MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.cc | January 2023 | Issue #62

RASPBERRY PI PROJECTS

DIY builds with the world's favourite TINY computer

RADIO CONTROL YOUR BUILDS

CIRCUIT PYTHON WITH PICO W

3D-PRINT PROJECT BOXES

DIY MUSICAL LIGHT SHOW

Jan. 2023
Issue #62 £6
62
9 772515 514006
Raspberry Pi PRESS

ARDUINO **SPAGHETTI** ROBOTS **CUTLERY**

Welcome to HackSpace magazine

Raspberry Pi computers have changed the landscape of making in many ways. They were the first widely available, cheap, small computers with accessible GPIO pins. This meant that for a relatively small amount of money, you could embed a Linux computer in your builds, and get it to control the electronics. This might seem like an ordinary thing now, but it wasn't until Raspberry Pi did it. It was revolutionary. The fact that it seems ordinary now is a testament to how effective Raspberry Pi has been in making the technology accessible to so many people.

The fact that it seems ordinary now is a testament to how effective Raspberry Pi has been in making **the technology accessible to so many people**

This issue, we're celebrating this seismic shift in the maker world by looking at some of our favourite Raspberry Pi projects spanning the full range of possibilities from this tiny board.

BEN EVERARD
Editor ben.everard@raspberrypi.com

PAGE 36
FREE PICO W
WHEN YOU SUBSCRIBE

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

get in touch at **hsmag.cc/hello**

GET IN TOUCH

hackspace@raspberrypi.com
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Olivia Mitchell

Photography
Brian O'Halloran

CONTRIBUTORS

Mike Bedford, Marc de Vinck, Andrew Lewis, Rob Miles, Phil King, Nicola King, Rosie Hattersley

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

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

SUBSCRIPTIONS

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

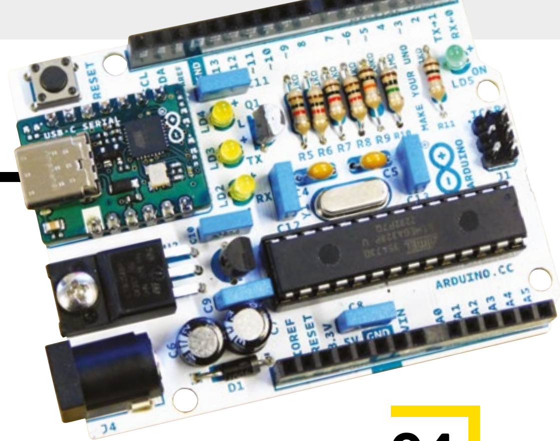
To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

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

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

ISSN: 2515-5148.



94

Contents

06 SPARK

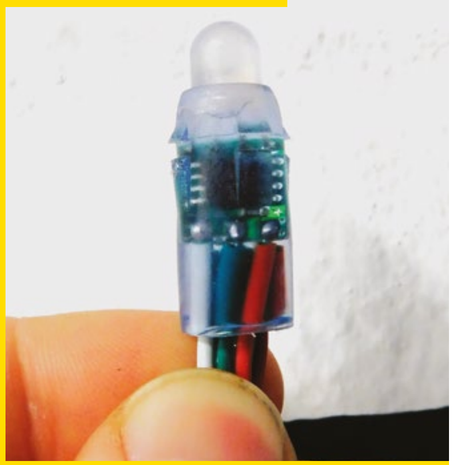
25 LENS

Cover Feature

26

Tutorial

Control many LEDs



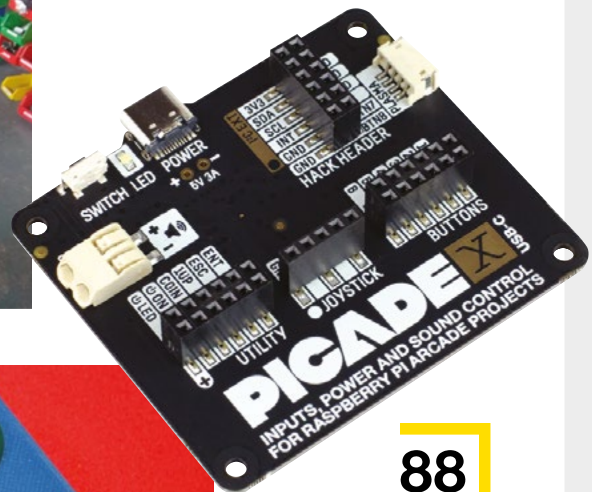
60 More light-emitting diodes than a single microprocessor can handle

52

96

Meet the Maker
Joseph Herscher

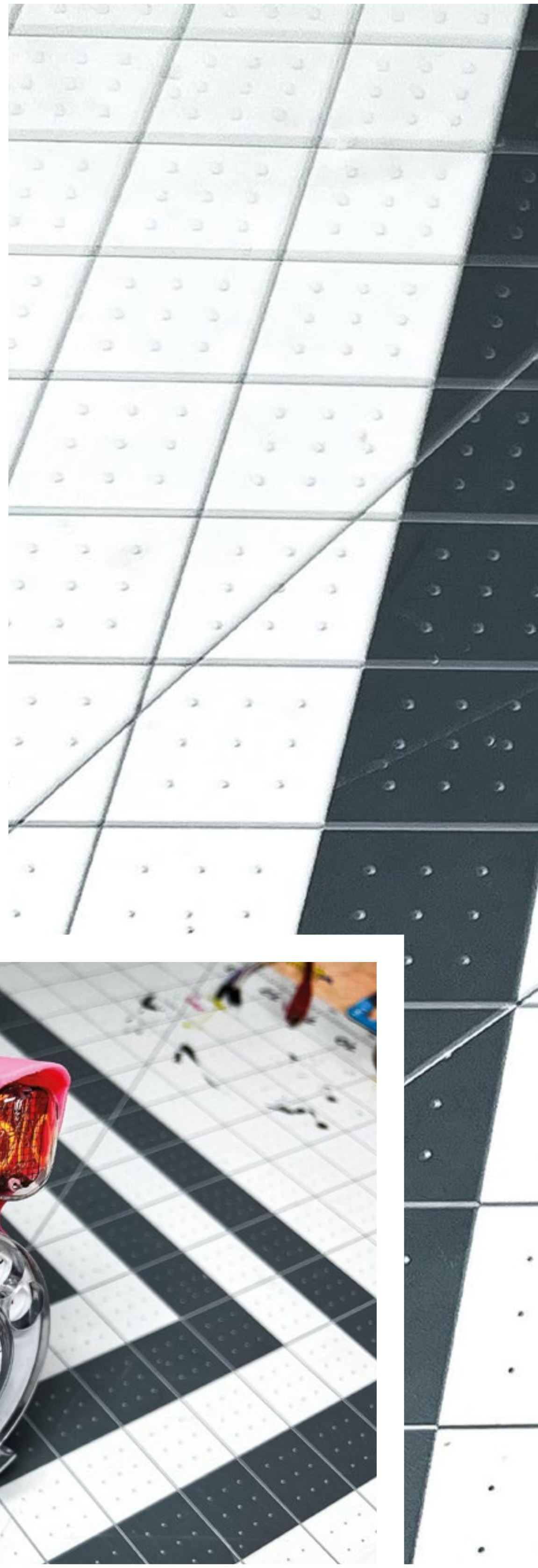
56



88

Tutorial
Custom cases

06



Rŏta

By **Architeuthis Flux** hsmag.cc/Rota

This thing is unbelievably beautiful. What does it do? The best answer is that we don't care; truth is beauty, beauty truth, nothing else matters. But if you must insist on more information, it's a device for counting. The maker built the first one as a gift for a duck hunting lodge, but since then, he's had interest from many more people, and has improved the design up to the current iteration, version 3.

The device uses a few esoteric parts: at its heart is an AVR64DD28 microcontroller; it also uses three IN-12B Nixie tubes, an NCH8200HV Nixie power supply, and an Automatic Electric Type 24 rotary dial. There are a range of kits with varying amounts of components included, and they're selling like hot cakes.

Right
Counting things never looked so lovely


GHI
4
DEF
3
JKL
5
ABC
2
MNO
6
1
PRS
7
TUV
8
WXY
9
OPERATOR
0
New
Day
Battery
Voltage
Begin
Stopwatch
Enter
Count
Show
History
Play
Golf
Day/Total
Display
Subtract
Clear
Enter
Menu
Röta

VFD Layer Clock Soldering Kit

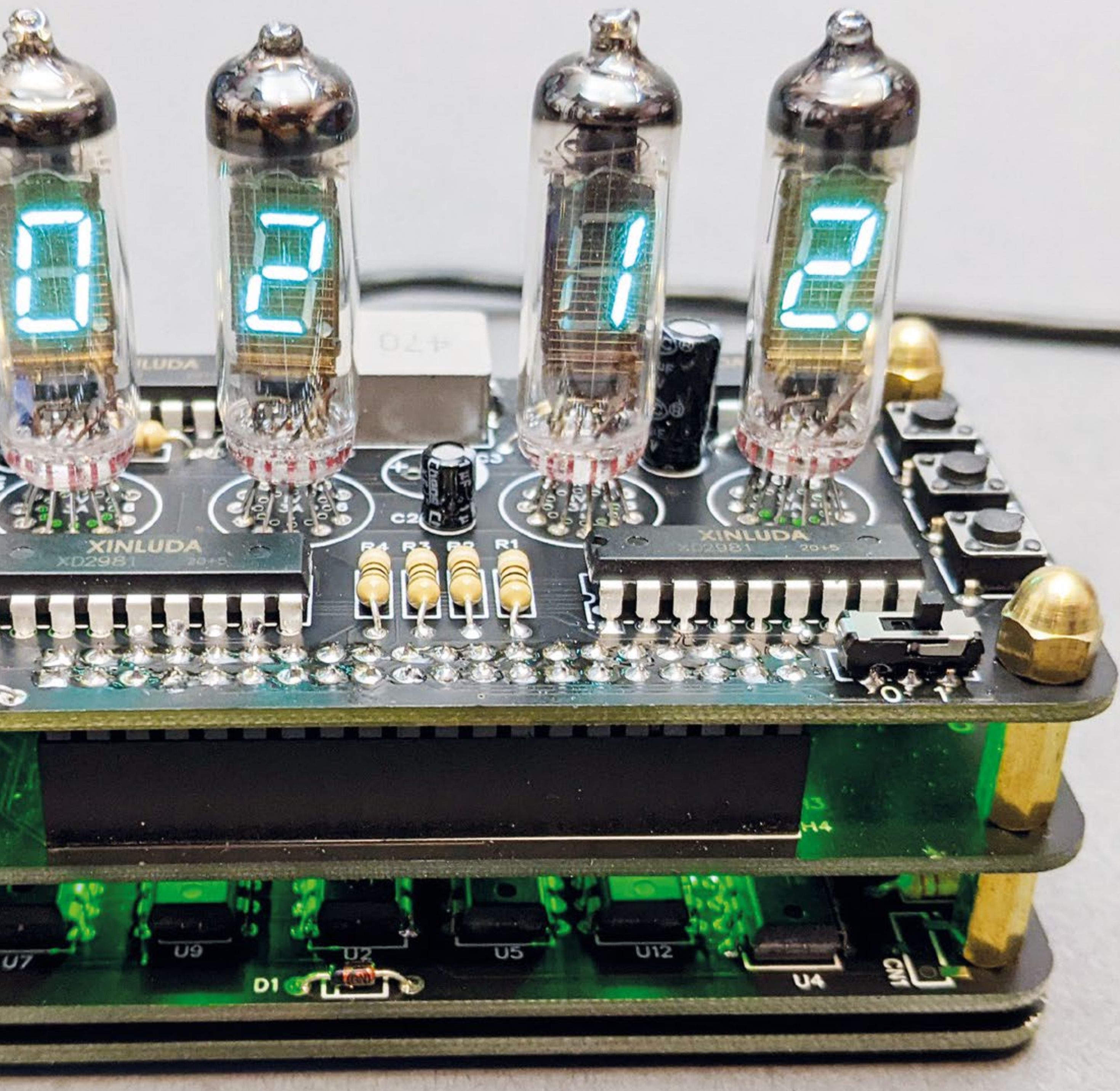
By Bolt Industries

hsmag.cc/VFDClockKit

If, like us, you have a box of Soviet-era VFD tubes that you bought on a whim from a man in Zaporizhzhia Oblast, you could spend ages deciphering the pinouts, assemble the right power supply, and work out the logic of making data appear on the gorgeous glowing seven-segment displays. Or, you could buy a kit such as this one by Bolt Industries. It's sold as a kit, and looks to us like it's not for beginners. Which is fine, because we're not.

