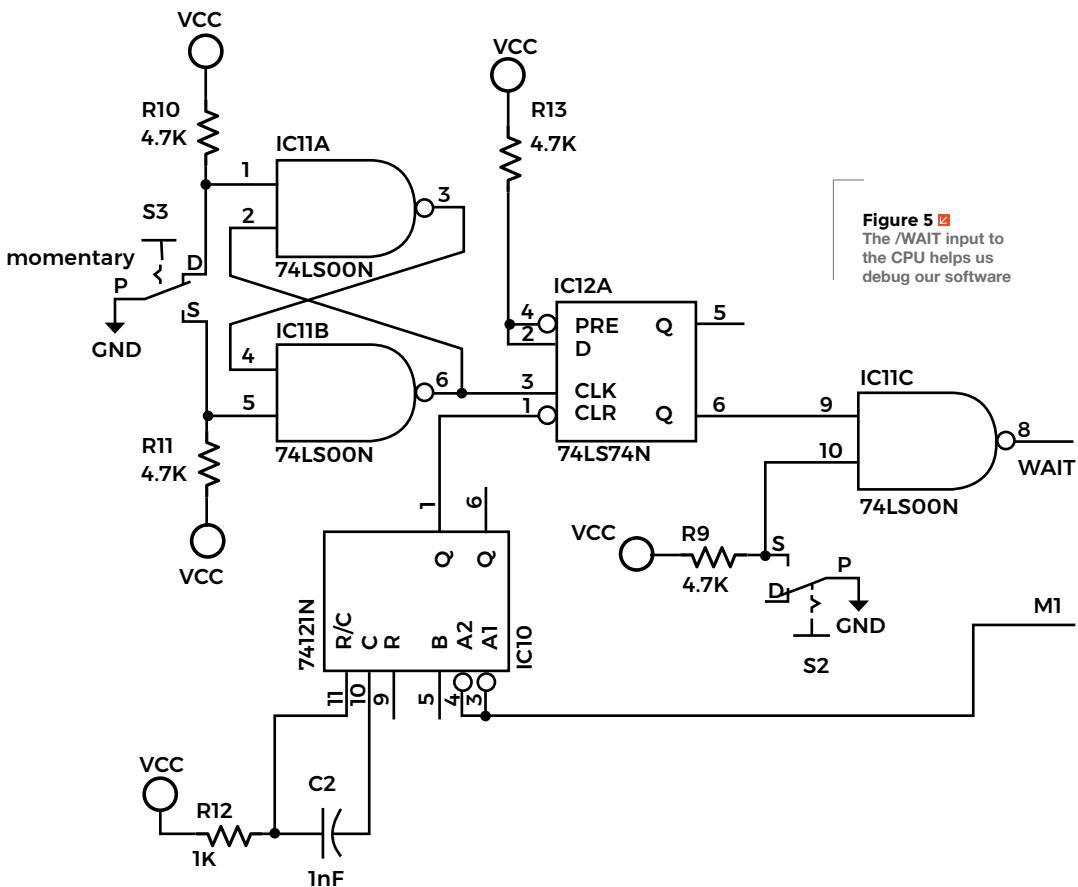


BREADBOARD

You can breadboard this design fairly easily. Laying things out logically and keeping the wiring tidy makes it straightforward even though it's not a trivial circuit. See the image below with the functional areas clearly labelled. Another thing that helps is using different colours for the different signal types. Generally:

- GREEN – address bus
- BLUE – data bus
- YELLOW – CPU control signals
- WHITE – system control signals
- RED – Vcc
- BLACK – ground





MEMORY

Figure 5 The /WAIT input to the CPU helps us debug our software

We use static RAM in this design because it's much simpler. Dynamic RAM has to be refreshed regularly or its contents disappear. Extra circuitry is required to make this happen (although the Z80 has special hardware support that makes this much easier). Dynamic RAM chips hold more than static RAM chips in general and are much lower priced. If using more than a few static RAM chips can handle, you're better off going with dynamic RAM. In fact, we rarely see static RAM being used now.

SINGLE STEPPING

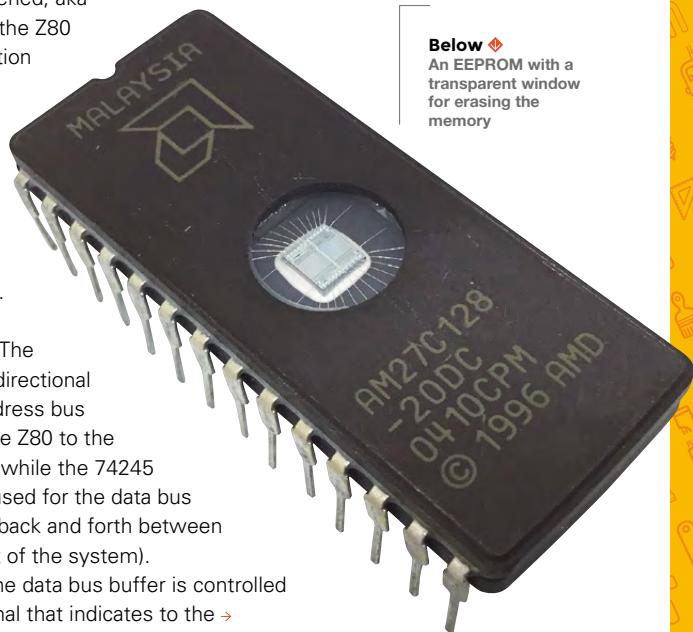
One thing that's handy to be able to do when debugging is single stepping through code. You might have done this in C or Python, but it takes on a whole new level of meaning at the hardware level. With the circuit shown in **Figure 5**, we can have the Z80 stop just before executing each instruction. This gives us an opportunity to check various signals and logic levels in the circuit. The toggle/slide switch S2 lets us turn single stepping on/off and the momentary push-button S3 advances the Z80 by one instruction. One key to this circuit is the /M1 output from the Z80 that indicates when it's starting the processing of an instruction by switching to a low state. The other key is manipulation of the /WAIT input to the Z80, which causes the CPU to sit and wait when it goes low until it goes high again. /M1 going low causes the 74121 to generate a short pulse that clears the 7474 flip-flop. Its negated output (now high) is gated by a NAND gate, controlled by S2. If S2 is open, the Z80 is asked to wait. When S3 is pushed, the output of the flip-flop goes low, causing the Z80's WAIT input to go high again so that it can continue processing the current instruction. When it starts the next instruction, the process begins again with /M1 going low. This is probably the most complex part of the design.

BUFFERING

Each output of the Z80 can support very little load, i.e. only a few connections. To add addressing decoders, memory, and I/O circuits, they need to be strengthened, aka 'buffered'. This lets the Z80 see a single connection while providing the ability to drive many. This bit of the design is very simple, as there are chips designed for exactly this purpose.

Figure 6 shows the basic configuration. The 74244 chips are unidirectional and used for the address bus (which goes from the Z80 to the rest of the system), while the 74245 is bidirectional and used for the data bus (used to move data back and forth between the Z80 and the rest of the system).

The direction of the data bus buffer is controlled by the Z80's RD signal that indicates to the →



Below An EEPROM with a transparent window for erasing the memory

Getting retro with the Z80: hardware

SCHOOL OF MAKING

Figure 6 ◆
Buffers let us drive external chips with more power than the CPU can provide

Figure 7 ◆
A decoding circuit links the input and output to the CPU

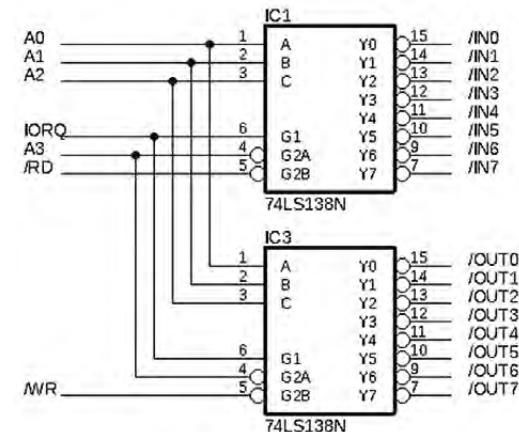
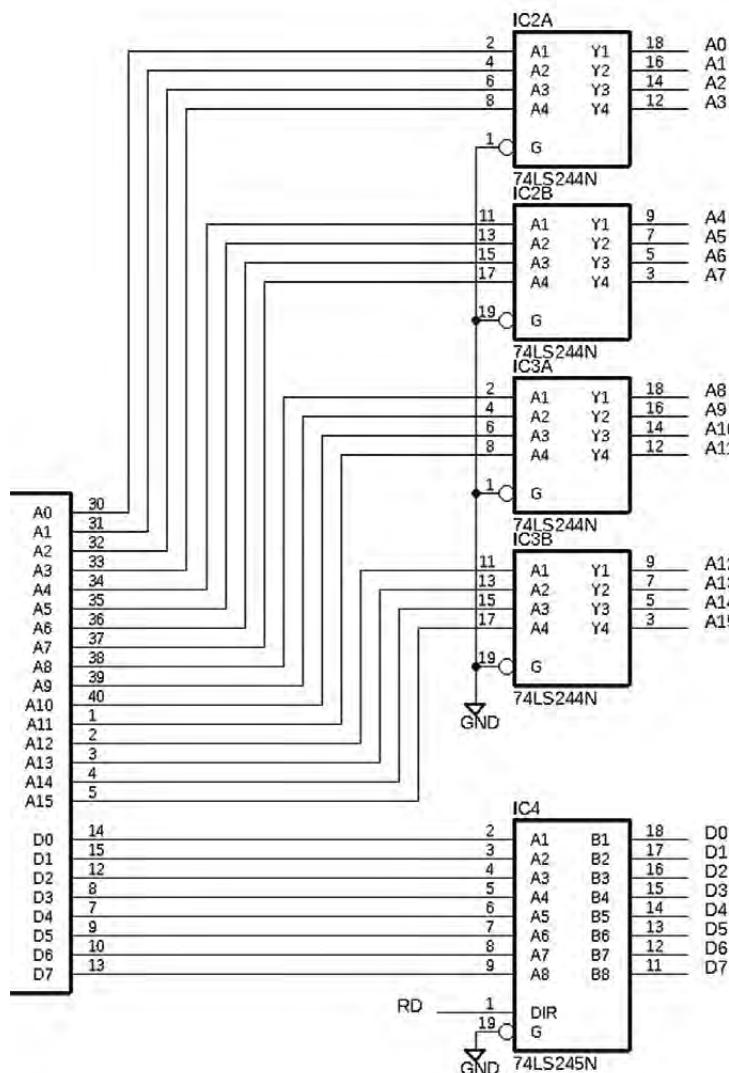
system that it wants to read from whatever is being addressed.

All the buffer chips are always enabled since this is a very simple design. If we were using DMA, we would have to disable the buffers while it was using the buses. Thankfully, the Z80 has built-in support for that, but we won't be exploring it.

In a system this simple, we could probably get away without buffering, but it's better to be safe than sorry. And it is so commonly required, it's worth seeing how it's done.

INPUT AND OUTPUT

Many CPUs have only one address space that contains any and all memory as well as I/O devices.



One of the advanced features of the Z80 is that it has separate address spaces for memory, input, and output. It supports 256 input and 256 output ports through specialised I/O instructions. Our design is kept simple by supporting only eight of each, specifically 0x00–0x07. The decoding circuit, **Figure 7**, uses the lowest three address lines (which can represent numbers 0–7) to activate one of eight port select signals. The I/O Request (IORQ) signal from the Z80 is used as one of the enable inputs to these decoders, while the read (RD) and write (WR) signals enable the input and output decoders, respectively. The address line A3 is also used to enable both decoders when it is low. This is a pretty much free trick that leaves the design open to adding eight more input and output ports if required in the future (i.e. the cases when A3 is high, e.g. port addresses 8–15).

We can imagine all sorts of input and output devices from switches and LEDs up to display and keyboard, or even disk drives and Ethernet. Once more, we're keeping things simple and limiting ourselves to the simplest devices: an 8-bit digital input port and an 8-bit digital output port. In fact, when we look at software for the Z80, we will connect these to some switches and LEDs. **Figure 8** shows the I/O port circuitry. We're using another 74244 to read the logic signals on JP1 at the time we access it. Notice how the /IN0 signal from the decoding circuit is used to have the 74244 place its inputs on the data bus for the Z80 to read. The output port uses a 74373 latch to grab the signals on the data bus when the /OUT0 signal is active. Those latched values are then present on the 74373's output until it is written to again.

MEMORY

The Z80 directly supports up to 64kB of memory. This almost always consists of some mix of read-

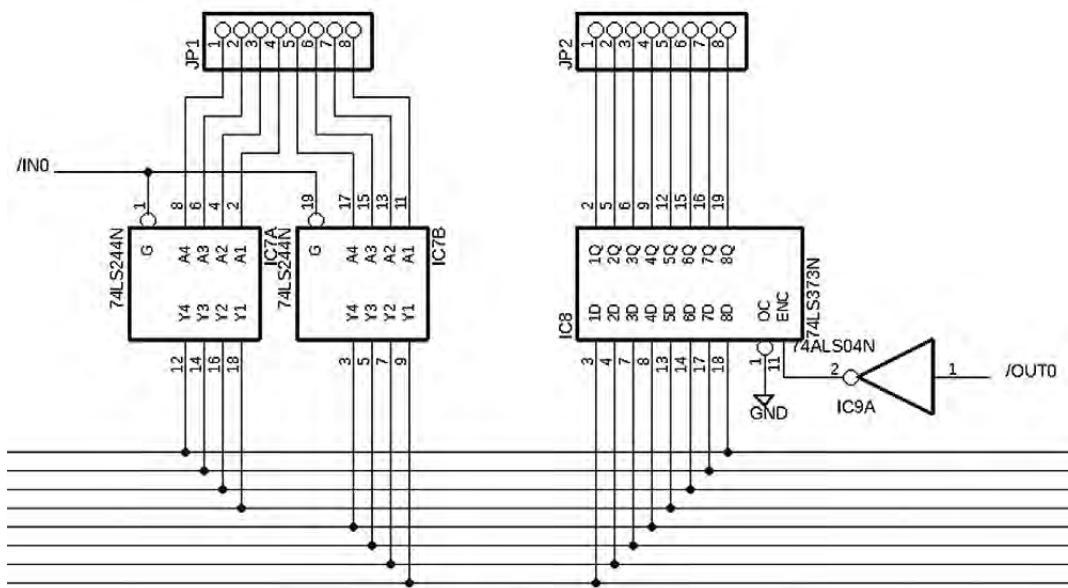


Figure 8 ◇
The ports' circuitry links our CPU safely to the external world

QUICK TIP

CPU signals can't drive much, they need to be buffered.

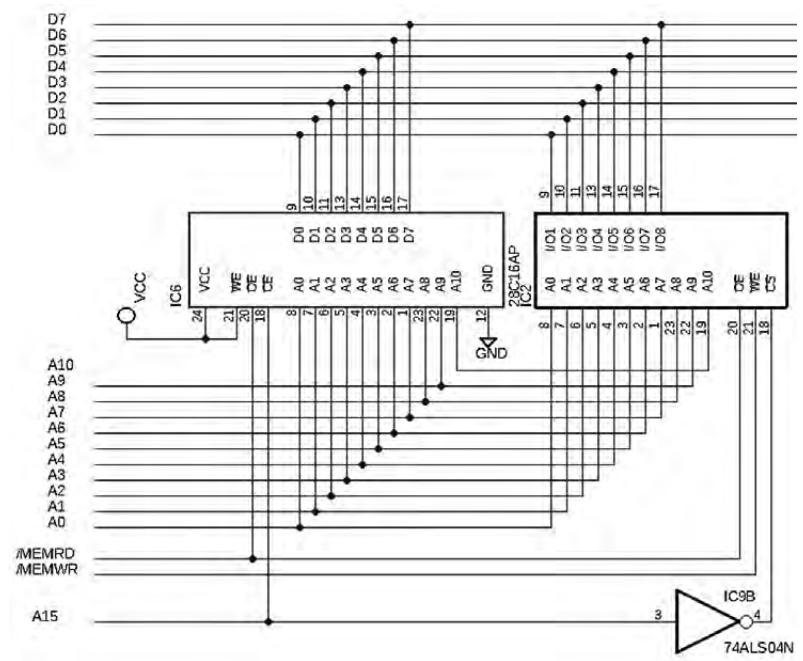
only (ROM, EPROM, and such) and read/write (RAM) memory. Memory chips come in various sizes; during the heyday of the Z80, 2kB was a common size for EPROM and static RAM. For the sample design used for this article, we have 2kB of ROM and 2kB of RAM (**Figure 9**). Normally in a fully designed system, we would have decoding built in (see the section above on I/O for an example decoding circuit) for most, if not all, 32 2kB blocks. Since we will only be supporting two blocks (one for EPROM and one for RAM), we can take a short cut with the decoding. EPROM has to start at address 0x0000 because that's where the Z80 goes when it's reset. If we put the RAM at address 0x8000, we can use the highest address line (A15) to select between them – costing us only a single inverter. The 'catch' is that the 2kB EPROM block is replicated throughout the lower 32kB of the

address space, and the 2kB RAM block throughout the upper 32kB. This doesn't pose a problem, and can be abused for benefit in some situations. The other downside is that additional memory can not be added without rebuilding the memory circuitry.

That's it. A basic Z80 system is pretty straightforward: CPU, RAM, ROM, and just enough I/O to do the job. Of course, we can add more memory and different types of I/O. It's the I/O that really differentiates embedded systems.

Join us next month as we take a look at how to program your homemade computer. □

Figure 9 ◻
The memory chips give us a little RAM to load our software into



STANDING ON SHOULDERS

Most of what I know about working at a hardware level with the Z80 (and computer hardware in general), I owe to Steve Ciarcia, who wrote a regular column in *Byte* magazine through the 1980s when it was a serious maker resource. Specifically, he wrote a book titled *Build Your Own Z80 Computer*. Many of the ideas in this design come from, or are inspired by, the material in that book. Originally only available in hard copy (it was 1981, after all), it has been scanned and released as a freely available PDF at hsmag.cc/PGandh, and other places online.

If you want to play around with the Z80, but don't want to build your own system, have a look at the RC2014 project on Tindie.

Laser-cut a 360° GIF turntable

Create stunning 360° animated GIFs easily with this geared turntable



Alex Eames

RasPiTV

Alex Eames loves making things and regularly blogs/vlogs at RasPi.TV. He makes a living designing and selling RasP.iO products.

A360° animated GIF (or video) is an excellent way to showcase your latest make or product. It will help you to provide a professional-looking all-round view. You can make your own with free software and around £10 worth of materials.

The mechanism in this plan suits turntables up to 40cm in diameter, but you could quite easily scale it to make whatever size you want. Angular resolution could be further increased by adding a third reduction stage.

To make a nice 360° animated GIF you need a minimum of 16 frames, but 32 is better. More frames results in a bigger GIF file, but a more realistic, less jumpy animation. To be able to make 1/16th of a rotation in an accurate, controllable, repeatable way, you'll use gears so that one turn of the crank handle

gives 1/16th turn on the turntable. (You can do 1/32nds with half-turns of the crank.)

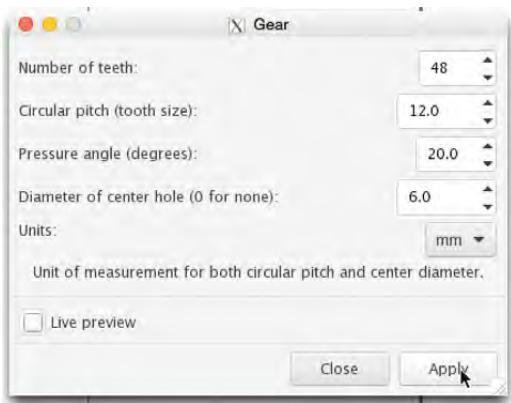
Designing gears is fairly simple with software. The good news is that you don't need to know any mechanical engineering theory to do it.

Inkscape is a free and open-source vector graphics package offering masses of features and is available on Linux, Windows, and macOS. It also works on the Raspberry Pi and is widely used by people preparing SVG files for laser cutting. Even better, it has a built-in extension for designing gears. If you haven't got Inkscape yet, you can find it at inkscape.org.

Below ↴

Fully assembled 360° GIF turntable, ready for use. View the animated GIF output from this photo session at: hsmag.cc/OzMEEJ



**Above**

Designing gears in Inkscape is as simple as selecting four parameters and choosing your units of measurement

To access the Inkscape gear designer, from the menu, choose...

Extensions > Render > Gear > Gear

Four design parameters determine the gear's characteristics:

- Number of teeth
- Circular pitch
- Pressure angle
- Diameter of centre hole

Below

Gears all assembled on their three axles. Note the handle on axle 1 and protrusion to hold turntable on axle 3

The number of teeth and tooth size (circular pitch) determine the overall size of the gear. You can work in pixels, mm, or inches. This project uses mm.

PROJECT CONSTRAINTS

12 mm circular pitch gives a suitable tooth size for 6mm ply, which our gears are made of. At this pitch size, a bit of experimentation shows that the minimum number of teeth for a gear to mesh satisfactorily with a much larger gear is 12 (6, 8, and 10 were all tried and found unsatisfactory.) The largest gear, at this pitch size, that fits comfortably on an A4 sheet of 6mm ply has 48 teeth.

The aim is to create an overall gearing of 1:16, so that one turn of the handle gives 1/16th of a rotation of the turntable. You will achieve this in two stages of 12:48 (1:4), using two 12-toothed gears and two 48-toothed gears.

Four turns of the 12-toothed gear on axle 1 rotates the 48-toothed gear on axle 2 once. Axle 2 also has its own 12-toothed gear. Four turns of this rotates the 48-toothed gear on axle 3 once. The turntable sits on spacers above this final gear, and rotates once for every 16 turns of the crank handle.

GEAR UP

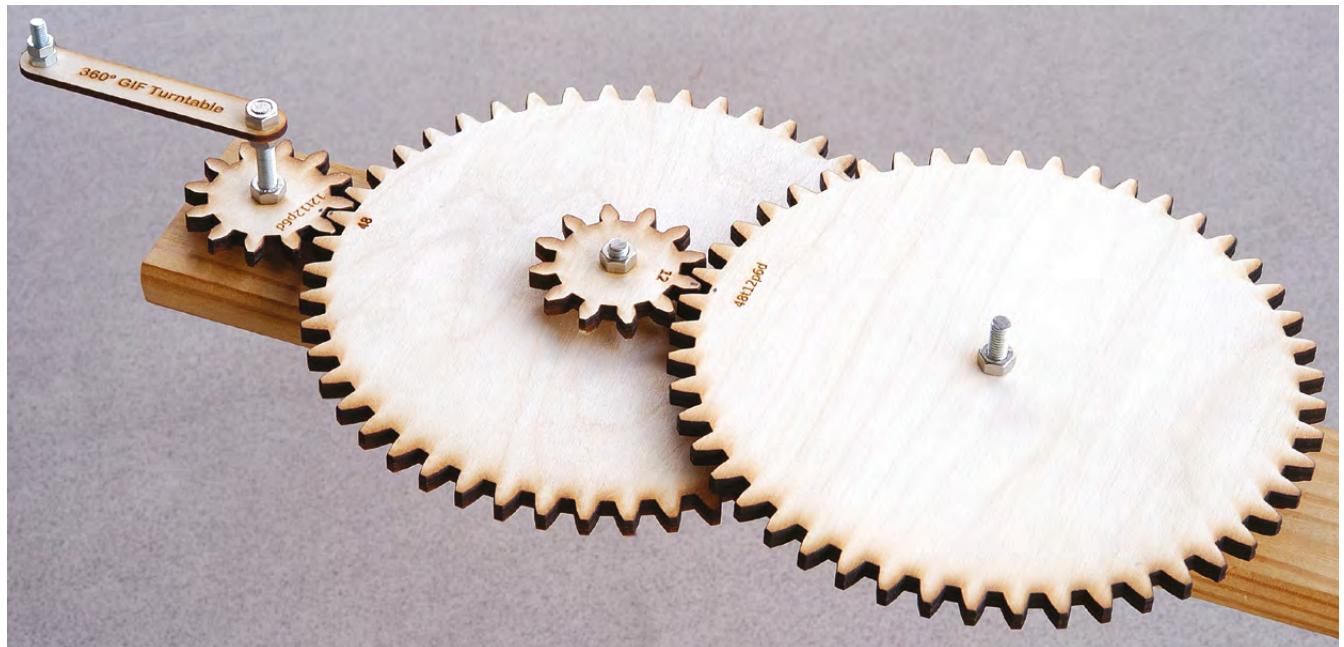
Either download the SVG from hsmag.cc/Xcwhir, or create your gears in Inkscape using the following parameters. You'll need two of each gear and one bar for the crank handle. →

YOU'LL NEED

- ◆ 2 x A4 sheet 6 mm birch laser ply
- ◆ 500 x 70 x 18 mm length of wood
- ◆ A4 sheet of 9 mm birch ply
- ◆ A4 sheet of white sticky-back plastic
- ◆ 30 cm M6 threaded rod
- ◆ 20 M6 nuts

TOOLS

- ◆ Clamp (to hold turntable to a surface while in use)
- ◆ Drill and 6 mm bit
- ◆ Access to laser cutter (can cut gears manually, but laser is much preferred)
- ◆ 10 mm spanner and adjustable spanner



SCHOOL OF MAKING

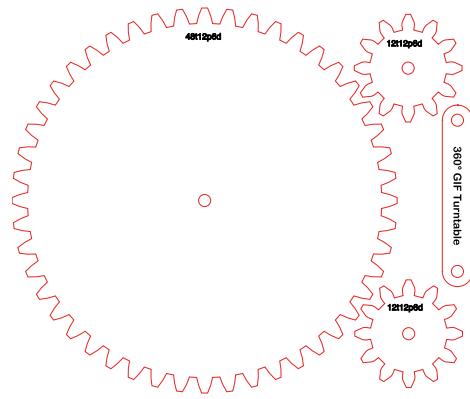
QUICK TIP

When designing gears, it can be useful to mark the parameters on each gear e.g. for 48 teeth, 12 mm pitch, 6 mm diameter: 48t12p6d

PROJECT FILES

The project files are all in the GitHub repository at hsmag.cc/Xcwhir, as are some output example GIFs made using this turntable.

If you don't want to design your own gears, you can grab the editable SVG file [GIF-turntable-12-48.svg](#) from there. You'll need two of each gear and one handlebar. It is possible to cut gears by hand, but it's very labour-intensive and intricate. You'd need a band-saw, scroll-saw, or coping saw and it would take a long time. Laser cutting is greatly preferred if available.



Right You'll need two of each gear size and one handlebar

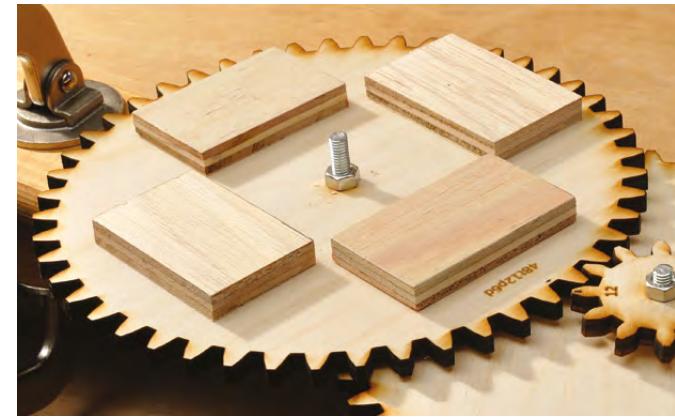
Below Side view showing vertical orientation of the gears. The large gear on axle 3 rests on three M6 nuts

48 teeth, circular pitch 12.0, pressure angle 20.0, diameter of centre hole 6.0, units mm.

12 teeth, circular pitch 12.0, pressure angle 20.0, diameter of centre hole 6.0, units mm.

For the base, take a piece of wood approximately $50 \times 7 \times 1.8$ cm and drill three 6 mm holes, starting at 3cm from one end and with 11.6 cm between centres. Make these as perfectly perpendicular as possible. These will position our three M6 threaded rod axles so that the gears mesh nicely.

Each of the axles protrudes through the bottom of the base by ~11 mm, which is long enough to attach two M6 nuts.



Above Boost turntable height and keep it flat and level with four plywood spacers

Axle 1 (~7.2 cm long) bears one 12-toothed gear and the crank handle bar. The gear is held in place with two M6 nuts, as is the bar. Using M6 nuts, loosely attach a small length of threaded rod to the outer hole on the handle bar so it rotates freely.

Axle 2 (~5.6 cm long) bears a 48-toothed gear, held in place between two M6 nuts, and a 12-toothed gear above it, held by another M6 nut. The nuts need to be tight enough to prevent the gears spinning freely on the axle. They also act as spacers to minimise friction.

Axle 3 (~6.4 cm long) bears a single 48-toothed gear, resting on three M6 nuts, raising it to the correct height to mesh with the 12-toothed gear on axle 2. This axle protrudes approximately 17 mm above the gear so that the turntable can be fitted to it.

Assemble all three axles and attach them to the base. Check that all the gears are meshing nicely and adjust axle tension as necessary (using the bottom two nuts).

Using a compass, mark and cut a 20 cm diameter circle from 9mm ply and drill a 6 mm hole in the centre. Cut four 6 x 4 cm pieces of 9mm ply to act as spacers for the turntable. Cover the circle with white sticky plastic film. This will give a nice clean, white background for your photographs.



Place the four plywood spacers on top of the 48-toothed gear on axle 3, and place the turntable on top of those. You don't need to fix the spacers in place, but you can if you want to.

GET SNAPPING

Clamp the free end of the base to a table top or workbench. Place your item(s) on the turntable. Set up your camera, tripod, and lighting. Take a photo and turn the crank handle exactly one turn. Repeat this until you have 16 photos. (If you want 32, you can use half-turns.)

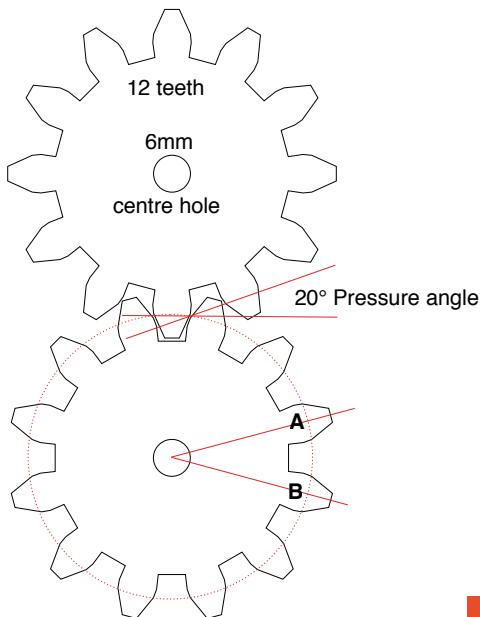
GIF ME A BREAK

Once you have your 16 (or 32) photos, you can convert them into an animated GIF using a free tool at: hsmag.cc/WlMAVY.

GET IN GEAR

Circular pitch determines tooth size. It's defined as 'the arc distance between the centres of two adjacent teeth'.

Pressure angle is defined as 'the angle between the line of force and a line at right angles to the centre line of two gears at the pitch point'. It's a key parameter of gear design, but 20° is fairly standard, so we'll stick to that. The software allows you to vary the value between 10° and 30° if you want to see what difference it makes. You don't need to know anything much about it except that it's important that all gears that you want to mesh together have the same pressure angle and circular pitch.



- Upload your images
- Check that they're in the correct order. You can drag and drop them into place if you need to move any
- Set your canvas size (in % or pixels)
- Drag the slider to set your animation delay speed (milliseconds between frames)
- Set your number of repeating loops (0 = infinite)
- Click the 'Create GIF animation' button

After a short while, below the 'Create GIF animation' button you'll see links to 'View the GIF' and 'Download the GIF'.

There's no preview or prediction of the file size, so you'll need to download the GIF file to check how large it is. You can tweak the parameters to get the target file size and rotation speed you want. For Twitter, your GIF needs to be under 5MB. Generally ~200 milliseconds works well for animation speed.

In the 'blue tin' example, hsmag.cc/yddMXd, 32 frames at full resolution of 1001 × 795 pixels gave a 7.9MB file. To get this below 5MB, it had to be scaled to 60% (600 × 477 px), which gave 4.8MB.

Above

A tripod or magic arm is essential to keep the camera position constant. A remote shutter-release really helps too. Note the turntable base firmly clamped to the bench

QUICK TIP

The bottom two nuts on each axle are 'locked together', but the axle should be able to spin without too much resistance. It's a trade-off between 'too tight' and 'wobbly'.

EXTRA CREDIT

This project is fairly easily scalable and tweakable. If you want a larger turntable or better angular resolution, you could use a longer base and add a fourth axle with an extra gear set on it. It could also be motorised and/or combined with a camera remote to fully automate the process. You could control that with an Arduino, ESP8266, or Raspberry Pi. But, however you decide to build it, have fun, and take some great GIFs. □

Left

Two intermeshed 12-toothed gears showing pressure angle and circular pitch

DON'T MISS THE BRAND NEW ISSUE!



HOME AUTOMATION WITH RASPBERRY PI

Practical projects to take your home into the future

SUBSCRIBE FROM JUST £4

- FREE! £5 ModMyPi Voucher
- FREE! Delivery to your door
- NO OBLIGATION! Leave any time

WIN!

7 Nanomesh sound systems worth £70 each

TESLONDA TEST DRIVE

The Tesla Honda hybrid car made with Raspberry Pi

Keep your Minecraft world safe with terminal multiplexing

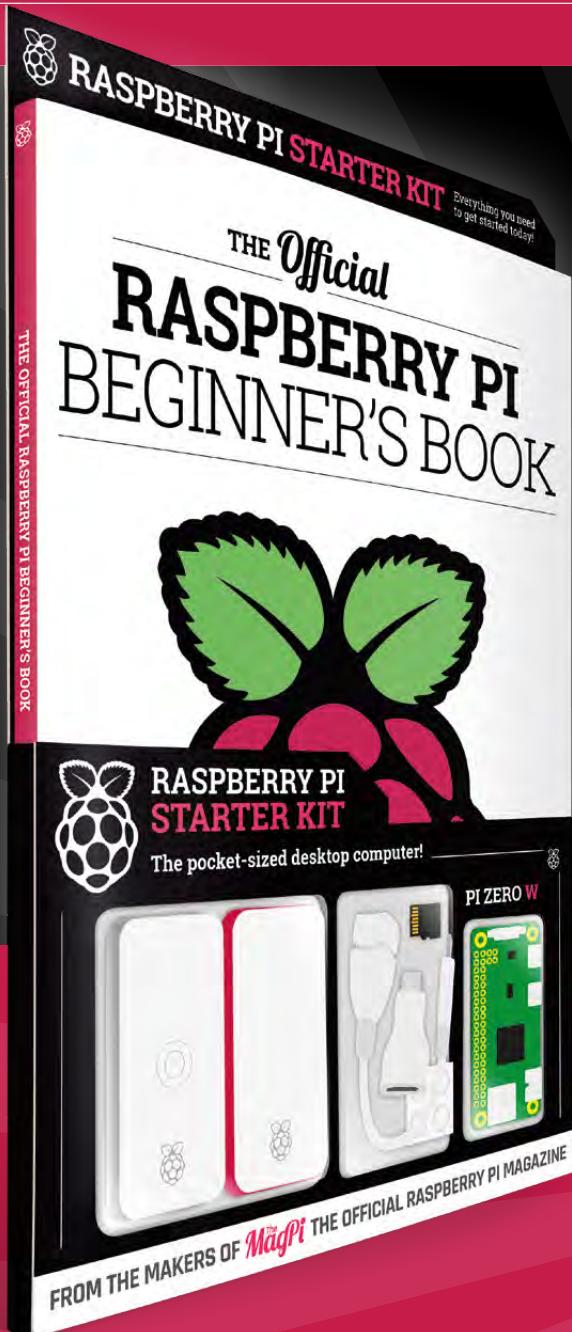
PLUS! FREE
CASE, THREE
COVERS &
CABLES

FREE PI ZERO W
With your 12-month subscription to the print magazine
magpi.cc/Subs1

Buy online: store.rpipress.cc

THE *Official*

RASPBERRY PI BEGINNER'S BOOK



LEARN
COMPUTING
THE EASY WAY!