The display technology may be of the Cold War era, but you get a choice of control options: the kit comes in a TTL (Transistor–transistor logic) edition that uses only 7400 and 4000 series integrated circuits, or a Raspberry Pi Pico-based version that runs on software. That means if you've ever doodled around with MicroPython on a Pico, there's a good chance you'll be able to get one of these clocks running exactly how you want it. □

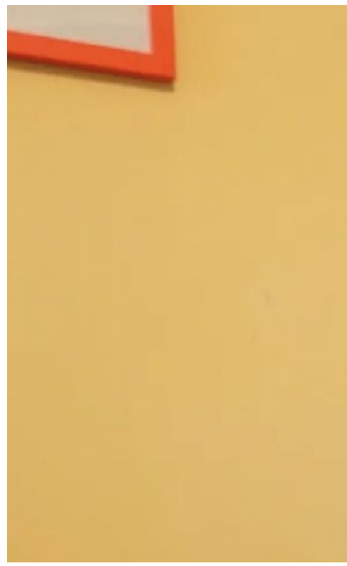
Right Modern brains in an old-school display. Marvellous stuff!


XINLUDA
XD2981
XINLUDA
XD2981
R4 R3 R2 R1
C2
470
U7
U9
U2
U5
U12
U4
D1

Bulbasaur Mini Mod

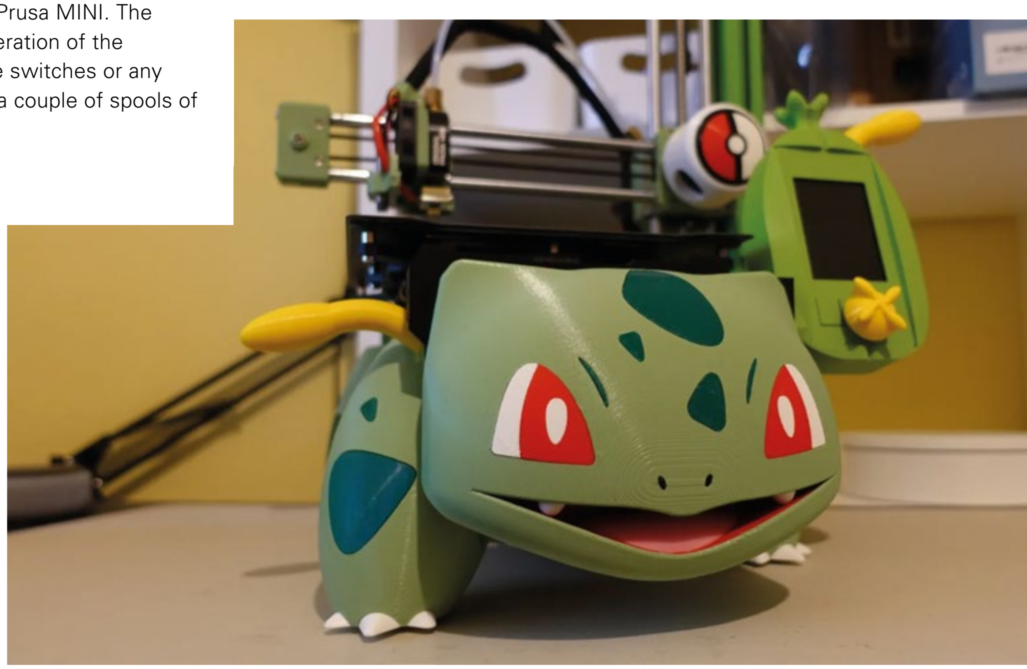
By **Neil Hengist**

hsmag.cc/BulbasaurMiniMod

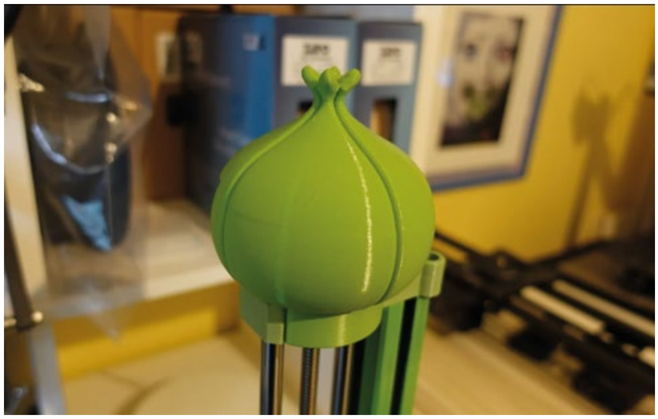


There are many benefits to using open-source hardware: one is the knowledge that even if the company you bought it from ceases production, there will always (as long as someone is willing) be repairs available. You can take your open-source hardware to bits and learn how it works without voiding the warranty, you'll never have interoperability issues, you know you're buying quality rather than a branding 'experience'… and, if you want, you can always modify your device to look like a Pokémon character.

That's what Neil Hengist has done with his Prusa MINI. The modifications are entirely aesthetic, so the operation of the printer is unchanged. There's no need to move switches or any electronics, but Neil does recommend having a couple of spools of Pistachio Green PETG on hand.



Right Open-source hardware is wonderful!




ZIRO
3D Filament
ZIRO
3D Filament

Servo Word Clock

By Moritz v. Sivers

hsmag.cc/ServoWordClock

This word clock is different to others. Where the traditional approach is to light up the words in order to spell out a phrase that tells the time, this version has all the letters lit at the same time – but not in focus. Each letter is lit from behind by an RGB LED mounted onto a linear actuator. To display the letter, the actuator brings that letter into focus; to stop displaying it, it moves the LED so that it's not in focus.

So it's simple in theory, but in practice, it uses 114 servos, 320 3D-printed parts, a custom PCB for the main electronics, a laser-cut acrylic case, and a whole lot of engineering precision.

Right
We now want to see a version of this word clock in Welsh

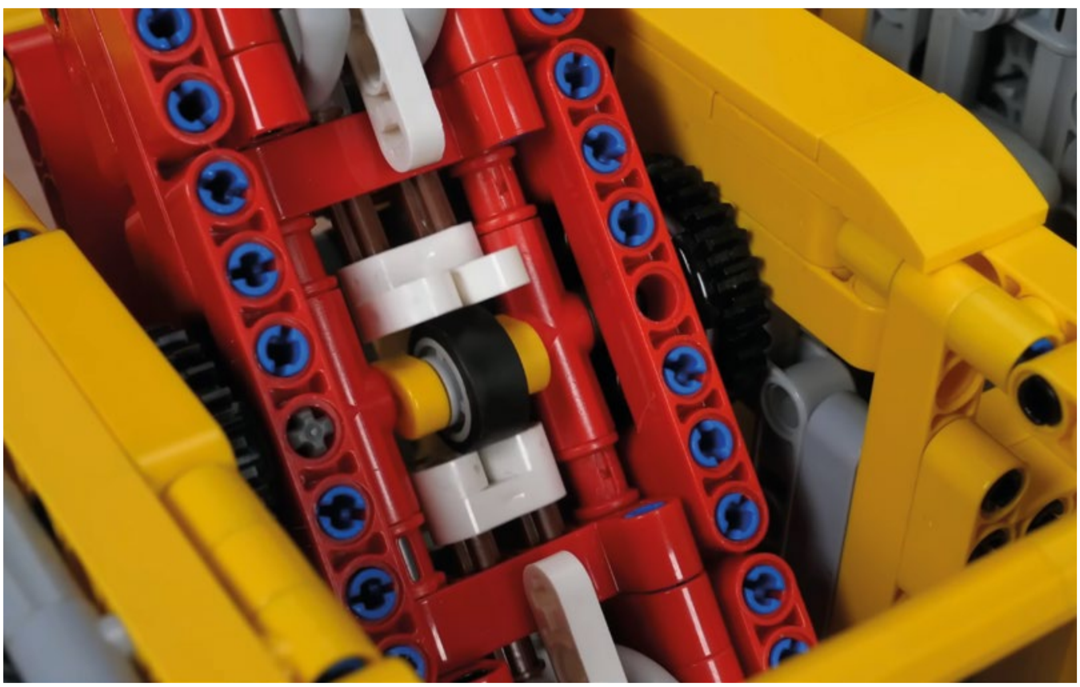
LEGO pneumatic engine

By **Nico71**

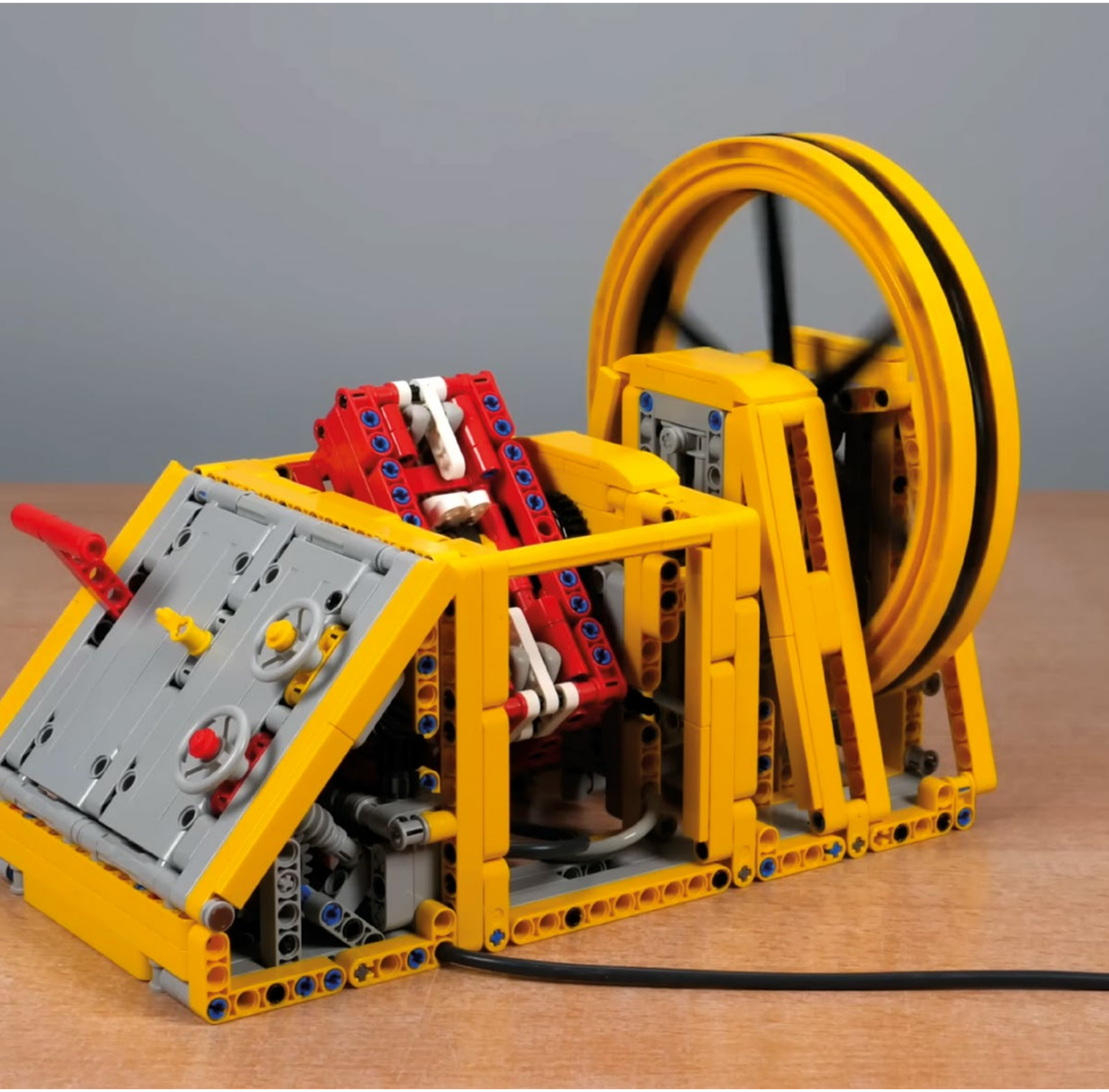
hsmag.cc/PneumaticEngine

As countless dads around the world will tell you on Christmas morning, LEGO Technic™ is not just for kids. This device is a seriously grown-up pneumatic engine, which converts a two-bar air supply into rotation motion via a single pneumatic cylinder and some ingenious switching techniques. Rather than spark plugs or valves, the maker has chosen to control the flow of air within the engine by using adjustable cams that press on flexible rubber pipes, closing and opening the air supply to affect the engine's air supply and timing.