Includes

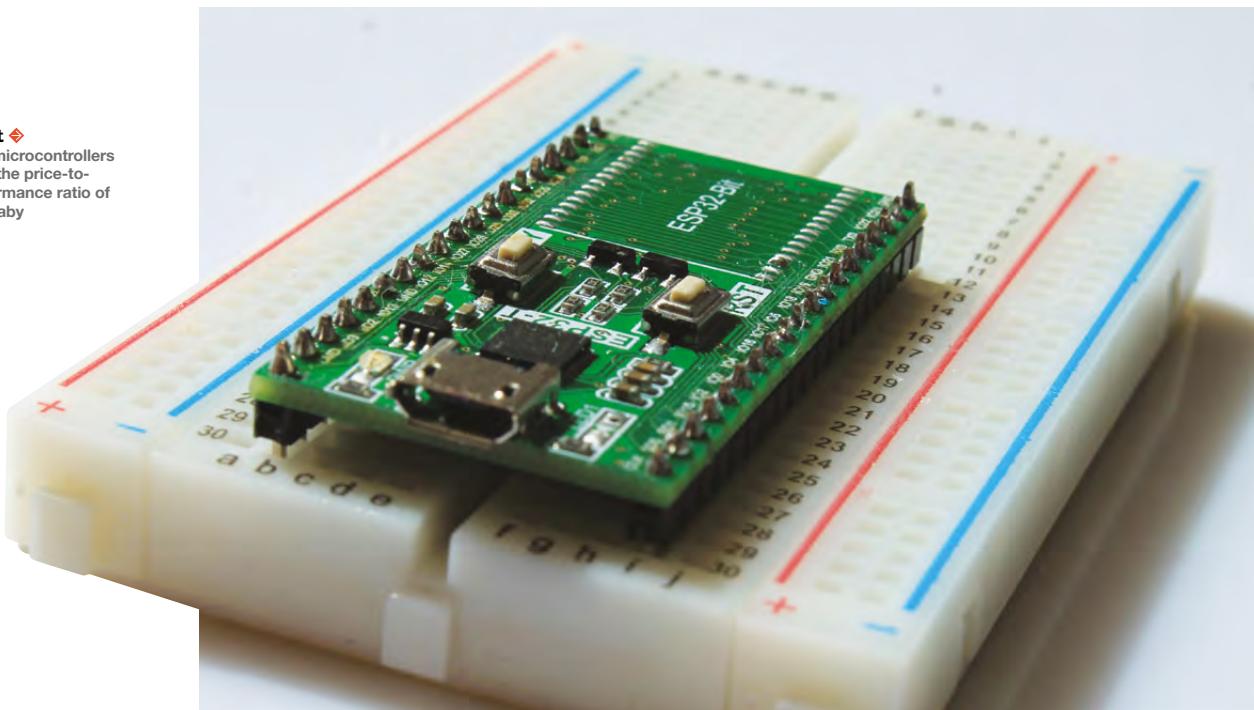
- Pi Zero W computer
- Official case with three covers
- USB and HDMI adapters
- 8GB microSD card
- 116-page beginner's book

Available
now



Buy online: hsmag.cc/store

Right ♦
Few microcontrollers have the price-to-performance ratio of this baby



Unleashing the full power of the ESP32

Getting used to the small but powerful microcontroller



Jorim Bechtle

Jorim is a student who is interested in practical, powerful but cheap MCU solutions for every problem.

There's a wide range of microcontrollers on the market with a range of different features and prices; however, few have quite the price-to-performance ratio of the ESP32. It's got two cores

pumped up to a clock rate of 240MHz and a bucket-full of connectivity built in including WiFi, Bluetooth 4.0, Bluetooth LE, DACs, a Hall effect sensor, and much more. On top of this, the ESP32 is even cheaper than the Arduino Uno.

To learn a little bit more about the ESP32, let's take a look at how to program the device. There are two main possibilities – either you use the Arduino software or go through the ESP-IDF (ESP integrated development framework). Although it is a little bit easier to use the Arduino software, it kind of limits what you can do with your ESP32, e.g. you can't use the second core and the clock rate is limited to 80MHz. As well as this, the

Arduino software produces much bulkier code, which results in a slower MCU with higher current draw. We're going to use the ESP-IDF to get the most out of our hardware.

There are two different approaches to set everything up. Firstly, you could just use a Raspberry Pi, code using a text editor, and compile and flash using some shell commands. Secondly, it is possible to integrate all the necessary commands for compiling and flashing into Eclipse on a Linux-running PC. Thirdly, kind of a mix of the first two approaches, it is possible to use any editor under Windows, upload the C files to your Raspberry Pi, then compile and flash from there.

To really understand what you're doing and what exactly is happening, let's start off with the full RPi build server approach and afterwards have a look at how to make this process easier. In this tutorial, we'll use a freshly set up Raspberry Pi over SSH, because you'll just need the terminal. To structure everything,

```
I (236) cpu_start: Pro cpu start user code
I (254) cpu_start: Starting scheduler on PRO
CPU.
I ( 0) cpu_start: Starting scheduler on APP
CPU.
Hello world!
This is ESP32 chip with 2 CPU cores, WiFi/BT
/BLE, silicon revision 0, 4MB external flash
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
```

Above

Communicating with the Hello World project for the ESP32 over the serial monitor

let's put all ESP32-relevant files into **/home/pi/esp**. So first, you'll need to create that folder.

There are two main components needed: the ESP-IDF itself and the additional Xtensa Compiler. In order to use those two components, you're going to need some prerequisites:

```
sudo apt-get update
sudo apt-get install git wget make libncurses-dev
flex bison gperf python python-serial
sudo apt-get install gawk grep gettext automake
texinfo help2man libtool libtool-bin python python-
dev python-pip
sudo pip install pyserial
```

The first list of programs to install are the prerequisites for the ESP-IDF; the second list is needed for the Xtensa-Toolchain. If you're using something other than a Raspbian or Debian distribution, just use your package manager.

The IDF can be downloaded with:

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

Don't forget the **--recursive**, otherwise you'll run into plenty of trouble later on when you're trying to compile your code.

The Xtensa-Toolchain can be compiled from scratch to get the latest version, but because it takes several hours with the RPi, it is recommended to use a precompiled Xtensa-Toolchain, which is available on GitHub: [hsmag.cc/qASTvv](https://github.com/Dschorim/xtensa-rpi). The precompiled version can be downloaded via:

```
cd $HOME/esp/
git clone https://github.com/Dschorim/xtensa-rpi.git
```

Now it is time to tell your system where your ESP-IDF is, so your project can be compiled and flashed correctly. You can either execute:

```
export IDF_PATH=$HOME/esp/esp-idf
export PATH=$PATH:$HOME/esp/xtensa-rpi/bin
```

...every time you start your shell, or you can add these lines to **/home/pi/.bashrc** so that it gets executed automatically every time you start a session with this user. In order to activate these path variables, just relog or reopen your terminal.

YOU'LL NEED

- ◆ **ESP32 development board**
- ◆ **Breadboard**
- ◆ **Potentiometer**
- ◆ **Any Raspberry Pi or computer**
- ◆ **LED**
- ◆ **Connectors**

MAKING ALLOWANCES

Now it's time to set the permissions for the ESP32. After plugging in the ESP via USB, you can notice a new device in **/dev**, most of the time called 'ttyUSB0' or something like that. If you're unsure, just disconnect and reconnect your ESP32 and look at which device popped up. To be able to use this device without having to sudo everything, you need to add your user to the group that possesses the ESP32: in case of the RPi, it's the group 'dialout'; on other Linux distributions it might be something like 'uucp'. Just make sure which group is the owner.

```
sudo usermod -a -G dialout pi
```

Here, **dialout** is the name of the corresponding group and **pi** the name of your user. To put these changes into action, it's time to reboot your Raspberry Pi – or at least, relog.

The next thing to do is create a new project. It is recommended to create a workspace like **\$HOME/esp/workspace**, where you put all your projects, so things don't look that messy, and to copy a project template from the ESP-IDF.

```
mkdir $HOME/esp/workspace
cd $HOME/esp/workspace
cp -r $IDF_PATH/examples/get-started/hello_world/
$HOME/esp/workspace/tutorial
```

You can check out if everything is working by setting up, compiling, and flashing the 'Hello World' project. Unless you want to alter some hardware settings, it is recommended to not change anything in the menuconfig, so you can simply see whether everything is working fine. Later on, you can use →

Below

The ESP32, which is ttyUSB0, belongs to the group dialout, and the user pi belongs to this group

```
pi@raspberrypi:~ $ ls -la /dev | grep "ttyUSB0"
crw-rw---- 1 root dialout 188, 0 Apr 21 17:02 ttyUSB0
pi@raspberrypi:~ $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games u
users input netdev spi i2c gpio
pi@raspberrypi:~ $
```

TUTORIAL

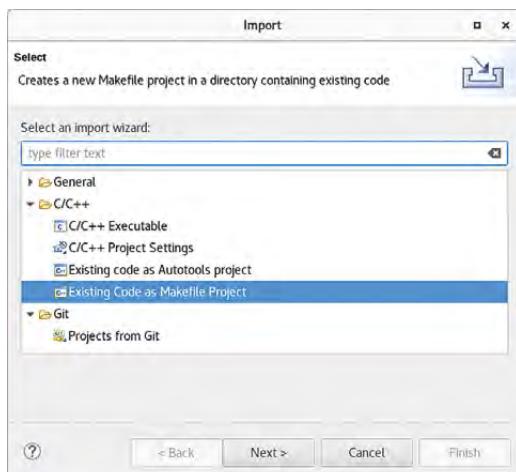


Figure 1 Importing the 'Hello World' project into Eclipse as a makefile project will fully use its features

the menuconfig to disable things – e.g. WiFi or Bluetooth LE – if you don't need them.

```
cd $HOME/esp/workspace/tutorial  
make menuconfig  
make -j5 all  
make flash monitor
```

Using the suffix `-j5` in `make all` is optional, but it enables multithreaded compiling. The number should be set to one more than your hardware cores/threads – in the case of the RPi 3B, that's $4+1 = 5$ – so that the Xtensa Compiler fully uses the hardware it has.

LINUXING

Now that this version works, let's have a look at using Eclipse. For this setup, we'll use the `$HOME/esp/` directory as a setup directory again.

First of all, change the permissions of the ESP32 device and download the ESP-IDF as explained earlier, but this time on your Linux PC, and download and install the C and C++ version of Eclipse, which can be found on the website: eclipse.org/downloads.

You'll also need to set up the Xtensa-Toolchain, but this time not the ARM version. You can get hold of the precompiled x86 and x64 versions officially from Espressif: hsmag.cc/RrOvss and hsmag.cc/BkNecM (automatic downloads) respectively.

Just download and extract the version matching your Linux version to `$HOME/esp/xtensa`. Now it's time to start Eclipse and to set your workspace – in other words, `$HOME/esp/workspace`.

The necessary adjustments are made for each project individually, so let's start by copying the 'Hello World' example over to our workspace, as explained earlier for the Raspberry Pi. The next thing to do is to explain that there is a new project. This can be done via File > Import and then C/C++ > Existing Code as Makefile Project, as shown in **Figure 1**.

Next, just enter the project name and the folder where you've copied the example project (which has

to be the same) and select Cross GCC. Now you can close this by hitting Finish and get on with modifying the properties of your project by right-clicking on it and selecting Properties. Here you have to specify the folder in your workspace, which you've just copied; you can leave everything else as default.

You have to tell Eclipse where it can find the ESP-IDF and the Xtensa-Toolchain. This can be set under C/C++ Build > Environment and adding the PATH `$HOME/esp/xtensa/bin` and the IDF_PATH `$HOME/esp/esp-idf`. Optionally, you can replace the default build command under 'C/C++ Build' with `make -j5` to enable multithreaded compiling.

After this, you need to tell Eclipse where the include files are. These settings can be found under C/C++ General > Paths and Symbols > Includes > GNU C. There you'll have to add all files you want to import. All standard files are found in `$IDF_PATH/components`. You can either add this one path here and you then have to add the full path in your program, which is unusual, or you can add all necessary paths like `$IDF_PATH/components/esp32/include`. It's advisable to just include the necessary paths, to speed up the compilation time and reduce the binary file size.

SPEED FREAK

The last thing to do is to set up the commands for compiling and flashing. This can be done via Build Targets. Create a new one and call it 'all' and another one called 'flash'. It is also recommended that you go on to create a third target called 'clean' to clean prior builds. But before you can start using these tools, you'll have to use the shell in your project's directory with `make menuconfig` in order to specify the hardware settings, because Eclipse doesn't support this. Afterwards you're free to use the build targets from Eclipse to try compiling and uploading the example project. The last thing to do is to write your own code and start your project.

DAFFY DAC

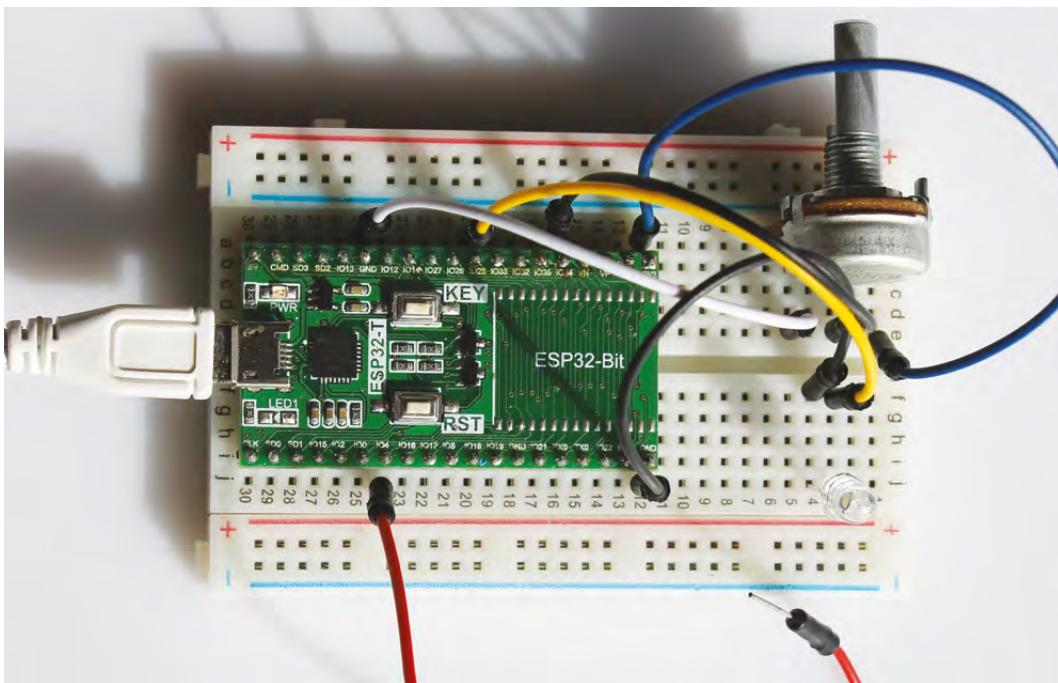
Let's take a look at how to use this software we've just set up.

There are two ADC units with nine channels each, but ADC2 is most likely used for WiFi, which has priority and therefore precludes the use of ADC2. The ADC can be used via:

```
#include 'driver/adc.h'  
adc_gpio_init(ADC_UNIT_1, ADC_CHANNEL_6);  
//initialize channel 6 from module 1  
adcX_config_width(ADC_WIDTH_12Bit);  
//set 12-Bit resolution
```

QUICK TIP

You can find a link to the newest version of the Xtensa-Toolchain at hsmag.cc/rVFOBW.



QUICK TIP

More information can be found in the ESP32 documentation: hsmag.cc/JmicUE.

Left Set up to test the following pieces of code with a variable voltage for the ADC

```
adcX_config_channel_atten(ADC1_CHANNEL_6,
ADC_ATTEN_DB_0); //set scale
adc_value = adcX_get_raw(ADC1_CHANNEL_6);
//read value into adc_value
```

First, the specific ADC channel gets initialized and the resolution (9,10,11, or 12-bit) gets set. Afterwards you have to set the scale (0dB is used for the full range from 0 to 3.3V). Now you can use `adc1_get_raw(ADC1_CHANNEL_X)` or `adc2_get_raw(ADC2_CHANNEL_X)` to measure the voltage.

There are two real DACs (not just PWM – which is, by the way, also possible) available on GPIO25 and 26.

```
#include 'driver/dac.h'
dac_output_enable(DAC_CHANNEL_1); //enable dac
channel 1 = pin 25
dac_output_voltage(DAC_CHANNEL_1, value);
//write value on pin 25
```

Just enable the specific DAC channel and set a value from 0 to 255.

The integrated touch module enables recognising physical touch by measuring the capacitance of a specific pin. It needs to be initialized and the specific pin needs to be configured. Afterwards you can read the capacity and store it in `touch_value`.

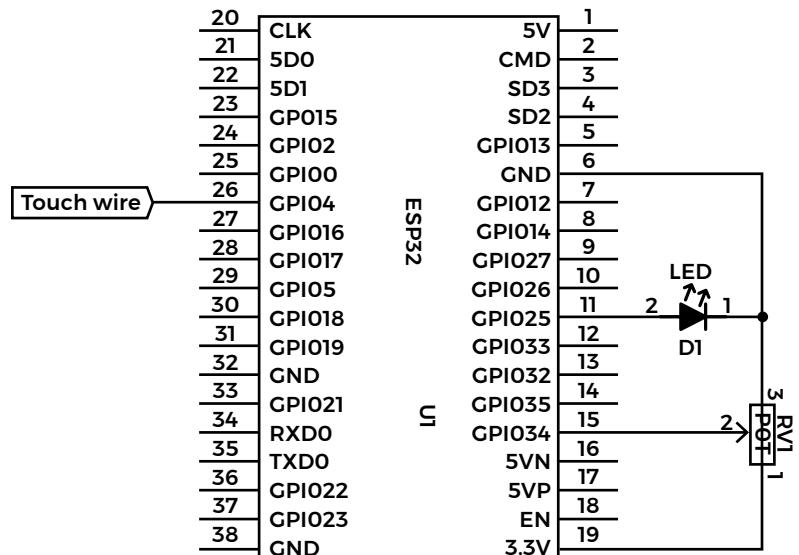
```
#include 'driver/touch_pad.h'
uint16_t touch_value = 0; //initialize variable
for touch value
touch_pad_init();
touch_pad_config(TOUCH_PAD_NUM0, 0);
touch_pad_read(TOUCH_PAD_NUM0, &touch_value);
```

Another useful feature is multicore application execution. There is the possibility to create tasks which use a free core, or you can dedicate a task to one core (Core 1 in this example). This can be done by using the following to start the method `task_method`.

```
#include 'freertos/task.h'
xTaskCreate(&task_method, 'task_method', 2048,
NULL, 5, NULL);
xTaskCreatePinnedToCore(&task_method, 'task_
method', 2048, NULL, 5, NULL, 1);
```

An example application using a WiFi AP and an HTTP server can be found at hsmag.cc/fjusKf. □

Below Schematic for the example project including an LED, a potentiometer, and a touch wire



WARNING



This is an advanced tutorial for people comfortable and knowledgeable about handling chemicals and UV light sources. This tutorial is a guide only. You are responsible for your own safety. Ensure you are familiar with safe handling of the substances involved before starting. If in doubt, get your PCBs made for you (see boxout).