The video by Nico71 is a mesmerising watch, and he's made full build instructions available for a very reasonable €10. □



Right If you feel like building your own pneumatic engine, you'll need your own compressed air supply



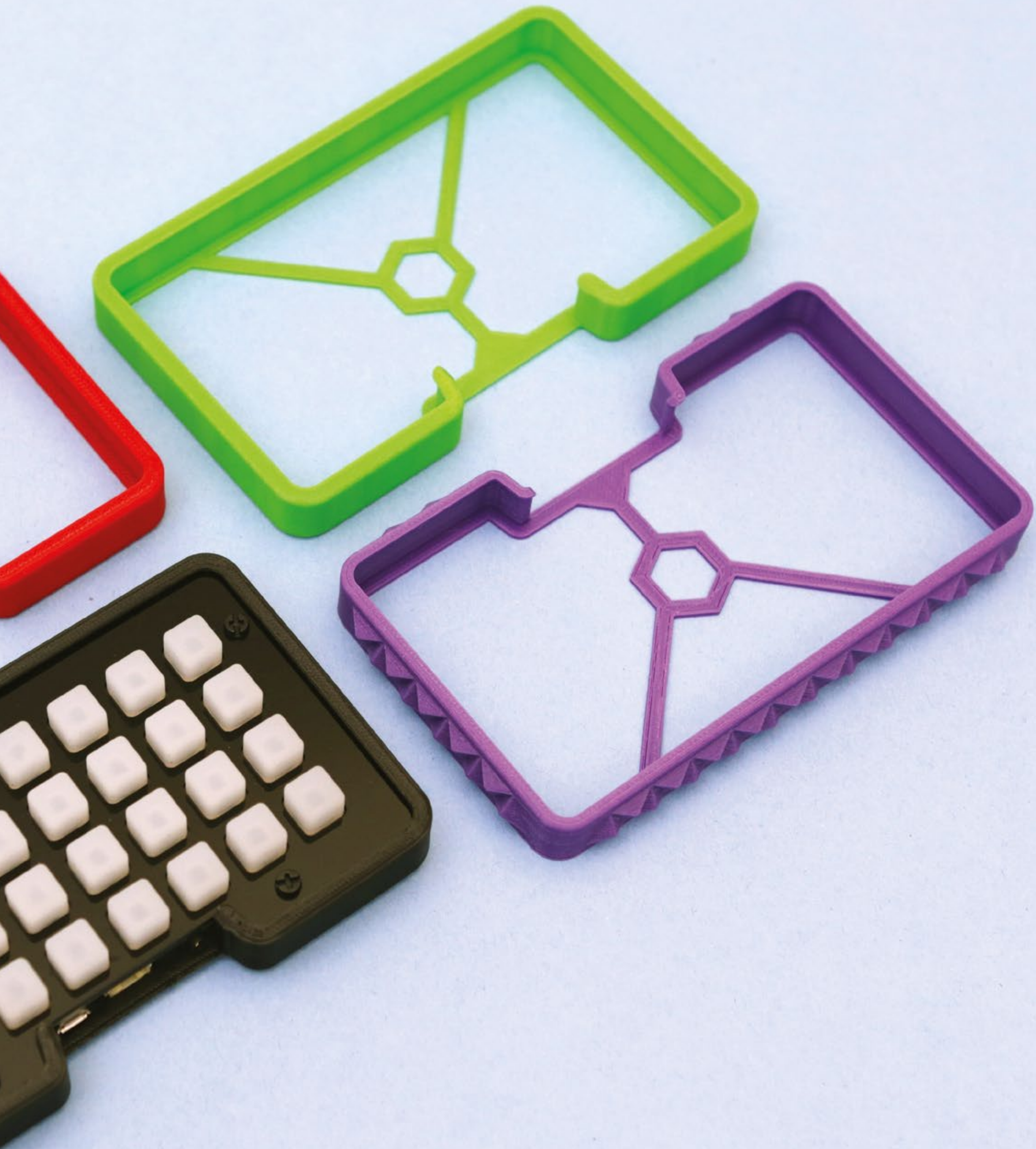
REGULAR

Objet 3d'art

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

If you've ever dropped and smashed a phone that had a train ticket on it, forcing you to pay for a new ticket, as well as the price of a phone repair, you'll be well aware of how broken tech isn't as good as working tech. For a standard-sized bit of kit, you can mitigate your risk by buying a case; if you're working with a more unusual shape, the best thing to do is print your own protection, as the Ruiz brothers have done with this 3D-printed number for the Adafruit NeoTrellis M4.

It's designed for flexible filaments, and adds a bit of protection and a grip to the NeoTrellis's acrylic case. Like it? Get the design files here **hsmag.cc/FlexibleFilamentBumper**, and see if you can work out how to modify proportions to fit your fragile technology.



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**

IT'S ALIVE!

Coding: it genuinely is a superpower. I'm working on a very small project that lights up a blue LED if it's too cold, a red LED if it's too hot, and a white LED if it's about right. As an object, it's useless, because everyone in the house has a different idea of what constitutes 'too cold' and 'too hot'. All it does is give me heart palpitations as I visualise the money I'm spending on heating. Despite that, it works, and I made it, and it gives me great pleasure when I think of the light bulb moment I had when I stopped getting the programming wrong and started to get it right.

Robert
Basingstoke

Ben says: I agree in the strongest possible terms. When you unhook the Raspberry Pi Pico, or Arduino, or Adafruit board from the computer and set it free into the world with only your code to tell it what to do, it feels like you've created life. Controlling devices from a code editor is all very well, but the real magic comes when you use the input from one pin to trigger something happening on another pin. For example, when a temperature sensor reports a value in a certain range, which then triggers an animation of (for example) money going up in smoke. It's a brilliant feeling, and one that we want to share with everyone.

3D DESIGN

Your interviewee last issue was spot on with his diagnosis of 3D printing: the physical product is amazing, but the software leaves a bit to be desired. Downloading someone else's design and printing it is great when you want to use, say, a Bosch battery in a Makita drill, but it's so hard for new users to design anything for themselves. It's not a problem that's unique to 3D printing – every new technology is the preserve of an elite few at first – but it is a problem that is getting better.

Barry
Dublin

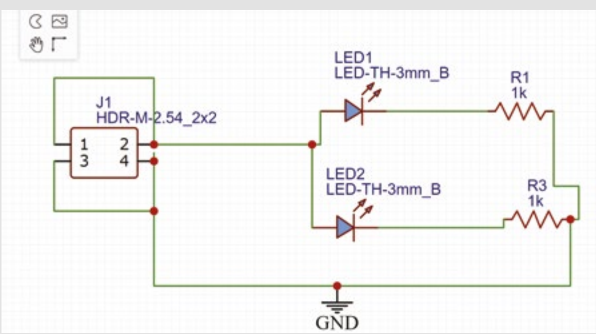
Ben says: I remember using Photoshop 4, painstakingly repairing images scanned from scratched bits of film, and being amazed when my company upgraded to Photoshop 7 with its magic healing tool. That was a few years ago; nowadays a similar technology is available on every smartphone, and it's incredibly intuitive to use. That's the sort of breakthrough that's needed in 3D design, and it will happen. Until then, I'm incredibly thankful to everyone who puts designs on to Thingiverse, Printables.com et al., so that I can make use of their good work.

CHRISTMAS PCBs

Mince pies, like sausage rolls, are always best when they're homemade. I'd never thought of homemade PCBs in quite the same way, but I suppose, having seen your tutorial last issue, that if I want something unique for the tree this year, I'd better get a move on and make my own tree decoration. And, if I don't manage to get one put together in time, I'll have all the skills I need to make something decent in time for Burns Night.

Graham
Oxford

Ben says: I never realised how much I wanted to see an animated PCB haggis until right now. Do it. And to everyone else reading this, the skills you learn using EasyEDA to create a unique piece of hardware will transfer to any national or religious holiday, not just Christmas. Let a thousand flowers bloom.



Meet The Maker: Joseph Herscher

Doing things the hard way

Depending on which side of the Atlantic Ocean you live, you're probably familiar with the work of either Rube Goldberg or Heath Robinson. These two cartoonists drew elaborate machines for doing everyday tasks. Although these artists probably never thought that their machines would be – or could be – built, there are people who have dedicated countless hours to building machines inspired by them.

Joseph Herscher has spent the past 14 years of his life building and videoing some of the most elaborate machines around. The results are on YouTube (**hsmag.cc/JosephsMachinesYT**). We caught up with Joseph to find out what he's been up to, and how to make our own elaborate machines.

Below
Save on heating in this cold weather

HackSpace: Your first YouTube video is of a machine that squashes a Creme Egg. How did that come about?

Joseph Herscher: Yeah, I made that. Gosh. It was 2008. So, 14 years ago. And I was just inspired by these Japanese kids showing contraptions I found on YouTube in the early days of YouTube. I thought they were really cool, so I had to make my own.

I'd always been making contraptions and machines growing up – it was like a little hobby of mine as a kid, but I'd never really done it as an adult. But I felt inspired, and I just started building this elaborate contraption around my thing and it took over my life. Every day after work, I would come home and build on it for about seven months. I, initially, just wanted to do it to entertain my friends, but that wasn't really realistic because it never worked – it didn't work the way I thought it would. And the tea would be cold by the time I finished it – originally it made a cup of tea – then I just changed the ending. I filmed it 300 times so I got one perfect run-through, and then I posted it online and I got a million views! So I started doing more of it, basically.

HS: What's your process for making a machine?

JH: Usually, I start with the ending. So I have a concept, I know the machine is going to do this task, like water a plant or put my dessert on my plate. And then I have to think about what the story is around that. So, putting the dessert in what would be the start of the machine, what's going to trigger the whole thing to begin and, in that context, it would make sense that maybe having a sip of the drink would start it, or putting the plate down, or something like that. And then I start filling in the bits in between based on, I don't know, the theme of the video, or specific objects that I'm interested in that I →


GUNTRIP
PERSONALITY STRUCTURE and HUMAN INTERACTION
PROVIDENCE ISLAND
Calder Willingham
The Uncivil Wars
Ireland Today
O'Malley
GAS
OFF
MSR
DANGER

Above Why use matches when you can use detcord?

found that do cool things, because it's all made out of everyday objects, and I like to find unexpected ways of using them. And then, I will start building and filling in all of the gaps to create one continuous chain reaction.

HS: You generally use everyday objects in your builds. How do you find the things to do the right bits?

JH: Originally, I would spend a lot of time looking for things – now I buy everything online, because I'm so specific in what I need, that I kind of have to order a specific size/dimension of the object. It would be much harder to do what I do without an online shop. And it'll just take longer to hunt around and find. In New Zealand we don't have very good online shopping. A lot of my time [when I lived there before moving to London] was just like walking around shops and playing with objects in shops and trying to see which ones work best, or what I could roll a ball across, tinkering, kind of. And sometimes I was ordering like ten different [things]. Like I wanted to use a spatula, and I would have ten different spatulas. And then [I'd] just try out all the different spatulas to see which had the best shape for the task.

Oftentimes, it's like I get an idea just from accidentally dropping something and it bounces in a funny way, or it flips in a way I didn't expect, and then I'll make a record of it and use it later on in a machine.

Right Looking for a job? Joseph is recruiting. Get in touch via josephsmachines.com

HS: You said that the Creme Egg Squisher took 300 takes to film all the way through. How many times do you usually have to film your machines to get the one perfect shot?
JH: It takes between 50 and 100 takes to get a [shot]. Yeah, so [for a shoot] often I have to eat a lot of food. I try to put the messy steps at the end so it's not so laborious to reset it each time for the food. The food elements that I have to consume a lot [of], I will put them at the end, so I'm not eating like a whole roast chicken in every single shot.

> **The biggest challenge is making it translate to an** audience so that they get it

HS: What are the big challenges in creating your machines?
JH: The biggest challenge is making it translate to an audience so that they get it, because you could make a really clever, cool mechanism, but people don't follow it and don't understand it. And that was the point – I'm here to entertain you, not to just make actual machines, and I'm not selling these machines to consumers. They're just there to make people smile and laugh and think differently about the world around them.

Sometimes you have ideas, and you think that they're going to be really clear and easy to follow, and then they're not. So getting things to slow down, not be too fast, and to be clearly visible, and fitting things in the frame so that they're not too small or too big.

The pacing is so important. You can't use strings, because if you have too many strings, it's really hard to follow because they move too quickly. All kinds of things like that. Or, sometimes, I'll have concepts I try that just don't work. Playing around with viewers' expectations and surprising them, and you think that [you're] going to surprise them, but they're just looking at the other part of the screen, and they don't even see the thing that you want them to look at. So you spent all this time building it and it didn't work. And for the viewer, you often don't know those things until you've posted it. That can be quite challenging – trying to predict what the psychology of the viewer [is].