Make your own printed circuit boards

Everything you need to know to fabricate PCBs at home



Ricardo
Caja Calleja

funwithcables.wordpress.com

An aerospace engineer by profession, Ricardo is deeply interested in robotics and automation. If there's nothing to repair at home, he'll make up some plan to build anything that includes cables or screws.

W

hen the prototype of the electronic circuit that you made on a solderless breadboard has reached its final version, you need to implement it on a more resistant and durable

board. This can be done either on a soldered protoboard or a printed circuit board (PCB). The latter allows for an easier and cleaner implementation of the electronic components. You can get these created in a factory, but it's quicker and cheaper to make them yourself.

Electronics manufacturers started using PCBs in the 1950s to reduce all the messy wire connections, thereby reducing errors and costs. Those wire

connections would be replaced by traces of a copper layer, bound to an insulating board (substrate) made of materials such as ceramic or fibreglass. This technology made electronics manufacturing far cheaper and more reliable, and paved the way for mass-market electronic devices.

Although not so complex and small as professional PCBs, you can also make PCBs at home, surprisingly with few materials and at relatively low cost. You will be limited to single- or double-layer boards (professional boards can have several layers). We'll look at single-layer boards, as these are simpler yet still powerful enough for many projects.

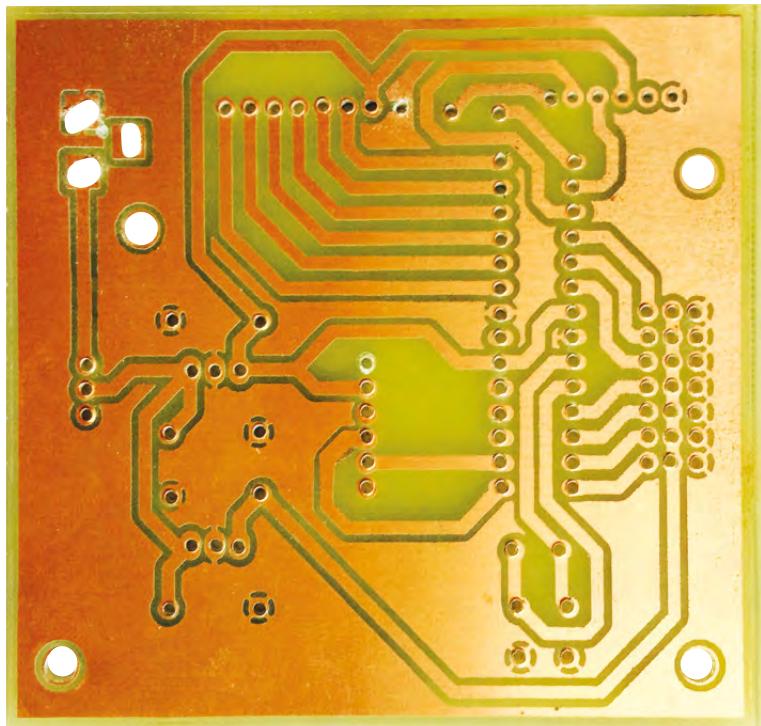
We're going to create our single-sided board using photolithography (also known as UV lithography

or optical lithography). It consists of exposing a pre-sensitised photoresist PCB to UV light. The main tool you will need is an ultraviolet (UV) exposure box, which you can easily build yourself.

MAKE YOUR TOOLS

In the past, UV exposure boxes were typically built with fluorescent lamps or tubes, but this resulted in a bulky construction which required a lot of space and power consumption. Now LED technology allows you to make a smaller and cheaper box.

You can build the circuit of your UV LED exposure box on a soldered protoboard or a PCB, if you already have the means for the second option. The size of the exposure box is up to



Right Finished PCB

you, which means that you will need more or fewer UV LEDs depending on its size. This is because the distance between LEDs (on both horizontal directions) has to always be the same: approx. 2cm. Since the light emitted by LEDs is a sort of cone-shaped beam pointing at the vertical direction, considering the LED characteristics and their vertical distance from the PCB that is to be exposed (4–5cm), the LEDs should be placed at the aforementioned distance one from another, in order to obtain an even amount of UV light on any point of the PCB that is being exposed.

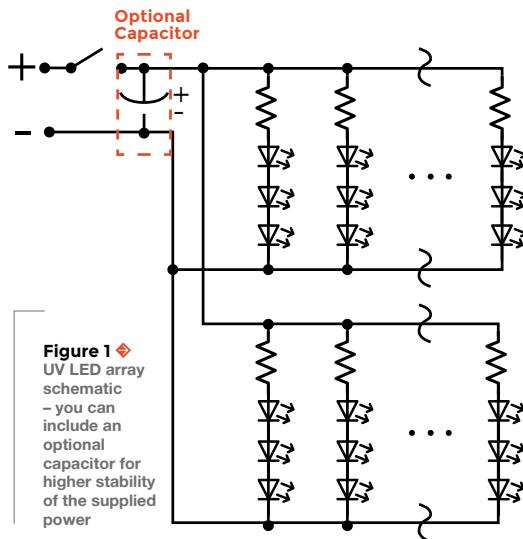
Do not turn on the LEDs outside of a sealed box. They can cause eye damage.

The circuit of the exposure box is a so-called ‘series/parallel circuit’ (**Figure 1**), as it combines parallel branches of three LEDs in series. Since every parallel branch is powered by 12V DC, the electrical current in every branch has to be limited to the forward current of the LEDs ($I_F = 20\text{mA}$) by means of a resistor. So the calculation of the resistor value (R) is very easy:

$$R = (\text{VDC} - \text{VLED}) / I_F = (12 - 3 \times 3.5) / 0.02 = 75\Omega$$

You can vary the number of LEDs if you re-calculate the resistors. Any $\frac{1}{4}\text{W}$ resistor with a value slightly above or below R will be just fine.

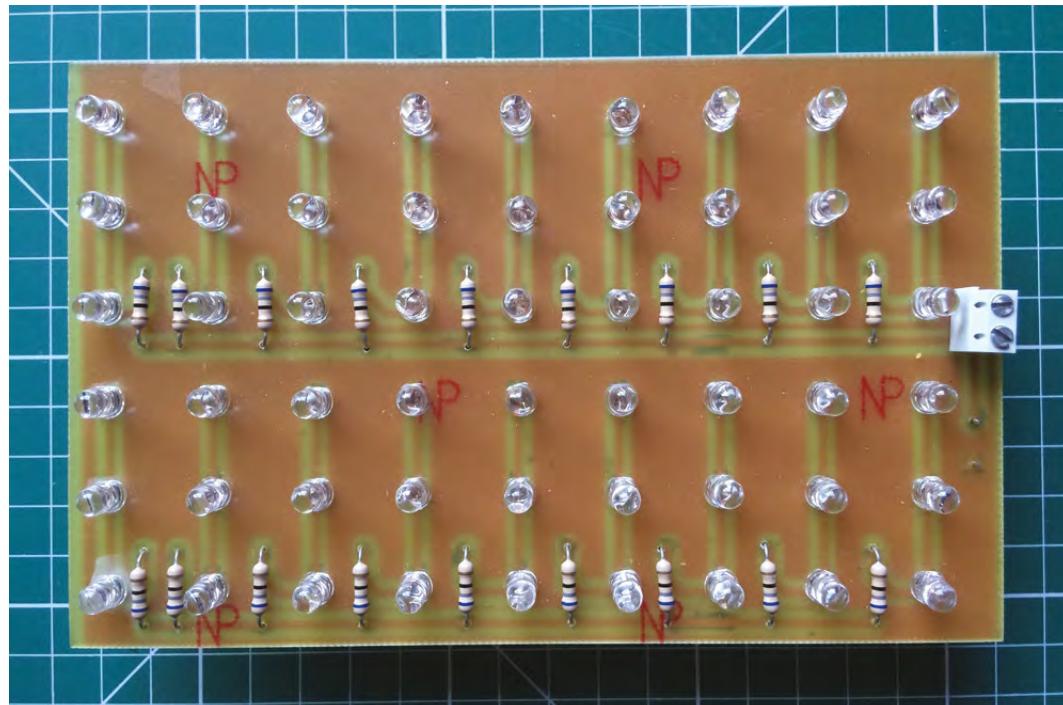
Once the exposure box circuit is built (**Figure 2**), you need to fit it into a box that can be closed



completely with a lid on the top, and which includes 3mm-thick glass on the upper part, where you’ll place the PCB to be exposed (**Figure 3**). As mentioned, it is essential that the vertical distance between the UV LEDs board and the glass is between 4 and 5cm.

Now you’re ready to expose pre-sensitised PCBs with your own UV box! A good idea is to include a switch, to turn the circuit on and off, without →

Figure 2 UV LED array



YOU’LL NEED

UV exposure box

- ◆ UV LEDs (number depends on exposure box size)
- ◆ 68 $\Omega \frac{1}{4}\text{W}$ resistors (number depends on exposure box size)
- ◆ Protoboard or PCB
- ◆ 100 μF capacitor (optional)
- ◆ Enclosure
- ◆ Wires
- ◆ Switch
- ◆ Power jack
- ◆ Soldering iron and solder

PCB fabrication

- ◆ Inkjet / laser transparency film
- ◆ Inkjet / laser printer
- ◆ Single-sided pre-sensitised positive photoresist PCB
- ◆ 2 x plastic trays
- ◆ Plastic tweezers
- ◆ NaOH (caustic soda)
- ◆ H₂O (tap water)
- ◆ Hydrogen peroxide (H₂O₂, 3% food grade)
- ◆ Hydrochloric acid / spirits of salt (HCl, 24% or less)
- ◆ Alcohol or acetone
- ◆ Rubber gloves
- ◆ Protective goggles
- ◆ Dust mask
- ◆ Dremel with 3/64" drill bit
- ◆ Hacksaw

TUTORIAL



For defining the PCB layout of your circuit you need some **CAD (computer-aided design) software**

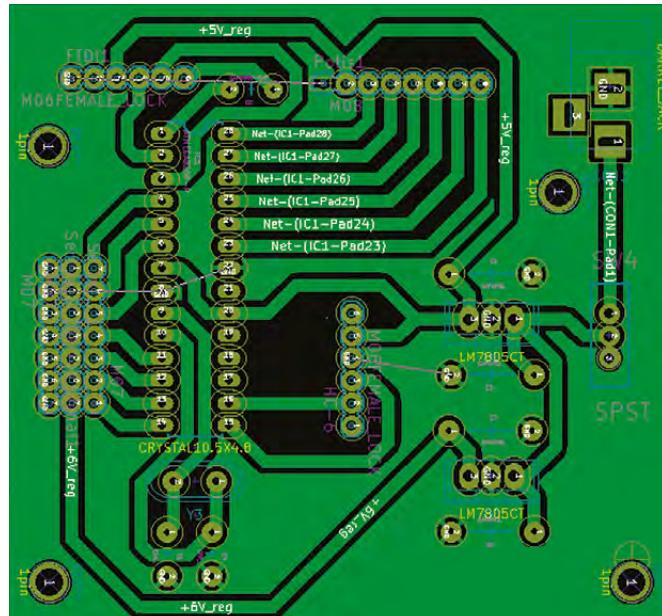
Figure 3 You can make the box yourself with any material you like, or repurpose an existing enclosure, such as a cosmetic case

having to keep plugging/unplugging the circuit. You can also include a timer, so that the LEDs would turn off by themselves after a specific time, which you define, has passed.

GETTING PHYSICAL

For defining the PCB layout of your circuit, you need some CAD (computer-aided design) software. There are several options available for this; KiCad (**Figure 4**) and EAGLE are very popular among makers. The first one is open-source (completely free) and the latter is proprietary software, but offers a free version with features that will be more than enough for most of what you'll need. These bits of software take a little getting used to – we'll just cover the basics here.

Figure 4 PCB layout generated with KiCad



STAY SAFE

PCB fabrication can be dangerous. Make sure you take proper precautions:

- The chemicals used in PCB fabrication are toxic. Always handle the chemicals with protective gloves and goggles, in a very well-ventilated room. Do not breathe the vapours released during the developing and etching processes. Avoid any contact of the chemicals with your skin and eyes, as they can produce severe burns or irritations, and immediately contact the emergency services if you accidentally come into contact with them. Keep out of reach of children and animals. Do not drink. If in doubt, follow the supplier's handling recommendations.
- Direct exposure to UV radiation may produce eye injuries. Avoid direct eye contact with UV light, always keeping the lid of the UV exposure box closed while it's functioning.
- Use a dust mask while cutting or drilling PCBs, as many small particles are released.
- Once you're done with the chemicals, follow local regulations for disposal.



Figure 6 ◊
Layers of pre-sensitised PCB

The first step with CAD software is the schematic capture, which involves drawing your circuit. Once this is done, the software will convert this abstract representation of the circuit (schematic) into a PCB layout, which is how the circuit will be on a physical board. Now you have to decide the size of the board, arrange the components, and define the traces (copper tracks that will act as conductors in the circuit). A bit tricky in the beginning, this is an iterative process to be repeated several times, until the whole circuit fits in your board.

It's a nice idea to print the PCB layout on paper and carefully check for errors. CAD software normally includes a functionality called DRC (design rules check), which helps you find any wrong or missing connection, so you can even skip printing it on paper if you're careful enough. Now you need to produce a

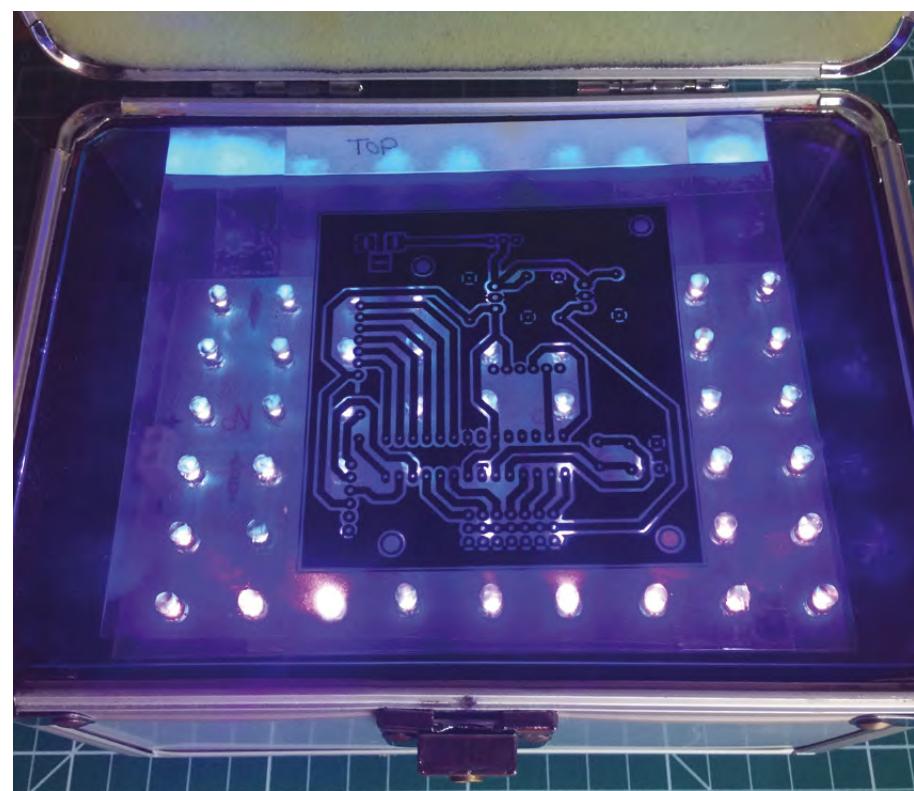
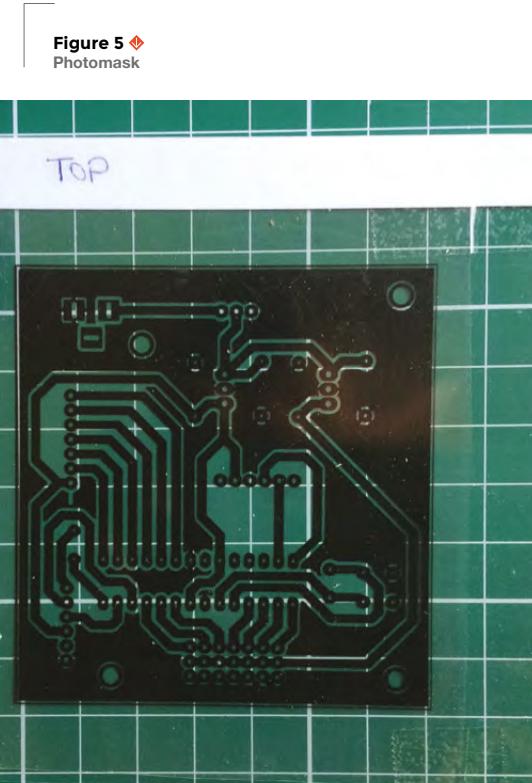
'photomask' (**Figure 5**), which is a printout of the PCB layout on a transparent film with black ink. To ensure the opacity of the printout, it is recommended to use a laser printer, but a high-quality inkjet printer can also do a good job. If the latter is your only option and you notice that the printout is not opaque enough, just print it twice, and very carefully stick the two films together with tape on the sides.

SUN TAN

Similar to photography, the fabrication of PCBs by photolithography consists of using UV light to transfer the pattern of the photomask to a light-sensitive chemical photoresist applied on the copper layer of the PCB (**Figure 6**). You can buy blank copper boards and sensitise them yourself with photoresist (sold in bottles), but pre-coated boards guarantee a uniform coating of resist and will give you better results. So it's better to pay a bit of extra money and go for the latter option.

The photoresist is sensitive to UV light, which means that your UV exposure box or sunlight will expose it, but not regular LEDs or incandescent bulbs that you may have at home. Although a short exposure to room lighting won't damage the board, once you have removed the protective foil of the →

Figure 7 ◊
Board exposure – you must always keep the lid closed while the UV LEDs are on (appropriate safety precautions were taken while shooting this picture)



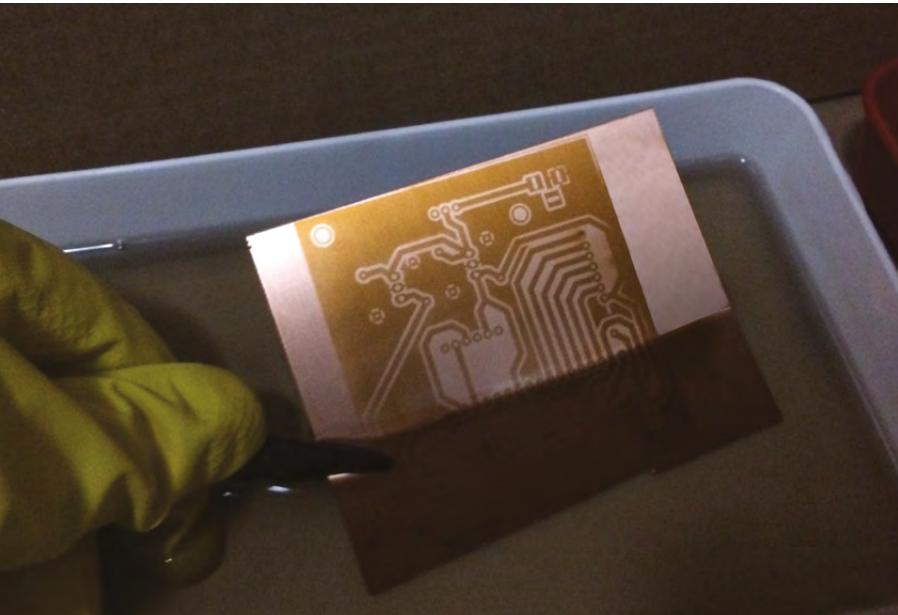
TUTORIAL



Right ↗
Pre-sensitised
boards come in many
different sizes; 1.5mm
thickness is one of the
most popular

Figure 8 ♦
The developing bath

“The photoresist is sensitive to UV light,
which means that your UV exposure box or
sunlight will expose it



pre-sensitised PCB, you should definitely try to keep the room as dark as possible.

Now it's time to use your brand-new UV exposure box. Just place the photomask on the glass, as in **Figure 7** (making sure it's the correct side) and put the PCB on top, with the pre-sensitised side facing down. Turn the exposure box on and wait for about four minutes. You should do several tests, varying the time as, depending on the board manufacturer, better results may be obtained with different exposure times.

As in photography, the PCB needs to be developed. There are several chemicals that may be used for this, a solution of caustic soda (NaOH) and water (H_2O) being one of the best choices: 1% NaOH and

THE EASY OPTION

Making your own PCBs at home is fun and you get to start using them instantly, but it's not the only option. You can send your PCB design files off to a manufacturing service. Prices and lead times vary significantly (often one is inversely proportional to the other). If you'd rather avoid the chemicals and get your PCBs made for you, see OSHPark.com, Ragworm.eu, EuroCircuits.com or one of the other internet options.

99% H₂O. You should add about ten grams of caustic soda pellets per litre of water (or the proportional amount of caustic soda if you're using a smaller amount of water) and wait for it to dissolve (it helps to gently shake the solution). Then the board has to be submerged in the solution (**Figure 8**) until the PCB layout is clearly visible on the board surface. It is recommended that you carefully move the plastic tray in different directions, so the whole surface of the board is homogeneously developed. The photoresist that had been exposed to UV light (through the empty parts of the photomask) will be now gone.

ETCHING POWDER

Again, there are many chemicals that you can use for etching, and one of the most popular choices is a solution of hydrogen peroxide (H₂O₂, 3% food grade) and hydrochloric acid (HCl, also known as spirits of salt): two parts of H₂O₂ and one part of HCl. One of the main advantages of this method is that it works

at room temperature, so you don't need to warm up the solution. It does need to be done in a VERY well ventilated environment. As soon as you take the board out of the tray with the developing solution, you have to quickly submerge it into the etching solution, moving the tray again in different directions, and wait for the parts of the copper that were exposed to UV light to disappear. When you are sure that all the copper in those areas is gone, take the board out of the etching solution and place it under running water, so the etching process is stopped. If you leave the board in the etching solution for too long, the copper that had not been exposed will also start to disappear! Finally, you can clean the board with alcohol or acetone.

The final touch is cutting and drilling. The fibreglass can be easily cut by hand with a metal saw, or you can also use a Dremel with a cutting disc. For drilling holes you should definitely use a small drill, such as a Dremel, with a 1mm (3/64") drill bit. □

Below ↴
Developing the PCB



Frozen reverse spherification

Create edible balloons of flavour to wow your guests



Ben Everard

@ben_everard

Ben is busy trying to find more culinary uses for his chemistry set. Bunsen burner-grilled marshmallows for dessert anyone?

There's a pretty good chance that when you look at this title you'll think to yourself 'what on earth is frozen reverse spherification?' Well, let's take a look at this backwards.

Spherification is a process developed at the El Bulli restaurant, where a flavourful liquid is infused with sodium alginate and then dripped in a bath of calcium lactate. This now-defunct restaurant was, at the time, hailed as the best restaurant in the world and the technique quickly caught on.

The important thing about this process is that water, with either sodium alginate or calcium lactate dissolved in it, is a flavourless liquid (unless some other flavour is added). However, if you mix both chemicals in water, it sets like a firm jelly. As you drip drops of the flavoured liquid infused with sodium alginate into a bath of calcium lactate, the outer layer sets like jelly, but leaves the inside liquid (because the calcium lactate doesn't penetrate

very quickly). This creates little spheres that burst in your mouth. They're often called 'pearls' or 'caviar' by chefs and are popular among fancy restaurants that practise what's become known as 'molecular gastronomy'.

There are a few problems with this process – the spheres are small (around 5 mm in diameter) and typically have to be made in small batches, so it takes a long time to make any meaningful number of them. They also can't be made in advance because the calcium lactate will gradually diffuse through the spheres, rendering the liquid centres solid.

There are hacks that make the whole process a lot easier and the end result more impressive. Firstly, if you switch the two chemicals around, so you infuse a flavourful liquid with calcium lactate and then drip it in a bath of sodium alginate solution, you find that a clear, almost invisible skin forms outside the spheres. This means that the spheres don't gradually harden and they can be kept for longer.



Above ↗
The spoon has become the standard way of presenting large spheres

**Above**

Filling the ice moulds to different levels will lead to different sized spheres

A problem with this method, however, is that you make the spheres by dripping one liquid in a bath of another liquid, so you can only make the spheres the size of a single drip. This varies from liquid to liquid, but they are always quite small, and this size limits the effect you feel biting into the sphere.

DEEP FREEZE

Enter Frozen Reverse Spherification. This is very similar to reverse spherification, except that rather than drip your liquid into the bath, you freeze the liquid in appropriately sized pieces and drop these frozen bits in a warm bath of sodium alginate solution. This is easier to do and allows you to produce spheres in a range of different sizes.

The first thing you need is a flavourful liquid to spherify. You can do this with almost anything, and it can work well with both savoury or sweet flavours. For your first experiment, it's good to start with something easy, so you don't end up ruining something you've spent a long time working on if you hit problems. We went for mango juice as it's got a bit of a smoother mouth-feel than other fruit juices. Of course, this is a great area for experiment.

Calcium lactate isn't particularly soluble, so does take some persuasion to get it to dissolve. The easiest way of doing this is with a blender (an immersion blender works better, but upright ones will do the job). You need 2% of the liquid's weight in calcium lactate. We used 200g of mango juice, which was more than enough for plenty of spheres. →

**Left**

Gently warm the pan to help melt the spheres, but be careful not to let it get too hot

YOU'LL NEED

- ◆ Flavourful liquid (such as mango juice)
- ◆ Sodium alginate
- ◆ Calcium lactate

TOOLS

- ◆ Accurate scales
- ◆ Ice cube tray
- ◆ Slotted spoon
- ◆ Blender

TUTORIAL



Spherification is a process developed at the El Bulli restaurant where a flavourful liquid is infused with sodium alginate and then dripped in a bath of calcium lactate

Above A slotted spoon with no sharp edges will help prevent bursting the spheres

This calls for 4g of calcium lactate. You'll need some quite accurate scales to get the weights right. Once everything is measured out, pop it all in the blender and whiz until the powder is dissolved. You'll probably find that some of the calcium lactate gets stuck in the blender – this shouldn't be a problem as long as most of it dissolves.

You can then put this in ice cube trays to freeze. There are manufacturers that sell special trays

XANTHIUM GUM

When the spheres pop, the liquid quickly fills your mouth and this can be quite a dramatic sensation with a runny liquid. If you prefer a gentler pop, you can first thicken the liquid. Xanthium gum is an easy way of doing this.

Add between 0.1% and 1% of the flavoursome liquid's weight – the liquid will thicken as soon as the gum is dissolved, so add it slowly and stop once you've reached the desired viscosity. This can mean that this liquid is more prone to trapping bubbles. If this is the case, you might need to leave it to stand until these bubbles have dissipated (it may be worth leaving it overnight). Once the bubbles are out, freeze and spherify as normal.

specifically for making ice spheres that are supposed to make the process easier, but we found that it worked fine using a standard ice cube tray. It is worth thinking about the size – we used a tray that created cubes about 2cm across, which is a little smaller than the average ice cube tray, but a good size for this. Remember that you don't have to fill each compartment completely and it can be good to experiment a little. Putting less in will result in smaller spheres, but if the resulting frozen liquid is too flat, it might struggle to make a good skin to hold the inner liquid in place.

TAKING A BATH

Once your proto-spheres have frozen, it's time to start work on the sodium alginate bath that will transform them from ice cubes into spheres with a jelly outside and liquid centre. This is just a case of dissolving 0.5% sodium alginate in water. We used 1 litre of water (1kg) and 5g of sodium alginate. The amount you need really depends on the size of your pan. You want to have at least 5cm deep liquid in a pan (we'll be heating this soon so it does need to be a pan and not just a bowl). Again, it's best to use a blender to thoroughly mix the liquid. Be sure to fully clean the blender after mixing the calcium lactate, as otherwise this stray chemical could cause this to set.

You'll end up with a gloopy liquid full of bubbles. There are plenty of companies that will sell you a vacuum chamber for getting rid of these bubbles, but we found that we didn't need it. It is, essentially, the viscosity of the liquid that holds the bubbles in and,

as you heat the liquid, the viscosity goes down and the bubbles escape. We need this liquid to be warm to melt the ice cubes anyway, so this isn't a problem. Heat the liquid to around 50°C (the exact temperature isn't critical, but it needs to be warm but not too hot), and wait for most of the bubbles to clear. It's not a problem if there are still a few bubbles left. Gently stirring the liquid can speed up the process.

COMBINE AND HARVEST

We're almost there, you just have to get a bowl of water ready to rinse off the spheres.

You've now got the crucial parts you need for frozen reverse spherification, you just need to combine them. Pop your ice cubes out of their mould and drop them a few at a time into the warm bath. The outer layers of these cubes will melt quickly, and the calcium lactate will combine with the sodium alginate to create a firm, jelly-like layer that holds the liquid in. After about 30 seconds, give the bath a gentle stir. About a minute or so later, scoop these spheres out with a slotted spoon (ideally a plastic slotted spoon with small holes as this is less likely to puncture the spheres, but any should work if you're careful). It's not a problem if the odd sphere bursts, but make sure to fish around with the slotted spoon to remove any lumps of solidified sodium alginate.

The longer the spheres are in the liquid, the thicker the skin will be. Thicker skins are stronger, but can leave a disconcerting feeling in the mouth. If you find your spheres are too delicate, try leaving them a little longer. Alternatively, if the skin is a little tough in your mouth, try reducing the amount of time it stays in the bath.

Remove the spheres and pop them in the water bath to rinse off any remaining sodium alginate. You can keep them in this water bath for a while if necessary – exactly how long will depend on how thick the skin is, but an hour or two should be fine. The problem with storing them for longer is that osmosis will occur as the outer layer of the sphere is slightly permeable, and the water will dilute the flavour. If you need to store them for longer, putting them in a bath of liquid the same flavour will eliminate this problem. Alternatively you can pop them in a bath of flavourful oil to add some different notes to the finished product.

Your spheres are now ready to eat! You can just pop them in your mouth and bite for a full hit of mango-y enjoyment, or alternatively, you could serve them with a dessert. Remember, we're just using mango juice as a test, and you can put almost any flavour into these spheres, and pair them up with either savoury or sweet dishes. □

OTHER OPTIONS

You can sphériser almost any liquid, and you can have great results with almost anything that's got flavour. It's good to think about things that aren't usually in liquid form, but can be blended – fruit is great for sweet options, and vegetables for savoury.

Most liquids work fine, but you might hit problems if your liquid has a pH of less than 3.6, or if it already has a lot of calcium in it.



Left ♦

The spheres are tough on the outside, but still completely liquid in the middle

Make a neon-style sign

Make your signs light up with glowing wire



Andy Clark

↗ @workshopshed

After an aerospace apprenticeship and electronics degree at Imperial College, Andy took a job as a software engineer. For the last ten years he's been making and repairing in a shed at the bottom of the garden. You can see more of his exploits at workshopshed.com

His project uses electroluminescent (EL) wire and a simple control circuit to light up the sign whenever there is motion in the room. The sign mimics the style of neon tubes without the cost, weight, and fragility. A motion sensor and timer ensure that the sign is only on when there are people in the room. The control circuit uses a Trinket M0, which means that it can be programmed using CircuitPython, a cut-down version of Python that is designed for microcontrollers.

BENDY IS TRENDY

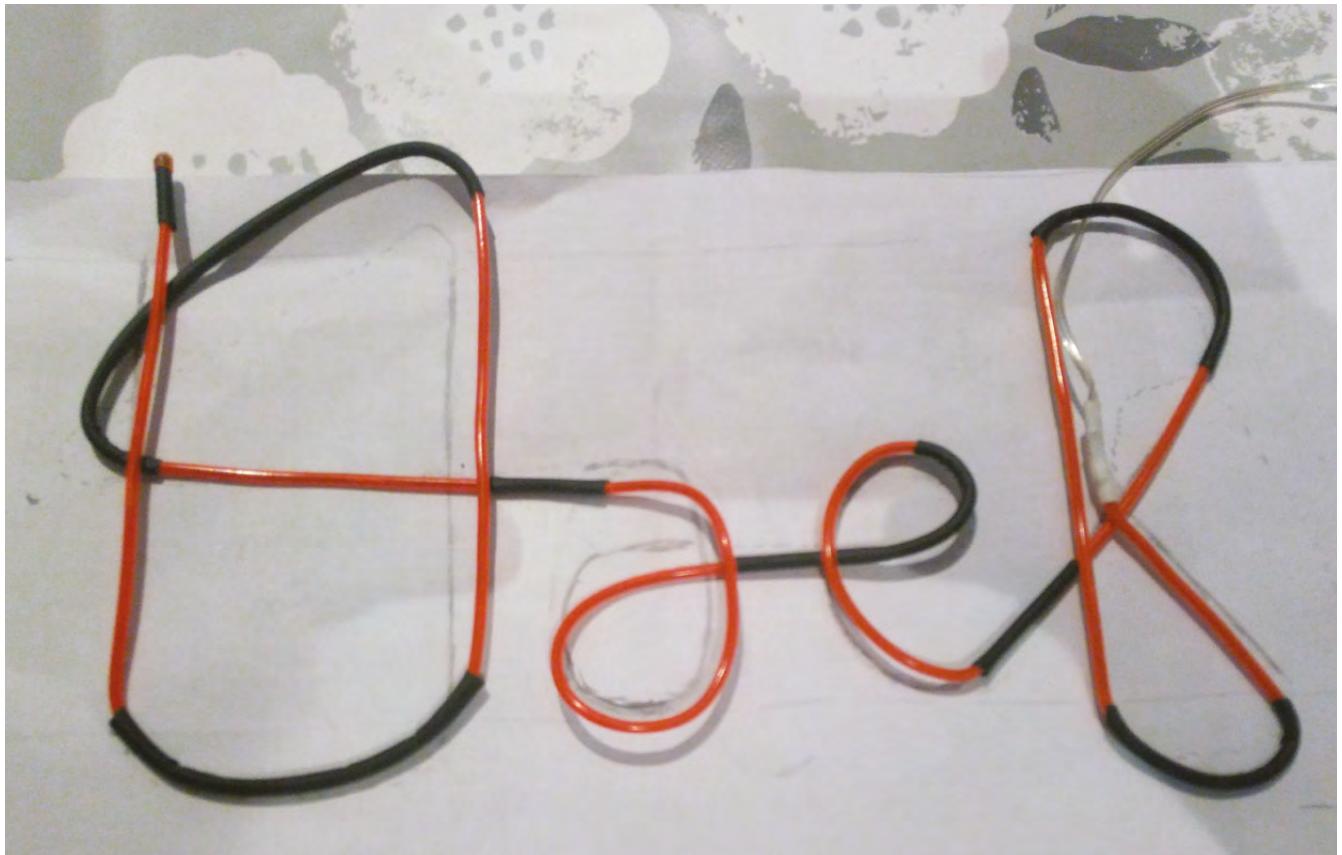
The key to designing the letters is that you must work within the limitations of the EL wire. The first thing is that you need to create the letters of each word with

Above Each colour is created with a single length of EL wire with some parts blacked out

a continuous line of wire. The second is that there is a minimum bend radius for the wire. If you bend it too tightly then it might crack and stop working. So that we don't need to keep cutting and joining the wire, black heat-shrink is used to block out the light between letters.