HS: Are there any bits of advice you'd give to people trying to build similar machines for themselves?
JH: Yeah, play and experimentation. Don't be afraid of failure. Take some familiar everyday objects in your life, and just play with them and see what they do that makes you smile or laugh. And once you've found that, then you can start thinking, well, what could that connect to? How could I connect it to something else?

And then spend lots and lots and lots and lots of time testing it, because you think something works a few times? Oh, that's great. That'll always happen. It never does what you want it to do on the day you're filming. So, I spend the whole week just testing a machine before I film.

Below Machines can be built out of any components that you can find

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

20 Amazing RASPBERRY PI PROJECTS

Armed with a Raspberry Pi and a few electronic components, a world of adventure awaits the intrepid maker

While a Raspberry Pi is a capable computer, and can even be used as a desktop PC replacement, its GPIO (general-purpose input/output) header is what makes it different and truly powerful. This enables you to hook up electronic circuits and components such as LEDs, sensors, motors, and servos.

To help inspire you to get creative when connecting electronics to your Raspberry Pi, we've compiled a list of 20 of the most impressive and ingenious projects around. Dazzling light-up devices include an interactive model Stargate, LED cube, London Underground map, and spinning POV (persistence of vision) display.

Moving into robotics with motors and servos, you could make a piano-playing, metal-detecting, chess-playing, or air hockey-playing robot. Or maybe a LEGO submarine or even a ceiling planetarium. If you fancy something musical, how about a synth guitar or glockenspiel player? Or get arty with a CNC drawing machine or even a social media jacket. The only limit with a Raspberry Pi project is your own imagination.

Phil King

A long-time Raspberry Pi user and tinkerer, Phil is a freelance writer and editor with a focus on technology

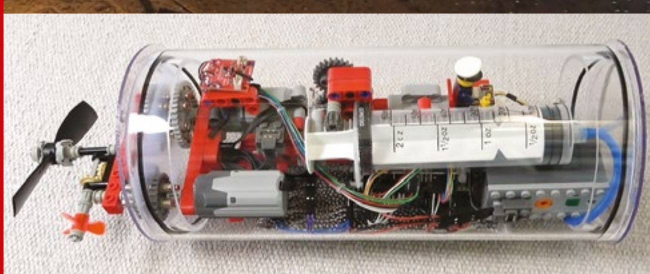
LEGO Submarine 4.0

The Finnish maker behind the Brick Experiment Channel on YouTube loves using LEGO to create working models that address complex physical and mathematical challenges. One of his most impressive creations yet is this working radio-controlled LEGO submarine (his fourth attempt).

With the processing power of a Raspberry Pi Zero 2 W on board, it uses a servo motor and LEGO rack and pinion system to operate a syringe as a piston ballast; this draws in or expels water to raise and lower the sub's buoyancy level to make it rise or fall in the water. Tungsten pellets are used for ballast. An absolute pressure sensor obtains accurate depth readings, while a micro laser sensor measures the distance to the 'sea bed'. Positioning is aided by a PID (proportional integral derivative) controller loop in the Python software.

Naturally, protecting the electronics from water damage was a key consideration in the build. To this end, a fair amount of time and effort went into creating the submarine's transparent acrylic case with tight-fitting end caps with rubber seals.

hsmag.cc/LegoSub



GIANT Battleships

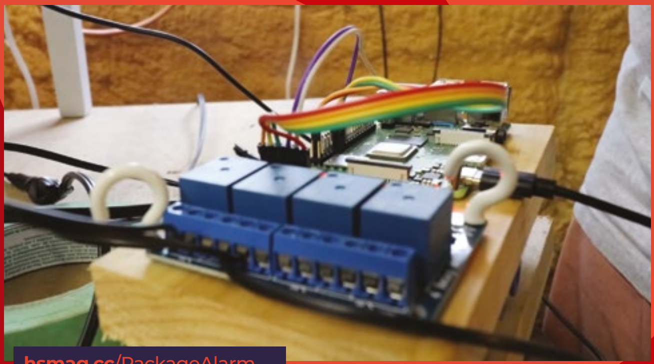
Maker Dan Aldred has taken the concept of the classic Battleships game and blown it right up. After previously recreating the board game on Raspberry Pi with a Sense HAT, he opted to go bigger by using a giant 10×10 LED matrix that he'd already created for light shows and discos. Cleverly, the NeoPixel LEDs in the grid have their light diffused by glass jars spray-painted white.

An innovative method was also employed for the user to choose the coordinates to target torpedoes at the hidden ships in this solo version of the game: a 1960s Bakelite telephone fitted with a Raspberry Pi. As you turn its dial to select digits, a circuit with a GPIO pin is interrupted by the relevant number of clicks. Each coordinate is sent in turn (using Python sockets) to the LED board's Raspberry Pi.

The project even makes use of the telephone's handset speaker to give voice instructions to the user and play sound effects such as explosions. →

hsmag.cc/Battleships

PACKAGE *Thief* DETERRENT



hsmag.cc/PackageAlarm

Like many people, Canadian maker Ryder (of the Ryder Calm Down YouTube channel) has experienced the problem of packages delivered to his home being stolen by porch pirates while he was out. Armed with a Raspberry Pi 4, a security camera, and a few deterrent devices, he set about solving the problem.

First of all, images from the camera connected to his Raspberry Pi are processed by a custom machine learning model (trained using Google Cloud AutoML) to detect if there is or isn't a package. If one has been taken unexpectedly, the Raspberry Pi sends signals to a relay switch board to activate a variety of alarms to entice the thief to drop the package. Those surprises for thieves include a sprinkler, a loud truck horn, and a flour shower.

Naturally, since he didn't want to trigger the alarm himself, he has trained the system to deactivate when it recognises him in the frame. One slight issue is that the AI sometimes thinks his cats are packages!

ROBOT ARM *Clock*

Even a stopped clock is right twice a day, but maker Hendrik Ohrens wasn't content with that when his timepiece broke. Instead, he opted to build a robotic arm to physically move the clock hands to the correct time every minute!

Having explored inverse kinematics and computer vision, Hendrik came up with a far simpler solution. Using Python code running on a Raspberry Pi 3B+, he trained the robot arm manually for each tiny movement of the minute hand required for a complete rotation.

Once trained, the Raspberry Pi relays the precise positional instructions to a connected Arduino MKR board equipped with a shield to control the arm's four servos.

Instructions, code, and files for the arm's 3D-printed parts can be found in his GitHub repo: **hsmag.cc/RobotArmGit**.

hsmag.cc/RobotArmClock

LONDON *Underground* MAP DASHBOARD

hsmag.cc/TubeDashboard

While many makers create Raspberry Pi projects for use in the home or just for fun, Richard Kirby built this one to help with his work. As a test manager for a company working on automating parts of the London Underground (aka Tube), he needed a way to monitor the real-time status of the train lines to instantly alert him to any problems.

Made from 5 mm plywood and printed network map, this large dashboard is ideal, featuring 284 individually drilled holes to house multicolour NeoPixels. A Raspberry Pi Zero 2 W gathers data from the Transport for London (TfL) Open Data site to get the statuses of the various lines, which it then converts into animations: 'Good Service' is represented by fully lit LEDs for that line; 'Minor Delays' is rapid flashing from 50% to 100% brightness; while 'Severe Delays' has slower flashing.

The map is also connected to a Bluetooth speaker for service announcements, with text-to-speech used to turn the data into spoken words.

DeMOOR *Orrery*

After seeing a 240-year-old orrery – a mechanical model of the solar system – on the ceiling at the Eise Eisinga Planetarium, Chris DeMoor was inspired to build one of his own. Rather than copying the pendulum clock-driven system of the original, however, he opted to use six Raspberry Pi Zero boards.

Each controlled by a Raspberry Pi Zero, the six inner planets of the solar system move in orbits around his ceiling in real time. Two different methods were employed for their movement. Mars, Jupiter, and Saturn are attached to front-wheel drive, 3D-printed cars which run on tracks on the upper, non-visible side of the project. As their orbits would be too small for tracks, Mercury, Venus, and Earth are mounted to dishes connected to rotating stepper motors.

To obtain its correct position, each planet's Raspberry Pi connects wirelessly to a web server which plots their orbits using the jsOrrery JavaScript library. The orrery planets are then moved accordingly. →

demoor-orrery.com

PIANO-PLAYING Robot

Regretful that the piano in his living room was played all too rarely, Étienne Allaire came up with the idea of a robot that could play music. To do so, he mounted a wooden frame with 15 solenoid switches to an electric keyboard. Controlled by a GPIO pin on a Raspberry Pi 3B, connected via a 16-channel relay module, each solenoid pushes down on a key to play a note.

The system can read a MIDI file, convert it into on and off signals for each note, and play the tune, although there is a limitation as to how fast it can play due to the speed of the solenoid actions. A basic UI allows the user to choose between scales, arpeggios, or melodies.

The most impressive feature, however, is the robot's ability to read previously unseen sheet music – using a Raspberry Pi Camera Module – and play it. The open-source tool Audiveris uses optical character recognition (OCR) to read the notes on the sheet music.

hsmag.cc/PianoRobot

STARGATE Model

As a fan of the *Stargate-SG1* series, Kristian Tysse was inspired to create his very own Raspberry Pi-based scale model of a Stargate, complete with animated spinning wormhole. To achieve the latter effect, he created an infinity mirror with the light from 122 LEDs reflected back and forth. The model itself was painstakingly created in great detail, built from 3D-printed parts.

Most importantly, it includes a DHD (Dial Home Device) to dial any Milky Way address featured in the show. Acting as a USB keyboard, its key presses are sent to a Python program running on Raspberry Pi, causing the Stargate's symbol ring to rotate accordingly and each of the seven chevrons to light up and move inward – done using stepper motors.

If a correct set of symbols is dialled, the wormhole is established and audio clips from the TV show are also played. The Stargate can even dial other Stargates over the internet, and receive incoming wormholes from them. Find instructions for making one on the website.

thestargateproject.com

AIR Hockey ROBOT

When students Ondřej Sláma and Dominik Jašek needed to write their course thesis, they chose the Air Hockey Robot they had built from scratch. Taking about a year, the project involved creating robot movement control algorithms, computer vision, game strategy algorithms, and a user interface.

The game table itself was designed in Fusion 360 and built from plywood, with a mesh of 920 holes creating an air cushion. The processing power for the robot's optical puck recognition and AI strategy is provided by a Raspberry Pi 4 connected to a Camera Module mounted in the overhead part of the frame.

For the mechanical aspect, the pair opted for an 'H-bot' design to move the robot's paddle. Held in a 3D-printed housing, the paddle is moved from side to side using a pulley and belt system with two stepper motors. You can find the code used for various parts of project in their GitHub repos (linked from the YouTube video).

hsmag.cc/AirHockeyRobot

CNC Drawing MACHINE

Made by Yannik Dolde, this seriously impressive Raspberry Pi project repurposes old CD drives as motors for the X and Y axes of a CNC machine that can draw a picture. The pen is mounted in a 3D-printed holder that moves up and down using a servo with a spring-loaded metal rod to regulate the pressure on the paper. A Raspberry Pi 3B+ with custom Python software runs a user interface and reads a standard G-code file, line by line, to send commands to an Arduino Mega to move the motors.

hsmag.cc/CNCDrawing

LED Cube

An LED cube is a popular maker project, using LED matrix panels for its sides. Inspired by other cube creators, Sebastian Staacks originally envisaged crafting a stationary mood light for his living room. However, his wife's disapproval led to him turning it into something more useful: a parametric animation reflecting the status of his PC's Ryzen 5 CPU.

As it's only intended to be viewed from one angle, his 'cube' comprises three 64×64 RGB LED matrix panels. These are held by a 3D-printed frame which also contains a Raspberry Pi 2 with an Adafruit LED Matrix Bonnet. It cost Sebastian less than €150 to make.

Making use of an OpenGL shader, the cube displays impressive glowing effects in real time based on the CPU's temperature and its core loads. →

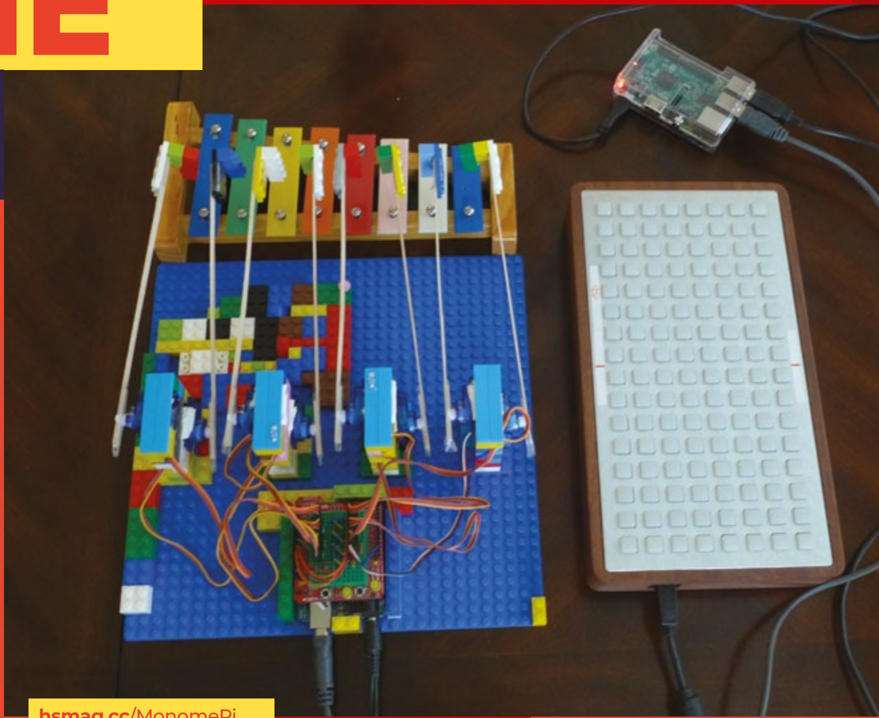
hsmag.cc/LEDCube

MONOME Pi

Joon Guillen's unusual music box features old and new technology working in perfect harmony. For input, a Monome Grid controller is connected to a Raspberry Pi 3 running a step sequencer program which registers the user's button presses on the Monome, lights them up, and sends serial commands to an Arduino Uno.

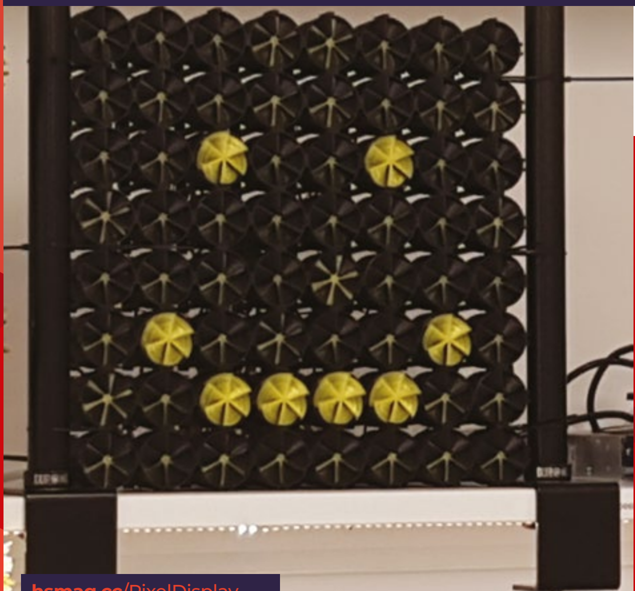
The Arduino is connected via a ProtoShield Kit to eight servo motors, which move hammers to play the correct glockenspiel notes to match the pattern shown on the Monome. The makeshift hammers are made from coffee sticks, sticky tape, and LEGO blocks borrowed from Joon's daughter – which shows that you don't need to be a master craftsperson to build an interesting project.

Joon says that you could substitute the Arduino with a Raspberry Pi servo/motor driver board, and the Monome with a touchscreen or web UI.



hsmag.cc/MonomePi

PIXEL Electromechanical DISPLAY



hsmag.cc/PixelDisplay

When considering how to make a decorative display for an office party, Gavan Fantom thought a standard LED matrix would be too obvious and simple. Instead, he designed and built Pixel, an ingenious electromechanical matrix display controlled by a Raspberry Pi and comprising no fewer than 448 3D-printed parts. Each of the display's 64 'pixels' is turned on and off by a servo rotating a 3D propeller-like shape to reveal its bright yellow blades from a black casing.

While it was inspired by traditional flip-disc electromechanical displays, its unique pixels can also be rotated to intermediate positions to achieve greyscale-style shading and ripple effects. Each pixel comprises seven 3D-printed parts, one servo motor, and two nails to transmit the latter's rotation to reveal the vanes from the casing. To drive 64 servos from a single Raspberry Pi, three Pololu Mini Maestro 24 boards are used.

The end result is an amazing display that just has to be seen in action – check the YouTube videos in the linked blog piece.

PUITAR

Finding his brother's old electric guitar difficult to play, Behruz Farshi opted to transform it into a digital instrument using a Raspberry Pi Zero. His Puitar (pronounced 'pit-tar') features a keypad matrix of 22 frets and six strings. When the player presses a string onto a fret, it forms an electrical connection that is detected by one of Raspberry Pi's GPIO pins and the appropriate note is played – which could be a piano or any MIDI sound you like. To expand the number of pins, an IO Pi Plus board is used.

To create the electrical matrix, Behruz drilled holes into the fretboard and soldered wires to the frets from below. While his prototype Puitar can't handle polyphony, and accidentally touching strings together can cause it to malfunction, it's a clever concept that could be adapted and possibly modified into a full-on synth axe.

hsmag.cc/Puitar

DIGITAL *Zoetrope*

One of the oldest animation methods, a zoetrope is a spinning drum with a set of static images that, when seen in rapid succession from a slit in the side, appear to be moving.

Fascinated by the concept since he was a child, maker Brian Corteil decided to make his own zoetrope using 15 e-ink displays – Pimoroni Badger 2040s. Brian designed the zoetrope drum using the SolidWorks CAD program, creating outlines to be laser-cut from 3 mm and 5 mm plywood, while some other parts were 3D-printed.

A Raspberry Pi 4 is used to send screen updates to the e-ink displays via USB connections, while a Pico in the base controls the drum rotation via a motor controller and monitors the emergency stop buttons. All the electronics spin around with the drum – to avoid the main central power cable twisting, there's a split ring in the base.

A second Raspberry Pi 4 connected to a flatbed scanner also enables animations created on a cell sheet to be scanned and uploaded to the zoetrope.

hsmag.cc/DigitalZoetrope

DISCOVERER *Metal-Detecting* ROBOT

Building a robot vehicle is one of the most popular Raspberry Pi projects, but this one is rather special and could be useful. Maker Ingmar Stapel came up with the idea after watching a TV show about people trying to find gold with a sophisticated metal detector.

For the chassis, he used a plastic storage box to contain all the electronics, adding PVC piping around the exterior to hold a dual-servo pan-and-tilt camera – which enables the streaming of live video to a web dashboard – along with a Gary's Pulse-AV metal detector. A step-down converter is used to change the detector's 12 V output to 3.3 V for a GPIO pin on the Raspberry Pi.

Discoverer's four motors are driven via L298N H-bridges. As well being remote-controlled with a range of 350 metres, the robot can be programmed to move autonomously from one waypoint to another thanks to the addition of a GPS receiver and a Sense HAT for the compass. →

hsmag.cc/Discoverer

SOCIAL MEDIA *without the* INTERNET

Imagine what a physical, real-world version of interacting on social media might look like. That's just what interactive artist Tuang Thongborisute (Tuang T) wanted to explore, leading her to create this very special jacket as a performance art project.

The blazer features numerous electronic elements – controlled by a Raspberry Pi, aided by an Arduino – which enable people the wearer meets to engage in six social interactions. For instance, a follow is made by tapping a pressure-conductive resistant sheet on the right shoulder, while unfriending involves a push of a button located on the left side, near the heart. Adding a new friend is achieved by shaking hands, connecting two conductive rings on the wearer's fingers.

The jacket also has a 7-inch touchscreen for further interaction and to show status information. In addition, three tiny cameras broadcast the real-time interactions on a local network for performing in a closed environment like an indoor gallery.

hsmag.cc/SocialMediaJacket

RASPBERRY *Turk*

raspberryturk.com

This amazing chess-playing robot was inspired by the 18th-century 'Mechanical Turk'. While the latter had a human player concealed inside to play its moves, the Raspberry Turk has a Raspberry Pi 3 for its brains.

Created by Joey Meyer, this open-source project is built into a small 3×3 ft (91×91 cm) table. A box on one side houses all the electronics, while the robotic arm is mounted on its top. The arm is built with Actobotics components, Dynamixel servos, and some 3D-printed parts.

The arm's motion is controlled by the rotation of two servos attached to gears at the base of each link. Another servo controls the gripper mechanism which uses an electromagnet to pick up pieces. Naturally, great precision is required to move pieces on the board, so Joey built up a dataset of the arm's movements to aid his mathematical model.

To evaluate the positions of the pieces, a high-mounted Raspberry Pi Camera Module captures a view of the board which is then perspective-transformed using OpenCV. The Stockfish chess engine is used for its AI.

Pico Projects

AUTOMATIC Dog Ball LAUNCHER

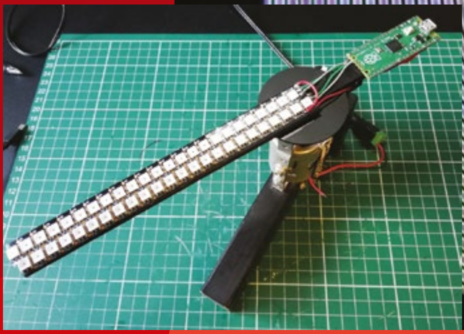
Rather than buy a commercial automatic dog ball launcher, maker Brankly opted to build one of his own, based around a Raspberry Pi Pico microcontroller. After playing around with different shapes, he found that a spherical design looked the best; it was also small enough for his 3D printer to print.

When a ball is placed into the launcher's funnel, it is prevented from falling into the launch channel by a piece of plastic that's controlled using an SG90 servo. The ball is then detected by a sensor, prompting Raspberry Pi Pico to get ready for launch.

To vary the distance of the ball launch each time, to make it more fun for the dog, a motor controller is used to randomise the speed of the motors using PWM (pulse-width modulation). Once a ball has been launched, the motors turn off and the servo blocks the entrance again so that the machine is ready for the next ball.

hsmag.cc/DogBallLauncher

POV Display



'POV' stands for persistence of vision, the optical phenomenon that makes moving pictures possible in cinema and TV, and on which this whizzy Pico project relies to depict an image from spinning LED strips.

Created by Japan-based family team of makers HomeMadeGarbage, it makes use of two of the Pico's PIO (Programmable IO) state machines to control, in parallel, a pair of super-bright 24-LED DotStar strips on its rotating arm. The arm, which also holds the Pico board, is spun at high speed by a motor while the LED strips are blinked in carefully controlled sync with the rotation speed to display a static or animated image.

A wireless charging module is employed to power the spinning arm, using a coil on the underside of the latter and another on top of the motor. □

hsmag.cc/PicoPOVDisplay

NEW INTERNATIONAL OFFERS!

Get 6 issues plus a **FREE Raspberry Pi Pico W** for just...

£35 USA and EU

£40 Rest of the world

Head to **hsmag.cc/subscribe** to get yours today!

FEATURE

HOW I MADE

By **Andrew Lewis**

WINTER IS COMING

Fighting the effects of winter in your workshop with heated chairs and insulation

Winter is coming. It's cold, it's wet, and it can be the most difficult season to find the literal and figurative energy to keep making things. The cost of heating is getting higher for everyone, and a dedicated workshop can quickly turn into a money-eating monster in the winter months. For all of the advantages of having a workshop, there are some pretty big disadvantages, too. A permanent workspace is exactly that. It continues to exist in space even when you're not using it. Some tools, consumables, and machinery can be ruined by cold temperatures, and in winter, a workshop becomes an extra place to heat. In this article, you'll find out how to heat a workshop more effectively, and how to stay warm yourself without relying on power-hungry heaters.

A maker's haven for the summer will become a burden without proper winter planning. It isn't just about feeling the cold, it's about how the cold effects your equipment. Inaction through ignorance or simple failure to prepare for the winter

can lead to expensive failures when the temperature drops below freezing. To get the best out of your workspace, you'll need to address both of these problems. Keeping your equipment in working order, and keeping yourself warm while you're working.