Draw out your letters on paper or CAD to determine the best routing, and to calculate the length of EL wire to purchase. Ensure your letter heights and widths are consistent to get a good result.

Start forming from the connector end. When you come to a break in the light, cut a piece of heat-shrink



to length and slip it over the EL wire. We used a hot-air gun on the low setting to shrink the sleeving. Try not to use excessive heat, as that could melt the coating of the EL wire.

Once you have all your letters formed, you can then cut the EL wire to length. When cutting the wire, make sure that the thin corona wires do not short-out on the central copper wire. Finish the wire with heat-shrink, or a specialist EL wire cap.

DANGER – HIGH VOLTAGE

The EL inverters produce high voltage of approximately 120V alternating current. Although it is technically possible to switch that, it's simpler and safer to switch the lower 12V input to the inverter. The current from the EL inverters is low, so it is safe to use EL wire around people. However, it is good practice to ensure any exposed cables are covered with heat-shrink. The type and rating of the EL inverter should be matched to the type and length of EL wire. Ask your supplier for advice if in doubt.

ELECTRONIC SWITCHING

The circuit uses two N-Channel MOSFETs to turn on the EL inverters. We used the FQP30N06L, as it can handle the 12V and 1A current provided by our supply.

When cutting the EL wire, make sure that the thin corona wires do not short-out on the central copper wire. Finish the wire with heat-shrink or a specialist EL wire cap

Above Forming letters

A low-value 120Ω resistor in series, with the gate connection, stops too much current being drawn from the Trinket's I/O pins. A higher-value $10\text{k}\Omega$ resistor, from the input to ground, switches off the MOSFET if the control circuit is removed, such as when the Trinket is turned off or disconnected.

When the MOSFET is turned off, the rectifier diode provides a path for any stored energy in the inverter to safely discharge to the supply rail.

HEAT AND MOTION

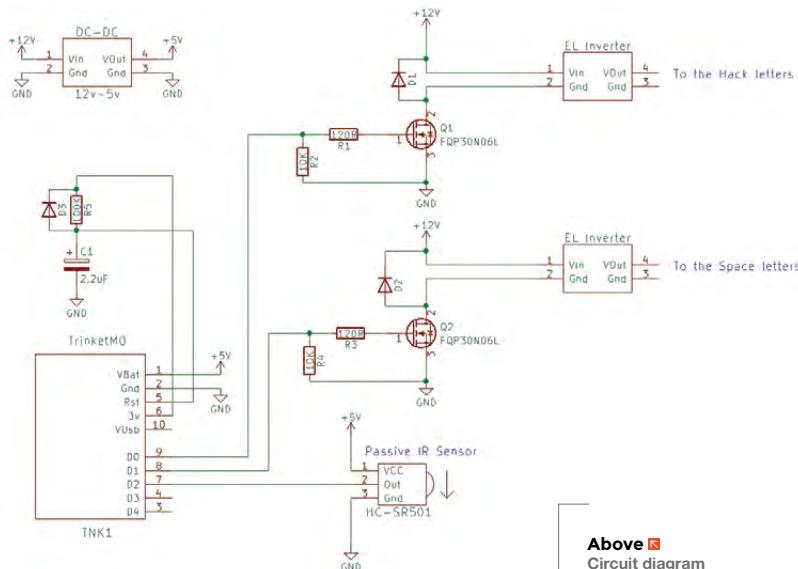
The passive infrared (PIR) module senses heat and motion. An item that is the same temperature as the background would not trigger, nor would a static but warm item. The module has a simple digital output when it detects motion. There are three wires for this →

QUICK TIP

Note that it is not possible to dim the EL wire using this switching technique. You should aim to keep the rate of switching to a few hertz, or less.

Make a neon-style sign

TUTORIAL



Above
Circuit diagram

YOU'LL NEED

- ◆ Power adapter
- ◆ 2 × EL inverter
- ◆ 2 × EL wire
- ◆ Heat-shrink
- ◆ DC-DC converter
12 V – 5 V
- ◆ 2 × N-Channel
MOSFET
- ◆ 2 × 120 Ω resistor
- ◆ 2 × 10 kΩ resistor
- ◆ 1 × 100 kΩ resistor
- ◆ 1 × 22 µF
capacitor
- ◆ 3 × rectifier diode
- ◆ HC-SR501 passive
IR sensor
- ◆ Adafruit Trinket
M0
- ◆ 6 mm MDF sheet
- ◆ Paint
- ◆ Edging strip
- ◆ Mounting
brackets

module: two for the power, and one for the output which is fed to the Trinket.

The Trinket should boot cleanly into the code on power-up. Ours proved a little temperamental, so a power-on reset circuit was added consisting of a 22 µF capacitor charged slowly via a 100 kΩ resistor. A diode is included to discharge the capacitor when the power is removed.

TIME TO GET THINGS UNDER CONTROL

It is possible to use the PIR module to directly trigger the two MOSFETs. However, so we have a bit more control over how the lights go on, an Adafruit Trinket M0 has been included in the design. This tiny board can be programmed using Python to respond to the sensor and turn on our driver circuits.

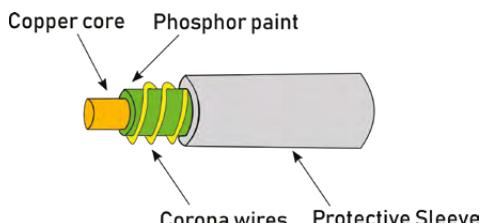
The first part of the code brings in the three libraries needed by the rest of the code. The 'board' library defines all of the constants for the Trinket, such as the numbering of the pins. The 'digitalio' library has the definitions for the objects that are used to interact with the digital input and output on

IS YOUR TAB ISOLATED?

Often the big metal tab on a device like a MOSFET is connected to one of the pins. This is definitely the case with the FQP30N06L, so we need to ensure they did not get screwed to the same heat sink. Check your device and circuit to ensure that screwing them to the case or to a shared heat sink will not cause a short circuit. A heat sink mounting kit will provide an insulating shim and sleeve for the bolt to keep the tab isolated from the heat sink.

WHAT IS EL WIRE?

EL wire is really quite simple in construction. A solid copper core acts as a conductor onto which a zinc sulphide phosphor is coated. A thin wire is wrapped around this, often called the corona wire. When a large alternating current is put through the two wires, the phosphor glows. Finally, an outer coating of clear plastic protects the wire from the elements and damage. Different colours are achieved either by doping the zinc sulphide with other chemicals, or by coating the wire with a fluorescent phosphor which glows in response to the light from the EL wire.



those pins, and the 'time' library gives us the ability to wait and check how long it takes for the code to run.

```
import board
import digitalio
import time
```

The next section configures each of our I/O pins, 0 and 1 as outputs to the driver circuits, and 2 as an input which is wired to our sensor.

```
ELHack = digitalio.DigitalInOut(board.D0)
ELHack.direction = digitalio.Direction.OUTPUT

ELSpace = digitalio.DigitalInOut(board.D1)
ELSpace.direction = digitalio.Direction.OUTPUT

sensor = digitalio.DigitalInOut(board.D2)
sensor.direction = digitalio.Direction.INPUT
```

The last section of the code is the main loop that runs constantly. This checks the input from the PIR sensor and turns on the outputs if motion is sensed. A `while` loop waits until for a period of time before checking for motion again. This ensures that the lights don't remain on for too long, but are not constantly flashing on and off.

```

while True:
    if sensor.value == True:
        #monotonic increments in seconds on the
        Trinket
        tstart = time.monotonic()

        ELHack.value = True
        ELSpace.value = True
        #On for 10 mins or until next motion
        triggers
        while time.monotonic() - tstart < (10*60):
            time.sleep(1)

        #Turn off if no motion or keep on if still
        movement
        ELHack.value = sensor.value
        # Set Space to same as Hack
        ELSpace.value = ELHack.value
    
```

BACK IN BLACK

The EL wire needs a support to hold it in place. So we are using 6mm MDF, painted black to emphasise the letters.

To create the rounded corners, we used a roll of tape to draw on a pencil line. The corners were then sawn off at 45 degrees. We used a surform to smooth the remaining wood, and finally it was sanded with a coarse sandpaper. Any holes or dents were smoothed over with filler, which was again sanded flat.

To form a case for the electronics, two pieces of MDF were laminated and a hole cut out using a drill and coping saw. We glued a third piece on top to form the lid. Holes were cut out for the wires and the corners rounded in the same way as for the main board.

MDF will absorb paint readily, so a primer is essential to stop the paint being patchy. After the

OTHER IDEAS

- Use more drivers and control each letter
- Use bubble writing for bigger and brighter letters
- Use LED strips instead of the EL wire for even bigger and brighter lights
- Alternate words by placing two wires on top of each other
- Use a Bluetooth module, instead of the PIR, so that the lights come on when a phone is detected

Silver edging designed for arcade cabinets was added around the outside to protect the edges and provide a smart finish

coat of primer, sand the paint back using a medium sandpaper to provide a key for the next layer. A second coat of primer/undercoat will help provide a smooth surface for the top coat. A thin layer of gloss paint was sprayed on and sanded with a 320 grit wet and dry paper. This was repeated until the surface began to look smooth, at which point the paper was swapped with 600 grit. Finally, as the surface began to look really glossy, a 1500 grit paper was used with very light sanding between coats.

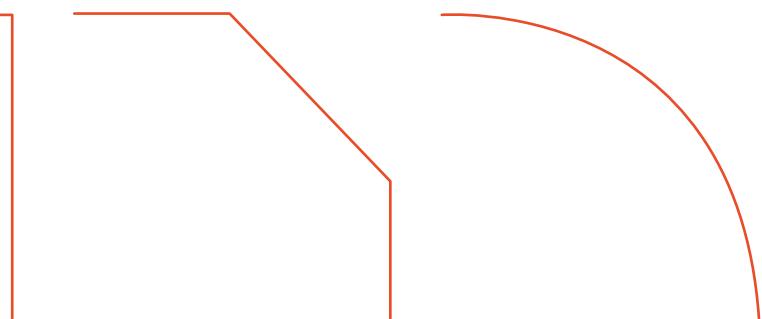
To support the wires, split pins were used. The wire was fed through the loops and we put tape over the back to avoid scratching the wall.

Silver edging designed for arcade cabinets was added around the outside to protect the edges and provide a smart finish.

Finally, a couple of hooks were added so that it could be mounted on a wall. □

QUICK TIP

Keep the wet and dry paper very wet, and keep washing off the paint so that it does not scratch the surface.



Square

Octagon

Circle

Left ♦
Three steps to round corners

HackSpace

TECHNOLOGY IN YOUR HANDS

Download the app

Out now for smartphones & tablets



£2.29

rolling subscription

£26.99

subscribe for a year



FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG
116

DIRECT FROM SHENZHEN: **MICROSCOPE**

When your feeble human eyes
can't see something small,
you need help

PG
122

CAN I HACK IT?



Robot toys – simple plaything
or the base for a project? We
pull one apart to find out

REVIEWS

124 NFC nail stickers

126 BeoCreate audio board

PG
118

BEST OF BREED

Stripped screw heads can be the bane
of restoration. We look at tools for
extracting them



128 Marty the robot

129 When Thea LED the way

DIRECT FROM
SHENZHEN

Microscope

View teeny tiny things for a teeny tiny price

By Ben Everard

 @ben_everard



Above ▶

The microscope stand allows you to hold the microscope at almost any angle

There's a whole range of uses for a microscope. Hardware hackers use them for soldering tiny parts, biohackers use them for watching tiny things grow, and bored students use them for looking under their fingernails, and then feel very, very gross. Microscopes range in price from quite affordable to extremely expensive. We set out to find out just what you get for your money at the low price range. For the princely sum of £11.54, we bought a 1000x magnification USB microscope from AliExpress seller, AIHOME Tech Store.

Now, before going any further, let's take a moment to think about that statement – 1000x magnification on a digital microscope. It doesn't really make sense. The output of the microscope is an image – well, more accurately a video stream. Displayed on a smart phone, this image is around 10cm across, on a large screen, it could be 100cm across, but has this increased the magnification?

ZOOMING IN

We'd love to be able to say exactly what the highest magnification was in terms of pixels per mm – however, we weren't actually able to take in-focus images at the highest (or even nearly the highest) magnification. There are a few fairly major design flaws that mean that this just doesn't work at a high magnification scope. The lens moves through the ring of LEDs providing illumination. As you zoom in, the lens moves from behind this ring to quite a long way in front. As you reach higher magnifications, the area that the microscope is actually magnifying is in shadow, and just black. Even with a second light source, we weren't able to properly illuminate the image at high magnifications.

The second major design flaw is that the stand is too flimsy to hold the device still enough to actually capture clear images at this level of magnification.

The high magnification, then, is more or less irrelevant, unless you're planning on building your own stand with lighting. However, there are loads of areas where microscopes can be useful at much lower magnifications.

There are two physical set-ups which do work, at least to some extent. The first is at low magnification in the provided stand (the lowest magnification is marked at 40x, though claimed on the website to be 50x, and we found that it worked out at 400 pixels per mm).

The set-up here doesn't look awful for, say soldering, but in practice, it's difficult to use. The light stand is very sensitive to knocks and vibrations and, as it's not stereoscopic, you don't get any sense of depth. In



short, it's not particularly easy to work in this set-up, but will do just fine for taking images and investigating objects up-close.

The second physical set-up is where the transparent plastic end is pressed directly onto the object you're looking to image, and the zoom is adjusted until it's in focus. This gives quite a high level of magnification (about 1600 pixels per mm) which works out as about 300x on our laptop screen. This set-up works well for investigating objects, particularly as this can be used in a wide range of physical orientations. The microscope can easily inspect something vertical, upside down, or in a tricky place. For this use, it's actually quite effective. The unit itself is easy to hold and manoeuvre with one hand, and the 1.5m USB cable gives enough slack for moving around. At less than 10cm long and 3.5cm wide, it's small enough to fit into a lot of tight spaces where better and more expensive microscopes simply couldn't go. This could save a lot of time by, for example, inspecting a PCB without having to remove it from its housing.

UP CLOSE AND PERSONAL

The device connects over USB and works as a webcam. This does open up the interesting possibility of controlling it via software which may allow, for example, time-lapse photography of something small-growing – though this would probably need a more solid stand than the one provided. The microscope worked without drivers straight away for us on both Windows and Linux environments.

Ultimately, our USB microscope is an interesting toy – and a great way of satisfying our curiosity for finding out what things look like up-close – and it's priced accordingly. £12 is a pretty reasonable price for something that allows you to poke, prod, and investigate small things. If you're hoping that this is a reasonable alternative to a serious scope for any kind of close-up work, you'll be sorely disappointed. For that, you'll need better optics and a better physical set-up – both of which come with a higher price tag. □

Above The scroll wheel along the top moves the lens in and out to increase or decrease magnification

DIRECT FROM SHENZHEN

**ONLY THE
BEST**

Broken screw extractors

Fear not the rusted screw head, help is at hand

By Ben Everard

 @ben_everard

Screws – they’re all around us, but we rarely think about them unless they go wrong. The all too familiar chunk-chunk-chunk as the Phillips or Pozi screwdriver fails to grip in the heavily-worn cross, or the more spectacular flick as a flat-head loses its purchase on a slit which is now a gaping chasm. There are things you can do to help avoid this – use good quality screws and a screwdriver with the right-sized bit (not one that sorta-kinda fits) – but sooner or later, you’ll probably still find yourself dealing with this problem, either through someone else’s carelessness or poor quality manufacturing.

It often happens on old screws, where rust both jams the screw in place and rots the head, meaning that you’re not just dealing with a screw with a stripped head, but one that’s firmly attached as well.

So, what do you do? Drilling out screws is arduous; we’ve had some success angle-grinding a new slit in screws; even welding a bar on for increased purchase can work. However, sometimes it seems like the only option is to reach for the crowbar and solve the problem with brute force.

What if there’s a simpler way? We put three tools, designed for this problem, through their paces to see what works and what doesn’t. In our test, we jammed three screws securely in a block of wood and tried to remove them using the different tools. None of the methods actually uses the indent in the head of the screw, so it doesn’t make a difference how stripped the head is.

While all the products we’re looking at are sold as screw extractors, they also work on bolts and other protrusions that need twisting.



Neji-Saurus PZ58

Pliers with a twist to help you unstuck stuck screws

ENGINEER ⚡ £16.29 | engineer.jp

This pair of pliers has a specially designed tip for gripping screws. The main difference is that, whereas the grooves in the tip usually run parallel to the tip of the pliers (allowing you to grip tightly as you pull something), in these tools, they run perpendicular to the tip which gives more strength when turning. There's also a curve in the tip that makes it better at gripping round screw heads. This does slightly reduce its utility as a pair of pliers (though it's still good for gripping larger objects and cutting wire). These pliers work best on small screw heads (under 5mm), as the screw-gripping part is slightly recessed.

Our biggest concern with these pliers is that their biggest innovation is rotating the gripping lines on the pliers by 90 degrees. This can simply be replicated by rotating a regular set of pliers by 90 degrees, and gripping the screw with the side of the jaw rather than the end. While there is a slight additional

increase in grip by the curve of the jaw, and it is a little easier to use, there's not a significant benefit in these unless you're removing a large number of broken screws, or you're working in limited space.

In our test, these were only able to remove one of the three jammed screws, and this was the least jammed. We weren't able to put enough force on the larger and more stuck screws to remove them. We were able to get exactly the same result with regular pliers turned sideways. Also, while this was the easiest of the three to use, it actually had the weakest grip. As such, it's only really useful in cases where the screws aren't stuck.

The area where the Neji-Saurus really excels is removing screws with unusual heads. When they're not stuck, the grip should be enough to remove a screw that's done to a moderate tightness, and this is the only tool on test that can both remove and tighten screws, so it's useful when you want to put the screw back in again. □



Left ⚡
The spring-loaded pliers are easy to use with one hand

VERDICT
Only slightly more useful for removing broken screws than regular pliers. Best suited to small screws in limited space.