When you're working on larger projects, you're generally moving around and doing things that will help keep you warm. Making sure that you're wearing appropriately warm clothing means that you'll be happy in your work. However, if you're doing fine work, sitting still at your workbench, you'll start to feel the cold very quickly. So how do you solve this problem? You can borrow a solution from the automotive industry and create your own heated seat. Heated seats are popular in modern cars, and it's not uncommon for people to want to retrofit them into older vehicles. There are plenty of kits available online that feature heating pads and controllers designed to fit into car seats. With a little bit of fiddling, it's possible to add one of these kits to most office chairs, and you can power it with an electric tool battery that will last between two and five hours (depending on how warm you want to be). →

Above ↑
Heating the person, not the space is a common way to increase comfort levels cost-effectively. While it won't solve all equipment problems, strategically placed low-power heaters like the one installed in this chair can make a big difference

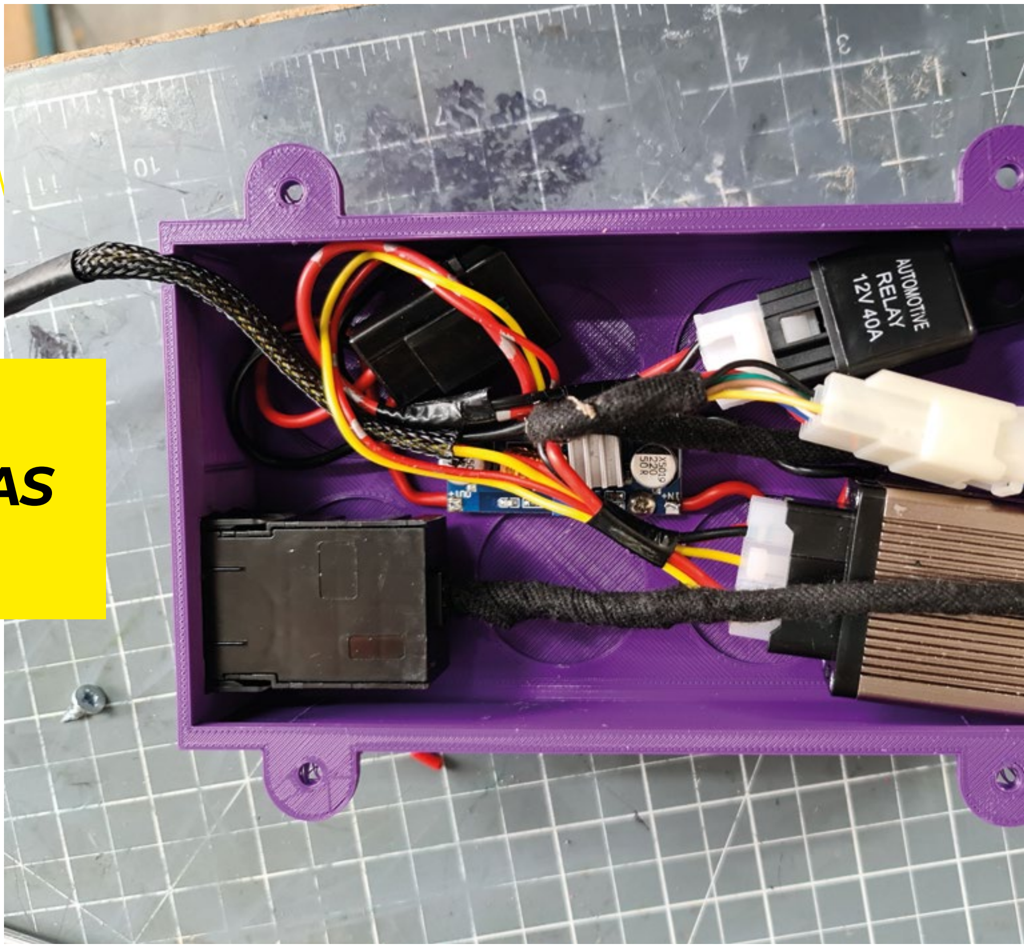
"HEATED SEATS ARE POPULAR IN MODERN CARS"

STAYING SAFE

When powering the heating elements from an 18 V power tool battery, you will need to use a DC power converter to reduce the voltage down to 12 V. A minimum of a 5 A DC converter is recommended, given the power requirements of the heating elements. The first item in your circuit after the battery should be an appropriate fuse. In this case, a 5 A fuse should be plenty. Adding a thermal fuse to each heating element is advised. A 76°C 3 A fuse like the SETFuse X0 can be used to cut power if the heating element ever malfunctions and overheats. Position the fuse in-line on the positive wire, and fix it somewhere on the heating. You can keep it in place with Kapton tape. If the temperature of the element gets above 76°C, the fuse will blow, preventing a possible accident.

MA

“REMOVE AS MANY STAPLES AS YOU NEED”



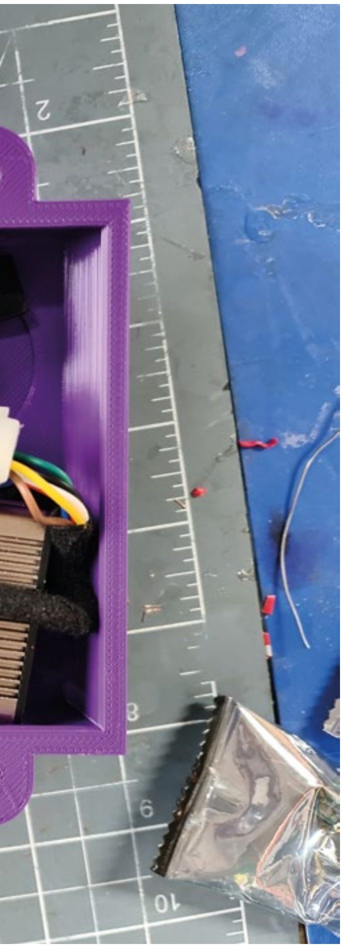
Right →
You can see all of the components that come with the seat heater here, along with a DC power converter that regulates the power from a tool battery to 12V

WATER WOES

A hidden danger of a cold workshop is condensation. If the workshop and its contents get cold enough for water to condense on them, chaos will ensue when you try to use them. Firstly, water can condense, drip, pool, and rust your machines. Secondly, water inside electronic equipment or on sensitive items like laser cutter mirrors will lead to disaster. One way to deal with this problem is to use chemical moisture traps to capture water from the air. Another way is to employ an electronic dehumidifier. The best approach is to keep the temperature of the workspace stable at a temperature that's high enough and dry enough to prevent moisture from condensing. A heater with a frost switch can help with this, turning on to keep the temperature somewhere above freezing. It's also possible to include a small heat source (such as a very low wattage incandescent bulb) to effectively heat the inside of some equipment and prevent condensation from forming. This can be a good solution for equipment with relatively large enclosures, like a laser cutter or an enclosed 3D printer, where you can use the machine's heated bed to keep the internal chamber warm. Heating an enclosure like this will use power and cost some money, but it's cheaper than having to replace components that have been damaged by condensation. If you do find that your equipment has gotten too cold, you'll have to gradually warm it and make sure that the water inside has evaporated before you start using it.

START AT THE BOTTOM

A typical seat heater kit for a car will have four heating pads, a relay, a dashboard switch, and a small circuit board or sealed box that lets you control how hot the seat will get. Often, these kits don't come with any instructions, and you might be a bit confused by the number of power wires that are sticking out of the control box. Normally, you will have two thicker red and black wires, which are the main 12V power input for the heater. You'll also have another thin wire that controls a main power relay, and possibly a thin wire that will make the switch light up on the dashboard when the headlights are on. Because you're probably not going to be driving your workspace around at night, you can safely connect these wires to the positive 12V line and forget about them. If you look at the switch for the heaters, you'll notice that there are two dials. One dial is for the passenger seat, while the other is for the driver's seat. As you'll only be using one seat, you can use these dials to



HEATING THE ROOM

A well-insulated workshop (or any other building) can remain remarkably warm without any heating. However, it's difficult to achieve such high levels of insulation unless the building has been built specifically with eco-living in mind from the beginning. For most people, some level of heating is going to be necessary. You can discount any idea of portable gas or paraffin heating, as they produce high levels of very moist air and fumes that make them practically useless or downright dangerous to use in an enclosed workspace. Conventional electrical heaters are potentially useful, although convector heaters typically use several kilowatts of electricity every hour, making them a very expensive option.

control the temperature on the back and the seat of your chair separately. Simply connect one pad to the passenger connector, and the other pad to the driver's side. You'll need to cut the wires from the control box, since it's designed to be fitted in a car a metre or more away from the seat, and you'll be fitting the box underneath your chair about 15 cm away from the heater panels.

Fitting the heating elements inside the office chair is the fiddly part of this particular project. If you're lucky, your chair's seat cushions will be removable and have zips or Velcro strips that you can undo to get inside the padded parts of the chair. Unfortunately, the vast majority of office chairs are not designed this way. The fabric for the seat will more likely be stapled onto laminated board, and you'll need to disassemble the chair to get access to the inside of the seat cover. Often, you'll need to remove the arms and legs of the chair, and then disconnect the seat from the back-rest if it is removable. You should be able to see the staples once the chair is disassembled.

Remove as many staples as you need to get inside the seat cover, and then slide the heating element in between the outer cover of the seat and the interior foam. This seems like a simple instruction, but in reality, you will probably spend 20 minutes cursing this article, the person who wrote it, and all of the ancestors of the person who designed the chair while you are trying to slide the →

Below ↓
Removing staples is much easier with a decent pair of pincers. They offer easy leverage and are less likely to accidentally grip and rip your chair's fabric than ordinary pliers

CHILLY CHALLENGES

It isn't just machines and people that suffer the effects of the cold weather. Greases and oils behave differently in cold weather, becoming more viscous and potentially becoming unusable. Some industrial paints can be ruined if they get too cold, and most paint won't brush or dry properly if it's below 10°C. Spray paints won't spray properly in cold weather, and butane gas canisters will stop vaporising gas properly at roughly the same temperature. The pressure in acetylene tanks will drop as the temperature decreases, too – so don't be too surprised if that full tank reads almost empty now that it's colder. Battery power drops off dramatically in cold weather, meaning that you'll need to change and charge batteries more often on power tools, and some items like multimeters that use disposable batteries might start to give battery warnings or read incorrectly. If you're going to need temperature-sensitive consumables, it's probably best to bring them into the warm for a few hours before you use them.

Right →
Pulling plugs and wires through fabric and foam can be tricky if you've never done any upholstery work. Push pliers through from the outside of the foam first to open up the hole, then grip the plug and pull it through to the outside. Don't use too much force, as you're likely to bend or break the plug

Below ↓
Screw the project box in place under the chair so that you can access the switch and the battery easily. The base of most office chairs has surprisingly thick wood, but take care not to use over-long screws

stubborn heating element flat against particularly grippy upholstery foam. Worse, you need to hold the heating element in place with double-sided tape, removing the backing while it's in position inside the chair, and then finally restaple the outer cover back into place while remembering to keep the wires for the heating element hanging outside the fabric.

SIMPLE BUT WARMING

All that remains is to connect your control box to the heating elements, and stash all of the electronics in a box under the seat. Make sure that you position the box so that

QUICK TIP

You can fix one of the remaining seat heating elements to the underside of a self-healing cutting mat, and use it to warm the mat up. This will help keep your hands (and tools) warm while you're working.

COOL CUTTERS

One of the most temperature-sensitive machines in the workshop is a laser cutter. The laser tube is fragile. The coolant used in lasers can freeze in the winter, cracking the laser tube, bursting coolant pipes, and damaging pump diaphragms. On top of this, bearing grease will stiffen in cold weather, increasing the chance of a stepper motor stalling, and contraction and expansion from temperature changes can throw off the alignment of lenses and rails. Running the laser when it's too cold can cause the tube to crack from thermal shock.

PRINTER PROBLEMS

3D printers are sensitive to cold, but not necessarily in the same way that other precision machines are. Issues with thickened bearing grease and frame warping do exist for 3D printers, but for machines without an enclosed chamber, the temperature of the environment can be critical for a successful print. If the room is too cold, prints can warp and fail, and bed adhesion might also be affected. For resin printers, the temperature of the resin affects the cure time, and a cold resin tank can cause unexpected failures on previously successful prints. A good solution for 3D printers is to (if possible) build an enclosed area just large enough to house the printers and their accessories, and then heat that area using the printer's bed heater or an incandescent bulb for some time before you start printing.

the dashboard switch is easily accessible while seated, and connect a 3D-printed battery connector that matches your power tool batteries to a 12V DC regulator. Connect the battery, turn on the heaters, and warm your heart by making comfortably.

It's one thing to pump heat into your workshop, but if you don't have any insulation, then you're just going to be throwing away money. Use whatever you can to improve the insulation in your workshop. Cover shed windows with bubble wrap to reduce heat loss through the glass, and use the best insulation you can get your hands on to insulate walls and doors. Even

"COVER SHED WINDOWS WITH BUBBLE WRAP"

layers of cardboard will help reduce heat loss through walls, ceilings, and floors. However, cardboard will absorb moisture and will lose it's effectiveness when wet. Foil, bubble wrap, or foam insulation is a much better option if you can afford the extra expense.

Draught excluders and neoprene foam tape can be very useful if your workshop is in an outbuilding or shed where doors and windows don't necessarily fit as closely into their frames as they do inside a regular house. Plugging up those leaky seams can go a long way towards reducing heat loss.

HackSpace magazine meets...

Mike Carroll

Inventing for pennies

Mike Carroll is an author, teacher, and co-creator of a whole system of building electronic circuits using commonly found items.

He's bringing up the next generation of engineers in his school district just north of Philadelphia, and if you have some pocket change and a bit of cardboard, you too can put together the building blocks of an electronic engineering system. Check out **scrappycircuits.com**, buy the book, *Scrappy Circuits*, and be amazed by how much you can do with some paper clips, aluminium foil, and cardboard, and rethink that expensive Christmas present you were going to get your nieces and nephews. →

Above
Michael Carroll: teacher, author, wearer of jaunty T-shirts

Left
No crocodile clips, no problem (as long as you have paperclips and aluminium foil)

HackSpace Hi Mike! First of all, tell us a bit about Scrappy Circuits.

Mike Carroll Scrappy Circuits is an innovative and scrappy way to learn about electrical circuits for less than $1 per person.

It came about because there was a need for it: I wrote a book previously called *Dewey Mac*, and I sold kits to accompany the book. And that was a lot of work. And I always gave out the parts [list] to people so they can buy their own kits. But you know, some communities don't have a RadioShack nearby. Some communities don't have an easy way to hop online and purchase these items. So I was trying to think about what is common in most communities that can be turned into a kit. And that was kind of the origin of Scrappy Circuits, which really started when a friend of mine, Erik Herman, invited me to come to Cornell University, where he worked. He was part of a group at Cornell that had inherited a whole curriculum, and they wanted projects to go with the curriculum.

And they had funds for the curriculum projects for about $2 per person, every time somebody borrowed it.

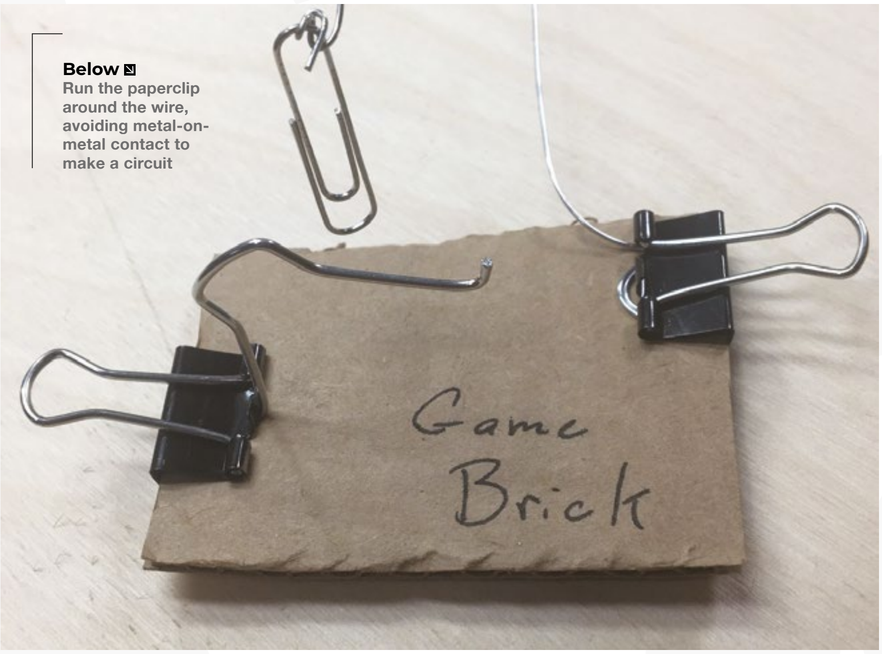
So, they had to have dynamic projects that could easily be refurbished for about $2. That led to coming up with a bunch of different projects.

HS I see a parallel with Raspberry Pi and Raspberry Pi Pico here: I have a couple of Pico projects here at home that I don't expect to last forever, but I don't care because the hardware only cost me £4.20.

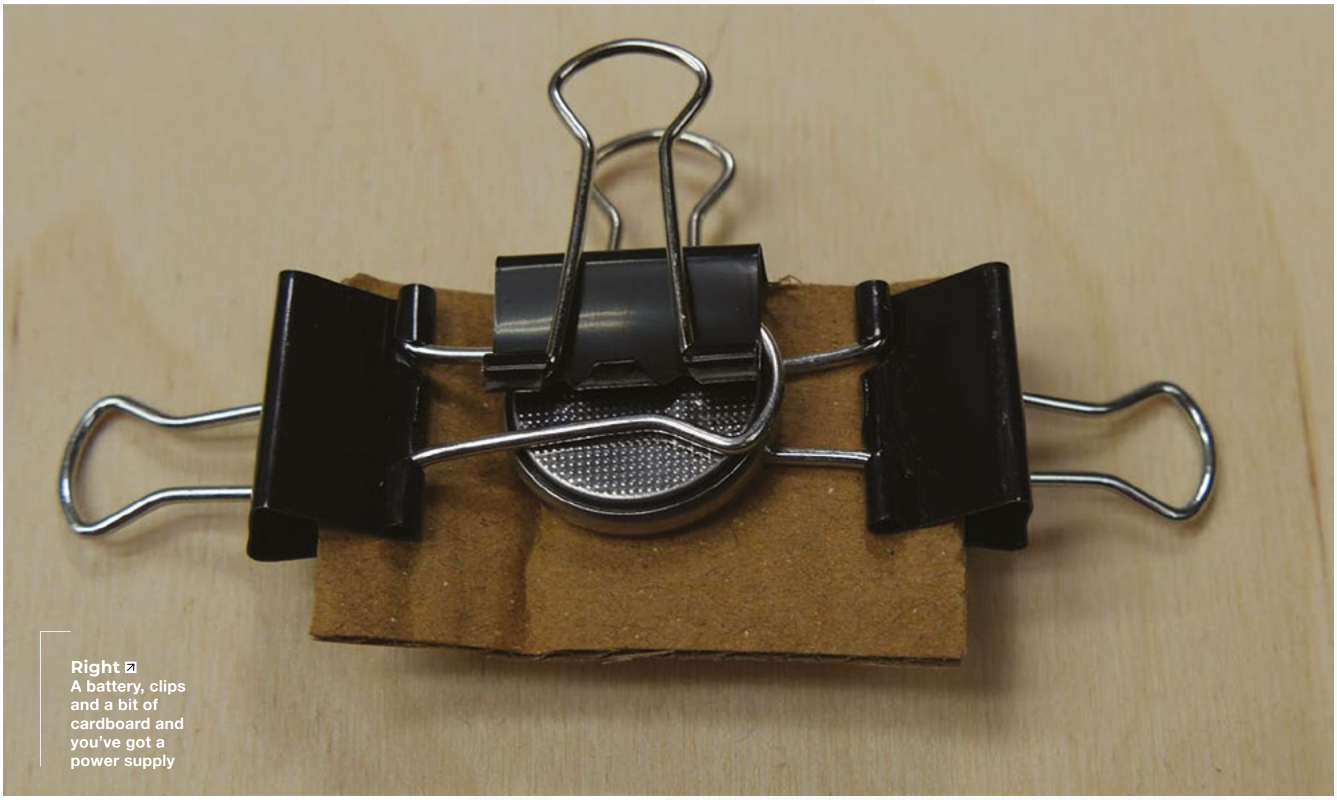
MC When education is a goal, the price point has to be low because if you're talking about a classroom of 20 students, whatever the item is, you're multiplying the cost by 20. And that dollar or two extra, when multiplied out, makes a huge burden on an educator or an education system. So that was really the reason for keeping it so inexpensive.

Another one is that you can get more use out of inexpensive things. I remember, my dad had an expensive set of binoculars when I was growing up. I had to ask twice to take them out of where he stored them, and handle them with kid gloves. But that shouldn't be the case for something like a pair of binoculars – they are a tool for learning,

Below
Run the paperclip around the wire, avoiding metal-on-metal contact to make a circuit



Below
Roll, kick or hit the ball into the V-shape made by the bent clips, complete the circuit



Right ↗
A battery, clips and a bit of cardboard and you've got a power supply

you should be able to just grab it and walk through the woods and look at everything. I learned a lot less because they were nice, expensive ones than I probably would have with a $20 pair of binoculars that I could throw in a bag and just take with me on hikes.

After that time at Cornell, my friend Chris Connors is really the one who realised that Scrappy Circuits is something that could live on its own, well beyond this project. He's a co-creator. He knew I like to write and he really championed me to build it up, and blow it out, and write about it and kind of get it out there to more teachers and beyond.

HS Your bio says that you're an instructional coach; that sounds like you do a lot of teaching, but you're not a teacher. Is that right?

MC I am a teacher. I'm a union member, contracted as a teacher, but I don't have a class of students. I work in a district that has 650 teachers, and I support them. So I go around, find out what people need, and help them to get it. Also my district just launched a sixth grade class called Intervention Lab, which are really based on a lot of the work I've been doing with Scrappy Circuits and other things. So I was a big part in putting that curriculum together with the team, a big part of it.

> "I am a teacher. I work in a district that has 650 teachers, and I support them… I kind of call myself 'the details guy'"

And in launching them, a lot of my days are just running around supporting those classes, supporting all the teachers, supporting any initiative that the district is doing… I kind of call myself the 'details guy.

HS For 650 teachers, that's a lot of details.

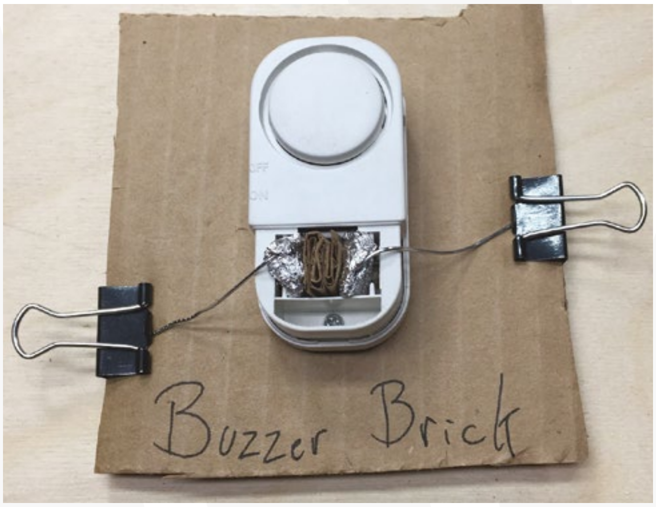
MC It *is* a lot of details. You just do the best you can. I was a third grade teacher in this district for a long time. Fourth grade teacher, first grade teacher. So I have a lot of great relationships with teachers here.

As crazy as it sounds, it works.

HS Can you talk us through the Scrappy Circuits system? What is it?

MC So when my friend Chris and I, we were at Cornell, they gave us their credit card, and we filled up two shopping carts worth of stuff from the dollar store. For a few days, we were in a room and we just took everything apart, and figured out what parts we can use. It was a wild mess, and there wasn't really a roadmap through it.

We came up with the idea of starting with an LED tea light, taking that apart, and then building what we call five core bricks from that. So, the first would be your battery brick: you use the three-volt battery inside of the LED tea light to build your brick and add the binder clips to it, →



Right
If you can get hold of one, Mike recommends a cheap window alarm as a good item from which to scavenge components

which are kind of your terminals throughout Scrappy Circuits.

You take the LED inside the tea light and clip it to the cardboard to build your LED bricks. So now you have your battery and your LED, and your binder clips are kind of your terminals. You can get some alligator clips, connect your battery to your LED, and then you have your first circuit.

Whenever I do a workshop, we start with that circuit, then I ask 'what's missing right now? What's missing from this circuit?' Eventually everyone will come up with the answer: a light switch.

Switches are my favourite part of a circuit. That's how we control electricity.

So then we build the binder clip switch and a push switch. The binder clip switch is a toggle switch; it's just two binder clips, one's down, you flip the other one down, they're connected, electricity travels through them; you flip one up, and then it's off. The push switch is similar, but it has a paper clip that you kind of bend up so it stays in that bent-up position. And then, when you push it down for that moment, it's on. But when you release, it's off. So you have your momentary switch and your toggle switch.

Everyone knows your light switch is a switch, but they don't realise that your keyboard keys are switches, or your remote control buttons are switches, or your radio dials. I like going into those two genres of toggle and momentary, and the different ways you could do them and the different ways you can operate them.