6/10

BEST OF BREED

Irwin Power-Grip Screw Extractors

A socket set for screw-mergencies

IRWIN £33.89 | irwin.com

This can be thought of as a socket set with teeth to bite into the screw head. They work exactly like sockets – just select the right size for your screw head, pop it over the jammed screw, and twist. If all goes to plan, the screw will pop out with remarkable ease.

The marketing claims that a single set will work on all screws – both metric and imperial. This is because the grips inside the socket are slightly conical, so you just push them on far enough to grip and then turn. At least, that's the theory. In our test, this completely

The Power-Grips are the most expensive option on test, but they're also the strongest

failed to remove any screws because the heads were too close to the wood to push the socket far enough over to lock on. The screws weren't countersunk, just screwed in until the base of the head was flush with the wood.

When we slightly unscrewed the screws, the socket was able to lock onto two of the screw heads (the third was made of metal too tough for the teeth to bite into), but this is obviously of no use when you're dealing with a really jammed screw. We suspect that these might be more useful for removing bolts with their corners rounded (yes, we're looking at you, people who use imperial tools on metric bolts and vice versa) – a use which they're also sold for – but we're looking at screw extractors in this test.

The Power-Grips are the most expensive option on test, but they're also the strongest. Once the Power-



Above ▶ The Power-Grip extractors should fit a wide range of screws

Grip has gripped onto the errant screw (that is, if you can get it to grip), it's jammed solidly and you can apply a large amount of force (we used a power drill at slow speed). When we say 'large amount of force', we mean in the context of a screwdriver. Using something like an impact wrench at full throttle could supply too much force and cause damage.

Although these are sold as screw extractors, we can only really see them as useful for removing chewed up nuts or bolts, where there should be more of the fastener protruding from the joint. ☐

VERDICT

Completely failed to remove any screws, but may be useful for bolt extraction.

3 / 10

SKMoon Damaged Screw Extractors

Double-team action to tackle stubborn screws

SKMOON £6.99 | hsmag.cc/gmidAb

These are perhaps the most involved tools we're looking at here. There's a set of four bits, each with two ends. On one end is a drill, the other is an extractor. Simply select the bit that's the right size for your screw (the four cover the range from 3 mm to 12 mm), and first drill down slightly into the screw, then flip the bit over, and use the extractor to twist it out.

There are a range of different sizes in the box, and the larger ones are able to exert more force than the smaller ones. However, if you use too large a bit, you risk drilling out the head of the screw, leaving it with nothing for the extractor to bite into. This is all complicated by the fact that you can't see how big the head and body of a jammed screw are. We found that starting with a small bit, and gradually moving up until one removed the screw, worked well.

While all the tools we're using here will damage the screws, this one will completely obliterate them. Usually this isn't a problem, but in some cases you may want to reuse the screw (perhaps they're an unusual size and you're confident that you won't have to replace them again). In this case, you'll simply have to avoid this option.

The SKMoon Damaged Screw Extractor was the most successful for us, and removed two of the three test screws. The third proved to be made of metal too tough to drill into, so we weren't able to attach the extractor. In practice, such a tough screw is unlikely to end up with a stripped head, so you're less likely to actually need an extractor. This is useful because none of the other extractors were able to get this screw either.



Left The small size and carry case makes this the most portable of the options

" The SKMoon Damaged screw extractor was the most successful for us, removing two of the three test screws "

Overall, this is the only method we can really recommend for removing damaged screws, as it was the only method tested to get a better than 50% success rate. There are a wide range of manufacturers making very similar products. We selected the SKMoon for our test because it's the most popular on Amazon. □

VERDICT

The most successful screw removers we tested – the best option for getting you out of a jam.

8/10

Can I Hack It? A Tankbot

Can a small robotic tankbot be hacked into a project?



Les Pounder

@biglesp

Les Pounder is a maker and author who works with the Raspberry Pi Foundation to deliver PiAcademy. He also helps teachers/learners to become creative technologists. He blogs at [biglesp](http://biglesp.com).

Right ♦

Lean, green, and not very mean. Our tiny Tankbot is a neat and rather cute robot that can explore the world using sensors, or be controlled via an app.

YOU'LL NEED

♦ **Desk Pets DP-TB-1851-GR Tankbot Green Toy Robot**

COST

£17.30

WHERE

hsmag.cc/JTbybV

Pi Wars has recently taken place in Cambridge, and there we saw robots of all shapes and sizes do battle in the arenas and courses created by the Pi Wars team. Robots are cool, but you don't have to start from scratch. There are so many on the market, so we popped a cheap robot into our shopping basket. Just how easy are they to hack, and can we use them in a project? Well, there's only one way to find out. Let's take one apart!

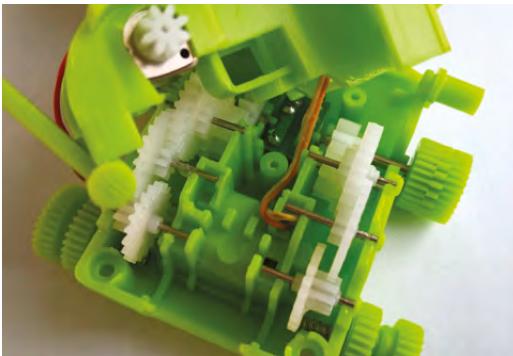
GENERAL CONSTRUCTION

Made of a thin, slightly pliable plastic, the upper outer shell can be easily worked with conventional hand tools. The upper part of the robot chassis also features two smoked plastic lenses, used to decorate the robot and provide a diffusion for two blue LEDs inside the 'head'. At the front of the head are two infrared sensors used to detect objects. The lower part of the chassis is a slightly firmer plastic, equally pliable, and can be worked with hand tools. On the bottom of the chassis we see an access port, which one would suspect would house a battery. Unusually, this port covers a small DC motor, which is one of two motors that provide movement for the robot.

The robot has a caterpillar/tank track used to drive it across most surfaces. Each motor provides power to a single track via one central wheel, while the other wheels merely move due to the teeth present in the track, catching the teeth in the cog-like wheels. The tracks themselves are a flexible rubber-like plastic that fits securely around the wheels.

So how can we take it apart? Well, firstly remove the security screws from the underside of the chassis. These screws are the same as those found on McDonald's toys. Next, remove the same security screws from the wheels. Now, move to the upper part of the chassis and using a 'spudger', or flat screwdriver, pop out the smoked plastic from the sides of the robot head. On one side of this you'll see revealed two cross-head screws; remove these and you are safely inside the robot.





Above

There are two DC motors tucked away in the lower section of the chassis. Each motor has gearing to drive two more wheels at the correct speed

POWER

Charging the robot is easy, largely thanks to a built-in USB A male connector. This plugs in to any spare USB port and takes around an hour to charge the 70 mAh 3.7V lithium polymer battery. The battery gives the robot around 20 minutes run-time. There is a charge control circuit built in to the main circuit for the robot.

ELECTRONICS

Our robot has multiple modes. Mode 1 enables the robot to detect the surroundings using the infrared sensor. Mode 2 uses sound and light sensors to detect a person or torch light and follow the sound/light. The third mode uses the included controller, along with an app on our phone, to control the robot. The controller sends signals via infrared pulses, so when a control is pressed on the mobile device, the appropriate signal is sent to the robot using the two IR LEDs in the controller.

Inside the robot is a custom circuit board, which has a central chip that has been unhelpfully covered in epoxy resin, so we cannot identify the chip used. But all hope is not lost, as all of the pins from the chip are broken out to test points around the board, so those with a steady hand and an oscilloscope can dig deep into the chip and learn its secrets. Two of the most obvious connections are those for the motor and these are at ML+ and ML- for the left motor; MR+ and MR- for the right. These connections go directly to the DC motors in the lower part of the chassis and, as the motor can be controlled in two directions (forward and backwards), we can conclude that there is a motor control chip with an H bridge, a chip that enables the polarity of the motors to be switched on command. Also present on the board are two connections for the attached speaker, used to generate pleasant beeps for our happy robot. But hidden on the underside of the board is something unexpected. Under the robot is what looks like an I²C connection – it has the usual VCC and GND, and SDA, but rather than SCL (clock),

PLAYING WITH TOYS

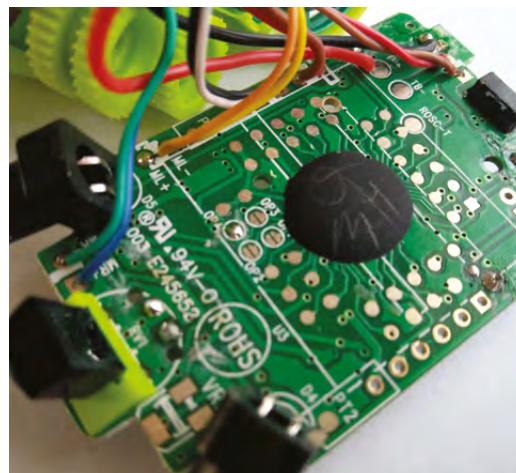
Toys are an excellent source of knowledge. They provide us with the raw materials for our projects. In this case we have motors and a chassis that we can repurpose. They also have circuits that we can debug and learn from. There are many different toys that can be hacked and reverse-engineered to understand how they work. If the toys run at a suitable logic level – the voltage that they run and control at – then we can use devices such as Raspberry Pi, micro:bit, and Arduino to add new features and control them in new ways.

So do we rush out and buy new toys to take apart? Well no! Look for old broken toys from charity shops / Goodwill / flea markets etc. They will often be very cheap and quite broken, but using your knowledge of electronics and a little patience, these toys can once again be used for further adventures!

we have SCK, which we assume is a reference to clock. So what does this mean? Well, connect this up to an Arduino and start sniffing the I²C bus to see what information goes to and from the robot.

CONCLUSION

This is a fun little robot to hack. It has been built with modular components and a sensible circuit board that enables us to poke around and learn all about it. Sure, the chip may be hidden under epoxy, but we can still use the test points to understand the various functions of the robot. Using an Arduino or another device with I²C, we can see deep into the robot and learn how it's controlled. There's even the possibility of controlling the robot using this method. If we removed the custom circuit board, then there is space for a small Arduino clone board; in this case 4tronix's 4duino Mini Pro would be an excellent choice, as it has a built-in DRV8833 motor controller. Taking the guts of this robot, we can also repurpose the app and controller to control other devices; for example, a relay to control much higher voltages (obviously only if you know how!) or use the sensors to trigger alarms. □



Left

This tiny board is packed with components and connections for I²C and test points for the main chip. A rare occurrence, but most welcome!

Pimoroni NFC nail stickers

Light up your manicure with NFC

PIMORONI ♦ £5 | shop.pimoroni.com

By Sophy Wong

 @sophywong

Below ♦
The nail stickers can sit on top of polish to give a coloured, light-up nail

Part nail art, part biohack, these NFC nail stickers from Pimoroni are awesomely cyberpunk. Each sticker is a tiny flexible circuit that contains an NFC antenna and a tiny LED; when placed near an NFC signal, the LED lights up. This ultra-compact package, with no battery or exposed wiring, is a perfect fit for nail art!



NFC, or near-field communication, is a subset of RFID technology. Electronic devices with NFC capabilities can communicate with each other wirelessly across a distance of about 4 cm or less. This technology is used for contactless payment systems like Apple Pay and Android Pay, and for ID badge entry systems. Many smartphones now come with NFC capabilities, and makers have been using NFC components with Arduino to build creative projects, like keyless locks and password vaults. NFC technology is spreading quickly as an easy and secure way to pass data from one device to another, but power can be transmitted as well. That's how these LED nail stickers work: power comes from the electromagnetic field generated by an NFC signal.

OUR EXPERIENCE

For £5, you get one hand's worth: five nail stickers with one LED on each. All five stickers are the same size: 11 mm long by 9 mm wide. This size should fit well on most adult thumbnails, but using these on smaller nails will be a challenge. They'll fit easily on larger acrylic nails, which is how we've seen similar products styled. Applying the stickers was fairly easy, though it took a few tries to get a smooth application. In our tests, we secured the stickers with several coats of clear gel polish, which held nicely for about five days. Without a coating, the adhesive held

for about 24 hours, before peeling up around the edges. A few drops of acetone easily removed the sticker, which is not reusable.

When lit, the LEDs are bright for their tiny size, but they definitely look best in low-light situations. Placing the stickers near an NFC device, like a smartphone with NFC enabled, makes the LED shine nicely. However, if you don't have an NFC smartphone, you'll have to find NFC fields out in the world to wave your hand in front of. This means your nails may only glow when you're paying for something at a store, or badging in at your office. If you're out for a day of shopping, this could actually be pretty often – you can bask in the glow of your manicure whenever you check out at a 'tap to pay' machine.

GOING FURTHER

Of course, we're makers, so naturally we decided to build our own NFC station. For this, Pimoroni suggests using the stickers with an RFID/NFC shield for Arduino (f40) from Adafruit. We paired the shield with an Arduino Uno. With some handy code from Adafruit's quick-start guide, we had our own NFC reader up and running: passing the stickers within 2.5cm of the shield made the LEDs glow satisfactorily bright. The shield is about 54mm by 118mm, and is not designed for wearable applications, so if you're

thinking of cosplaying with these nail stickers, you'll have to cleverly build the Arduino and shield into some kind of hand-held prop, like a book. The LEDs need to be very close to the antenna on the shield to glow, and too much material over the antenna can obstruct the field, making this a challenging application.

It's important to know that these NFC stickers cannot be written to or read – they simply light up when placed inside an NFC field. This may feel like a bit of a dead end, but they could still be useful as wireless lights in projects. If nail art isn't your thing, you could stick these to objects, or even embed them in resin. As long as the sticker can be placed near an NFC signal, the LED should light up, and the effect is magical.

NAILED IT

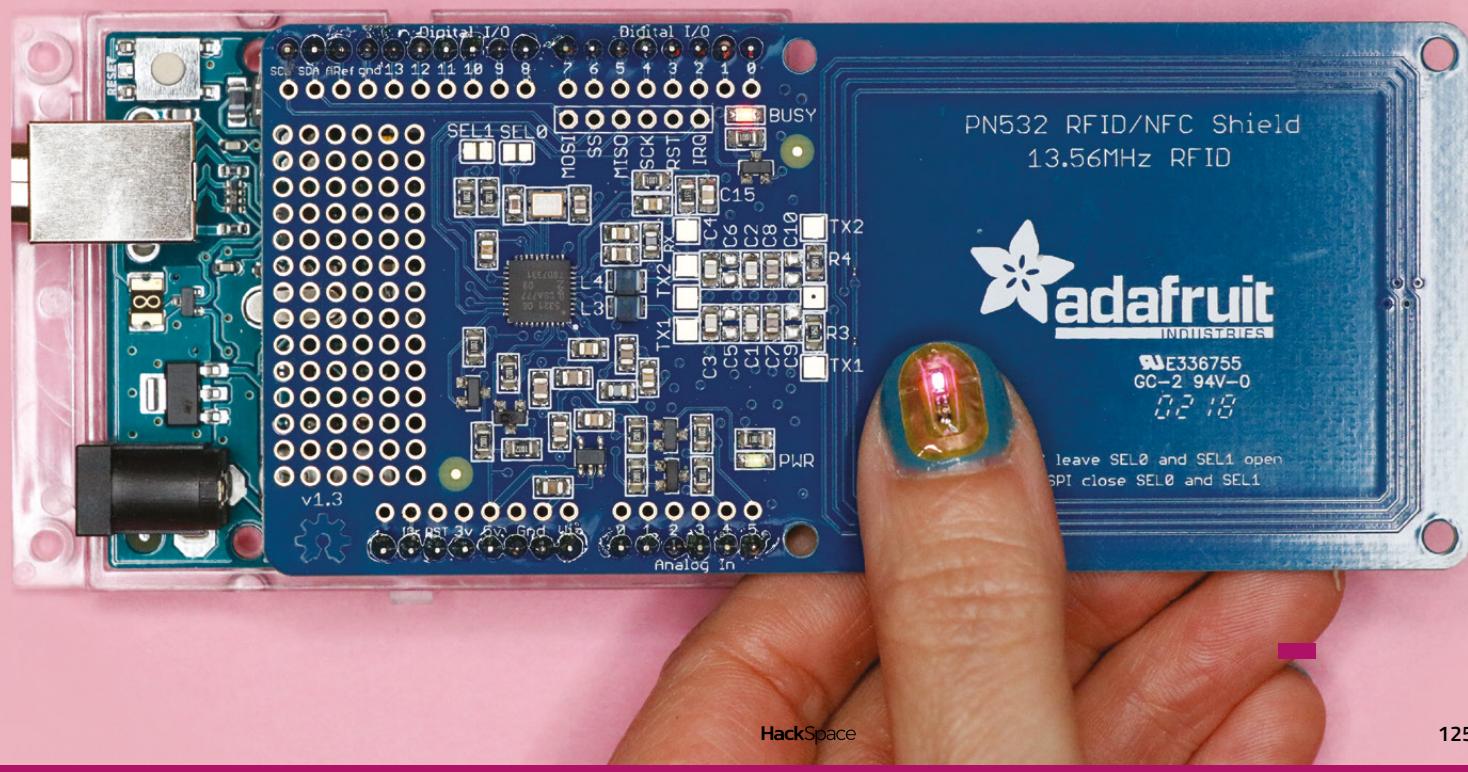
Tinkering with these nail stickers sparked our curiosity about NFC and RFID, and this author really enjoyed hunting down NFC fields while out and about. It was a sci-fi moment every time the LED lit up, and it revealed many invisible electromagnetic fields we pass through every day. While the stickers are a bit of a novelty without the ability to hold or transmit data, their low price and high impact makes it easy and exciting to get started with NFC and wearables. □

Below The nails need a source of NFC power, such as a mobile phone or NFC development board

VERDICT

Cyberpunk jewellery, NFC field-seeker, or pure curiosity. These stickers are great fun and easy to use.

8/10



BeoCreate

Upcycle your old speakers with WiFi and Bluetooth

BANG & OLUFSEN AND HIFIBERRY ♦ £149 | Hifiberry.com

By Ben Everard

 @ben_everard

The BeoCreate is a four-channel amplifier for adding WiFi and audio streaming capabilities to old speakers. As well as the BeoCreate board, you'll need a power supply and some 3D-printed parts. If you want to include Bluetooth and wireless functionality, you'll also need a Raspberry Pi and microSD card. The full working setup comes in at £181.80 – this is just for the equipment to upcycle the speakers and doesn't include any actual speakers

Below

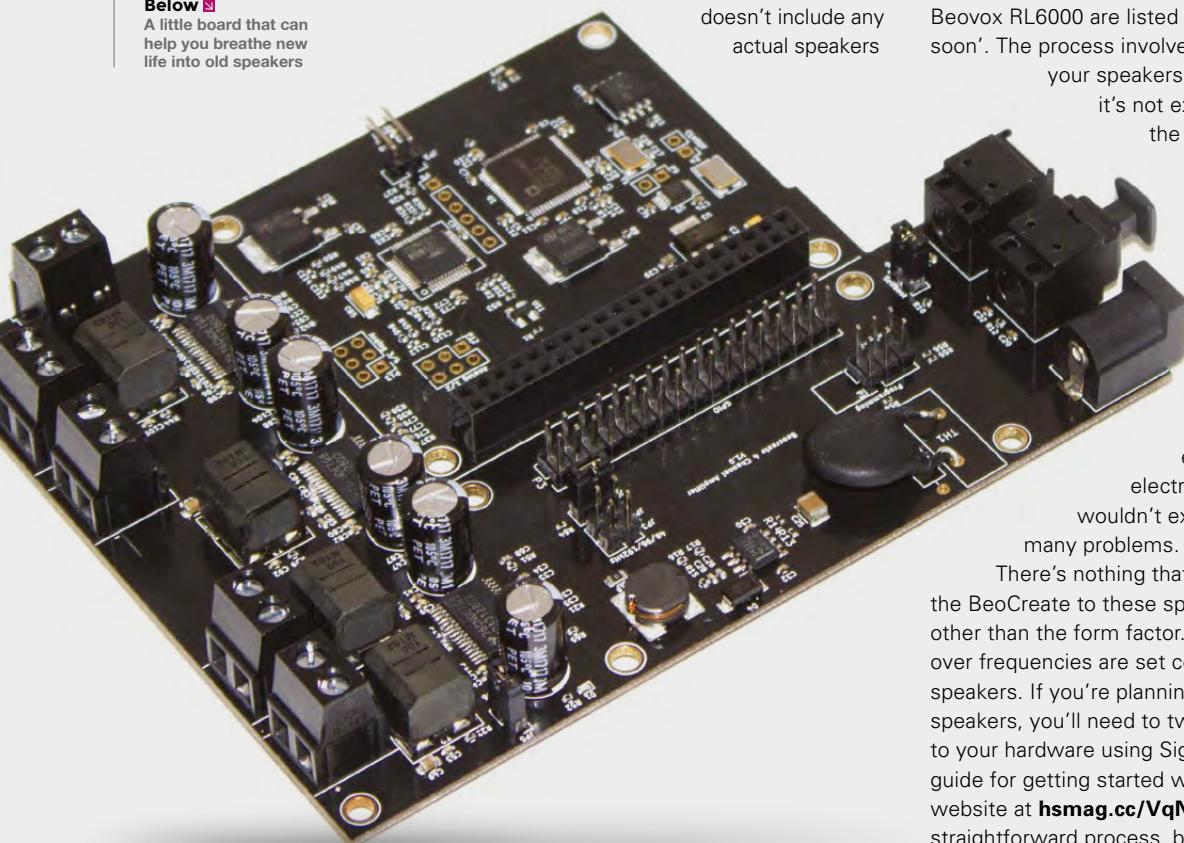
A little board that can help you breathe new life into old speakers

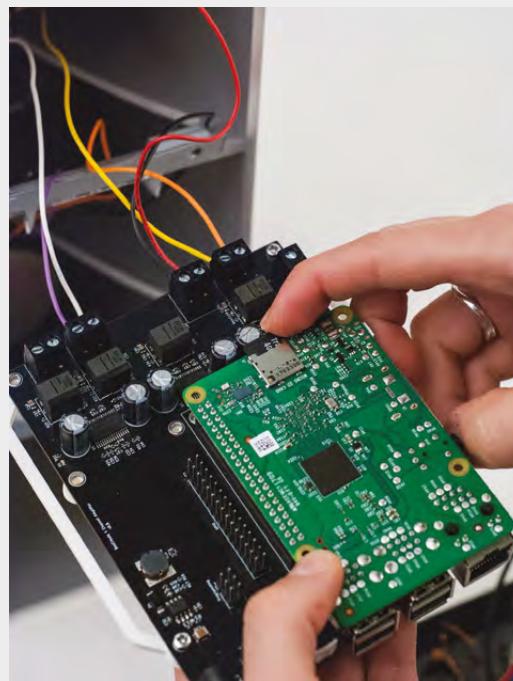
themselves. This is obviously not cheap, but high-quality audio hardware rarely is.

Altogether, this provides essentially all the hardware and software you need for creating a high-quality wireless speaker, except the actual speaker and enclosure. This is because it's designed for upcycling old Bang & Olufsen speakers to give them the ability to work with modern music sources.

HiFiBerry has detailed guides on how to use the BeoCreate to upcycle the CX100 and CX50 speakers, and guides for the Beovox RL2000 and Beovox RL6000 are listed on the website as 'coming soon'. The process involves taking a hacksaw to your speakers and cutting wires out, so it's not exactly plug-and-play, but the guides are detailed and include pictures of all the steps to take you through. We really wouldn't recommend this for beginners – mostly because of the cost of making a mistake – but if you've got some experience tinkering with electronics and soldering, we wouldn't expect you to encounter many problems.

There's nothing that fundamentally ties the BeoCreate to these specific speakers, other than the form factor. However, the crossover frequencies are set correctly for the B&O speakers. If you're planning on using this in other speakers, you'll need to tweak the frequencies to your hardware using SigmaStudio. There is a guide for getting started with this on the HiFiBerry website at hsmag.cc/VqNNoJ. It isn't an entirely straightforward process, but it does mean that you'll





be able to adapt the hardware to perfectly suit a wide range of different speakers.

VERSATILE SYSTEM

With four channels (each of which can drive a 4–8 ohm speaker), you can power two double speakers, but if you need more than this, you'll need additional hardware. You can do this by using the fibre-optic TOSLINK input and output to daisy-chain multiple BeoCreate boards together.

The BeoCreate can provide up to 180W of power (via two 30W and two 60W outputs). This audio is fully controllable from a Raspberry Pi via a four-channel digital-to-analogue converter with sample rates up to 192 kHz. As it uses its own DAC, the sound quality is better than systems that rely on the audio coming directly from the Raspberry Pi's audio out.

In addition to hardware, there's a suite of Raspberry Pi software that allows you to play directly from AirPlay, Bluetooth, or Spotify, as well as upload configurations to the Digital Signal Processor (DSP) on the BeoCreate. You don't have to use this software (or indeed, the Raspberry Pi), as the board can work as a standalone amplifier.

While you could probably put something similar together from other parts, the BeoCreate is the most versatile of the integrated high-end solutions we've seen. However, the downside of getting a fully integrated solution like this is that you may not need the full range of bits. For example, if you've already got an amplifier for your speakers, it will be far simpler and cheaper to add a Bluetooth or Spotify

input than to rip out the entire electronics system and replace it with a BeoCreate.

BeoCreate is designed for upcycling old speakers, but we also think it's a useful option for using as the basis of a homemade stereo. Pick your speakers, build your own housing, and you can add a BeoCreate and Raspberry Pi to create a custom sound system. The software behind it is open-source, so, for people who like fiddling with audio, there's a lot of tinkering to be had. The DSP can be programmed over the network, so you can hack your sound settings without having to physically connect to the speakers. For those who prefer to listen to music than tweak endless audio settings, the defaults are good enough for most people, other than audio geeks.

GOOD SOUND

Both Bang & Olufsen and HifiBerry have a history of producing great audio products, so it should come as no surprise that the BeoCreate sounds great. We're not an audio magazine so we'll avoid using adjectives that are wholly inadequate at describing sound such as 'open' or 'smooth', but we were impressed with the sound quality. When matched with good speakers, the BeoCreate can fill a room with clear, clean sound (or Guns N' Roses if you prefer).

There are a wide range of high-quality speakers sitting in lofts and cupboards around the world. Many of them have made their way onto internet auction sites, and many can still produce high-quality audio. Not only is this a great way of hacking together your own high-quality speakers, it's a great way of keeping this old gear out of landfills. □

Far Left When a Raspberry Pi slots into the headers, the BeoCreate provides power to the Pi, so only one power source is needed

Left The BeoCreate fits perfectly into supported speakers, but other speaker models may have less space

VERDICT
A powerful and high-quality amplifier and control board for wireless speaker projects, with a price tag to match.

L 9/10

Marty the robot

One small step for Marty, one giant leap for roboteers

£150 | robotical.io

By Lynn Andrews

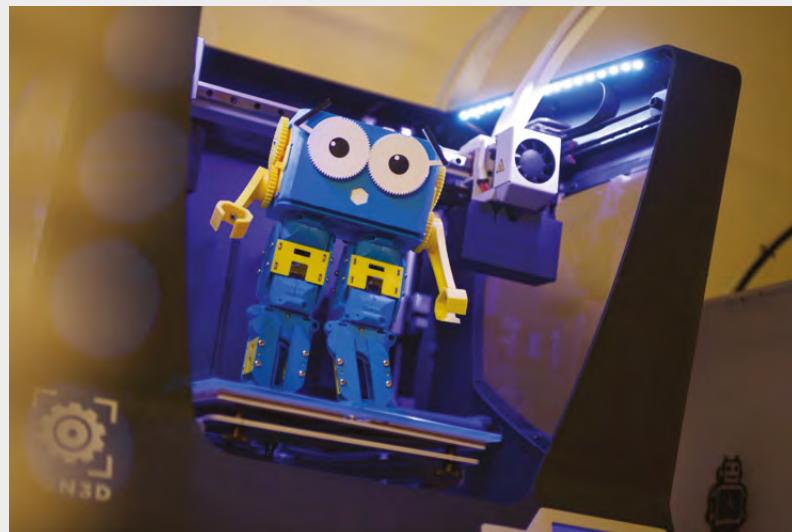
 @STEMBananas

Marty the robot is an entry-level walking robot, marketed at parents and children, the education sector, and makers. Its body structure is made of rigid plastic which comes in a kit for self-assembly. There's a control board (called Rick), in Marty's head and there is room to add a single board computer such as a Raspberry Pi.

You can program Marty via a WiFi connection using Scratch, Python, or C++ languages. Also, if a suitable single board computer is added (such as a Raspberry Pi), ROS (Robot Operating System) can be used. Furthermore, Marty can be simulated using Coppelia Robotics' V-REP v.3.4.0+ simulator.

Marty's design can be downloaded if you want to 3D-print your own, and it's designed to be extensible with other 3D-printed parts. The associated control board and servo-motors are sold separately to accommodate self-printing.

We found the assembly process straightforward and the instructions clear. There's a check diagram to make sure you have assembled the legs correctly,



which is a nice detail. It feels like the design is well thought out. We do have slight concerns over the robustness of the robot – if deposited in a class of eight-year-olds, it might be good to plan for some maintenance and replacement parts at some point. For makers, this is less likely to be an issue.

The hands that it comes with are more aesthetic than functional so, once familiar with the basics of programming Marty, developing new hands would be an interesting next step. The legs are controlled in a slightly unusual way, with three servos, plus some springs to give a slightly wobbly walk.

Marty offers an intuitive way to start learning programming as the coding has a three-dimensional physical context, the robot. The product is easy to pick up initially, but has enough depth to enable learning progression over time. From an educational point of view, Marty can be used to introduce children to coding, simple robotics, design, and manufacture. The open-ended nature of the product allows children to be active developers rather than passive learners.

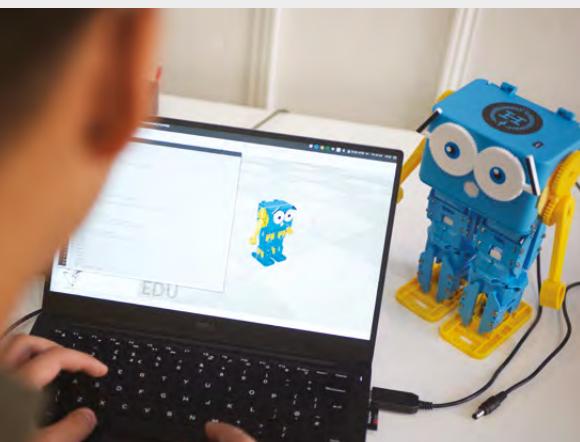
For any aspiring amateur robotics engineers, Marty offers a starting point for developing your skills in coding and mechanics, helping you grow your ideas. For makers, Marty offers an easy to program vessel for any build that needs two legs, whether that's a robot butler, an AI-powered cat repeller, or anything else. □

Above

As the 3D printer files are available, you can build your own Marty (though not in one go!)

Below

You can simulate Marty using V-REP and Robot Operating System (ROS)



VERDICT

A fun way to learn coding and start conceptualising your robotics ideas

8/10

When Thea LED the Way

C Penning ⚡ £30.00 | chibitronics.com

By Richard Smedley

 RichardSmedley

While there are fewer barriers nowadays to getting children involved in technology, it often takes something striking to spark that initial interest. *When Thea LED the Way* deftly weaves its electronic education into a tale of injustice, as well as discovery, as the reader constructs circuits to help star searcher, Thea Prom, bring light to her darkened world.

The book is a thing of beauty: landscape format, traditional illustrations with a contemporary twist, and ring-bound, with a tactile cover similar to that of the box, which Chibitronics has filled with electronics. As well as the craftbook, you get a roll of copper tape, two 3V batteries, a conductive sheet, two battery clips, and 18 LED stickers.

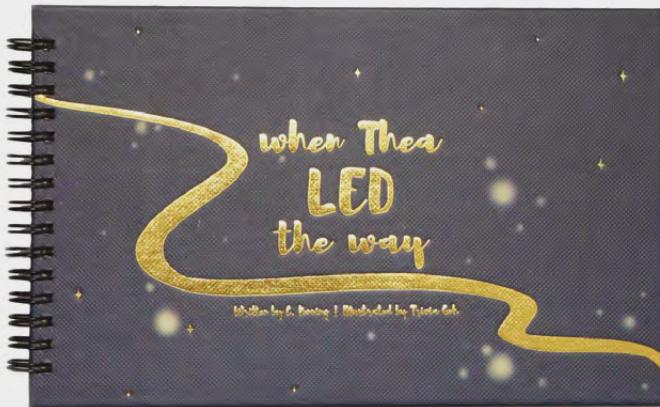
Made by Chibitronics, and initially crowdfunded (achieving 600% of its target last autumn), the kit is available in the UK from Pimoroni, and internationally from Chibitronics, and via Amazon.

Children's stories often benefit from real threat of mythic proportions. Here, Thea Prom – whose name is a play on the Titan Prometheus, who stole fire from the gods and gave it to humanity – searches for light throughout the book to heal her father and the other villagers, and must use ingenuity to prevent it being snatched away by those who would keep it for personal gain, as Lenny Lender forces villagers to work and pay for essential light.

Finding learning in books is natural – but here it's embedded physically as well, as you unwind sticky copper tape, carefully bending it to stick on tracks in the book, or clip in a battery to the book's opening corner, and light up an LED sticker. Penning has crafted an involving tale, as well as a great introduction to the basics of circuits. You will finish with several circuits to go back to, and knowledge you can take to other electronics kits.

It's not just electronics, or even the downsides of a debt economy, that are taught here – switch logic appears in the final section to leave the reader with the foundations for Boolean logic.

The underlying story of discovery, persistence, and inventiveness, as well as the beautiful presentation, will make this a great gift for any child of primary school age. ☺



VERDICT

Uses a strong narrative to show the right way to explore and discover – with persistence – as well as teaching children circuit basics.

L 8/10

next mag

issue
#8

ON SALE
21 JUNE



FEATURING **TOOLS! TOOLS!** **TOOLS!**

OUR PICK OF THE **BEST**
TOOLS AROUND

ALSO

- OPEN SPACE AGENCY
- ARTIFICIAL INTELLIGENCE
- GLASS-BLOWING
- ARDUINO
- AND MUCH MUCH MORE



DON'T MISS OUT

hsmag.cc/subscribe

HackSpace

TECHNOLOGY IN YOUR HANDS



WEAR PROTECTION!

Toxic chemicals will **NOT**
give you superpowers

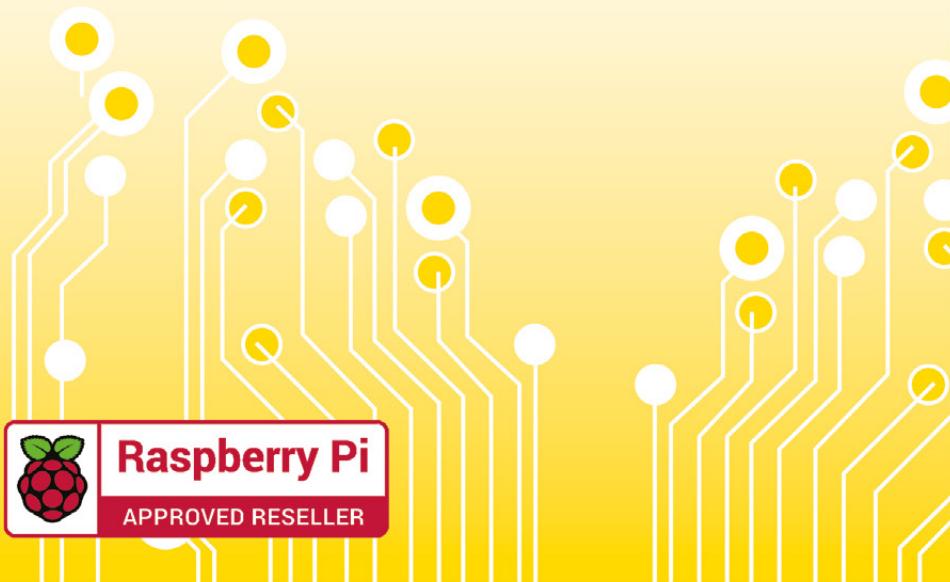
— hsmag.cc —

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*



Available for worldwide shipping at:

WWW.CANAKIT.COM

\$74^{.99} **£59^{.99}** **€64^{.99}**
US DOLLARS EXCLUDING VAT EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation.
CanaKit is a registered trademark of Cana Kit Corporation.