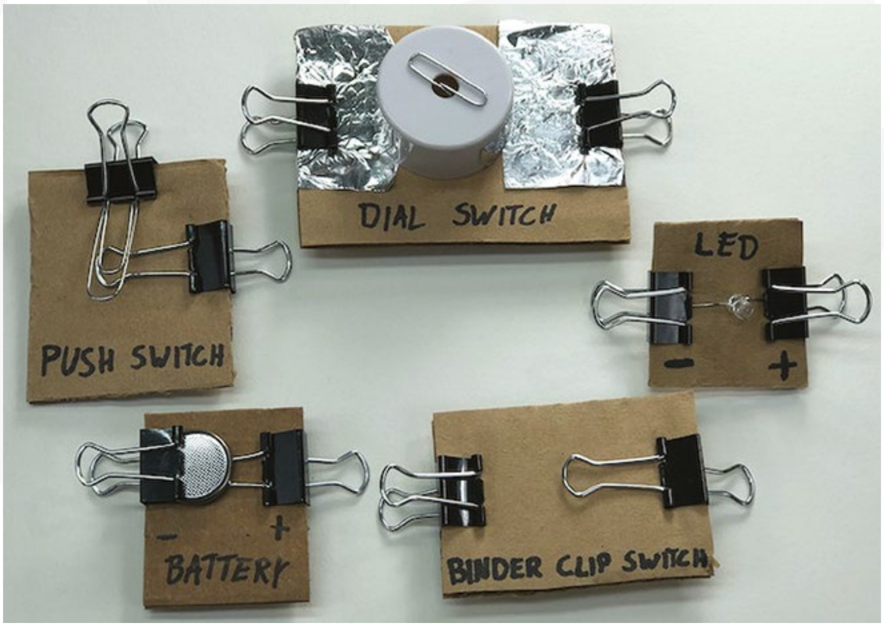
The fifth brick is a dial switch; you use the cylinder of the LED tea light and you cover about half of the lower edge in tin foil so that when you turn it, it'll bridge the sides, so it's either on if it's bridging or off if it's not. I love that one. Because that just shows to students how crazy and creative you can be with switches and how simple they are.

One of the lines I use in the book is that a switch is like a one-step dance. It really is just one movement; it's either on or it's off, it's on or it's off. It's not this complicated process. It's just two conductors, disconnecting and connecting. So, once I think students realise how simple switches can be, and how they can be used in different ways, that really unlocks a lot of creativity for coming up with really cool projects in the whole Scrappy Circuits ecosystem.

HS I love that quote of yours that a switch is the domestication of electricity.

MC That came from the book *Sapiens*, by Yuval Noah Harari. It asks why humans are where we are, compared with animals. How come frogs haven't built bridges? How come turtles haven't landed on the moon? One of his points was when we learned how to control fire, that was really important. And you know, controlling fire led to controlling electricity. And that's probably why turtles haven't been to the moon – yet.

Above
This brick completes a circuit when an aluminium foil ball lands in the tube



Above
There's enough here to get a class of kids lighting up LEDs and learning why resistors are important

HS You must see thousands of kids' projects every year. What sort of things do they come up with?

MC A lot of the time they use Scrappy Circuits as the guts for an art project. I really love seeing those. They'll be drawing a picture or making clouds and a diagram or something. And they're just tucking Scrappy Circuits inside and behind all the parts, so the clouds kind of light up or the eyes light up of their picture.

Those are the things that I really, really love. I ran a summer camp this past summer, called Camp DIY Arcade. It was fun, because we actually went to an arcade – we held the whole thing at a retro arcade. And we looked inside the machines, saw how the wiring works, talked about how the computer works, the coding. And then, at the end of the week, the kids actually coded their own games, and then built their own arcade controllers for the games.

I always love looking at switches and circuits because they're essentially the same thing. Every arcade cabinet is gonna be a little different. Like, some are gonna have more switches, some are gonna have rolling balls, some are gonna have, you know, joysticks, but you look under the hood, and it's all wires leading to a computer in the end. That idea really opens up kids' creativity. That one-step dance can be a button, or it could be a laser that triggers a little solar panel or solar cell. You can really get creative with how you're controlling and interacting with your inventions.

HS You give credit to Forrest Mims's *Engineer's Mini-Notebook* in the *Scrappy Circuits* book. Was that a big influence for you?

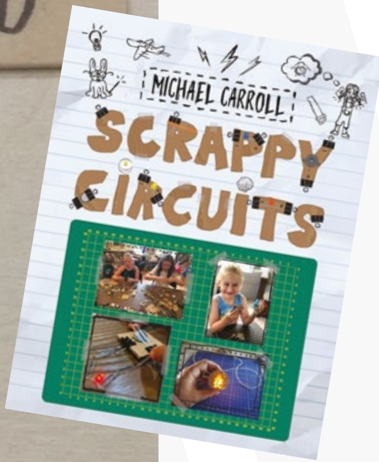
MC I had the RadioShack kit, you know, where you wire things and [it] has all the dials; he wrote the book for that, though I didn't know his name back then. But it was just something I was really into and enjoyed using. And then, as I got older, I refined this interest in using common items in uncommon ways. I think I was always kind of living in the maker world and living in the teacher world. The maker world is really resourceful and scrappy. And the teacher world kind of needs that. And they don't have the funding for the $300 item, they need to find the $3 alternative, or the $2 alternative.

And that's when I just devoured all of his books, and tried to see anything that was applicable at this elementary level that could work in Scrappy Circuits.

HS The Dollar Store Challenge you set yourself at the start of the Scrappy Circuits project sounds like fun. What other common bargain objects would you recommend people get their hands on if they want to tear them down and build stuff out of them?

MC If you went to the dollar store with more than $1, I would definitely recommend getting a window alarm. The dollar store window alarms usually have a piezo element in them, which you could do a million things with: they amplify guitars, you can produce electricity with them, and they →

Below
The dial switch uses the plastic shell of an LED tea light



Above Scrappy Circuits the book owes a lot to Sylvia Martinez, who wrote the book *Invent to Learn* with Dr Gary Stager. Mike wanted to keep a lot of the book's content free on the internet at scrappycircuits.com, an idea that Martinez supported from day one.

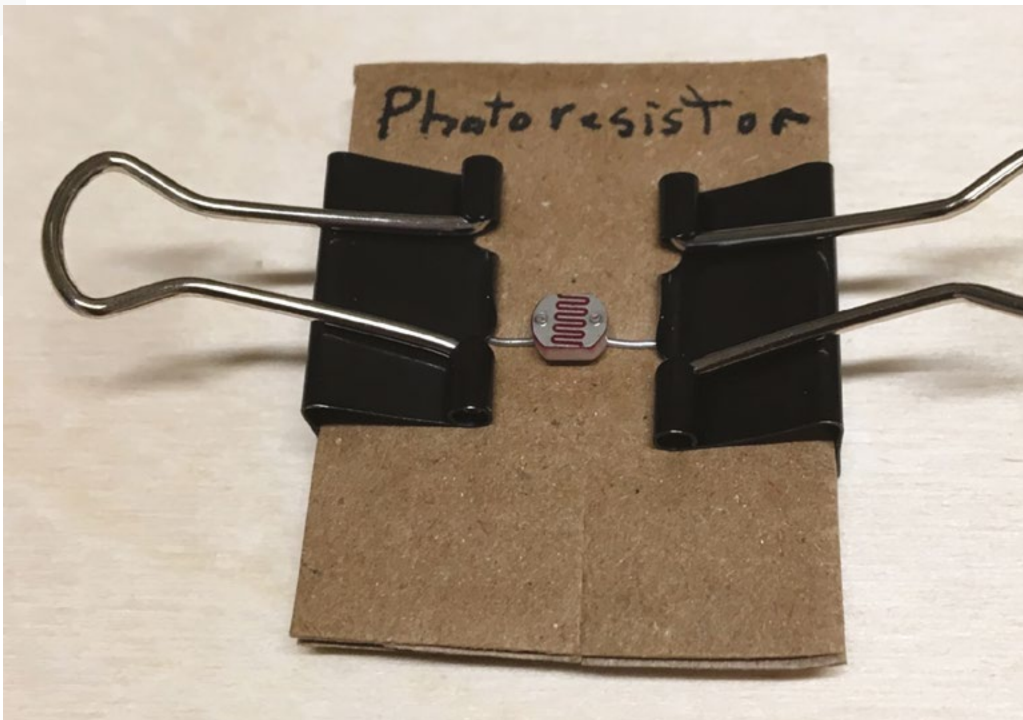
produce sounds, which is what they do in a window alarm. So there's a bit in the *Scrappy Circuits* book about using one of those to get an LED to blink for a moment.

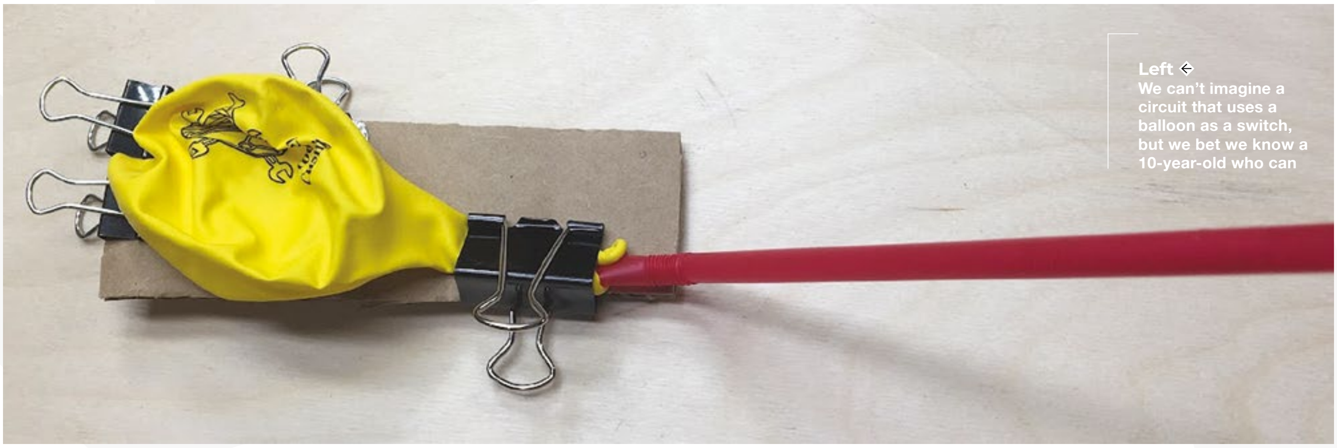
And it also comes with a magnet and a magnetic reed switch. So you're kind of getting like three great parts that you could tinker with for $1.

They're used a lot in the *Scrappy Circuits* book – I think one of the first beyond the core bricks is the buzzer brick, which uses the window alarm as a buzzer. I always wanted to think about arcade games; if you're shooting a basketball tin foil ball into a cup, a light lighting up is cool, but a buzzer buzzing as well is a little bit cooler.

HS How did you initially get into making things?

MC Very simply, I was bored. I think boredom is always step one in this recipe. And I was bored. I was playing





Left
We can't imagine a circuit that uses a balloon as a switch, but we bet we know a 10-year-old who can

basketball outside. And I noticed that our vacuum cleaner was sitting next to the trash can. And I asked my mom what happened, she's like, 'Oh, the vacuum cleaner broke.' So here I am, like, looking at my dad's workshop looking at the vacuum cleaner, and I'm bored. I just grabbed a Phillips head screwdriver, [and] started taking it apart.

And I realised, very simply, that a belt had just slipped. So I immediately take it apart, immediately realise the belt slipped, immediately put the belt back on, put everything back together, really not using any high-level skills. And I just walked back in and I was like, 'Oh, Mom, I fixed the vacuum cleaner.' She gave me one of those 'Sure Mike, go play basketball' looks, so I plug it in and I show her. That was kind of like having a superpower of taking things apart and putting them back together and fixing them a little bit.

And I've always been curious [about] how things work. So that was kind of the beginning of it. And then I kind of didn't stop taking things apart and putting them back together in different ways.

HS But only broken things I guess: I was never allowed to take apart anything that was fully functional.

MC Yeah, I've never trusted myself with that. I grew up with a bunch of car friends who would be happy spending the weekend taking their car engines, or whatever, apart and putting them back together. I never trusted myself to that level of taking things apart.

HS What you said about switches reminds me of the NES game, Duck Hunt. The one where you shoot the TV with a little gun, and somehow it knows if you've hit the target.

MC I read about how that works. I'm always curious about that stuff.

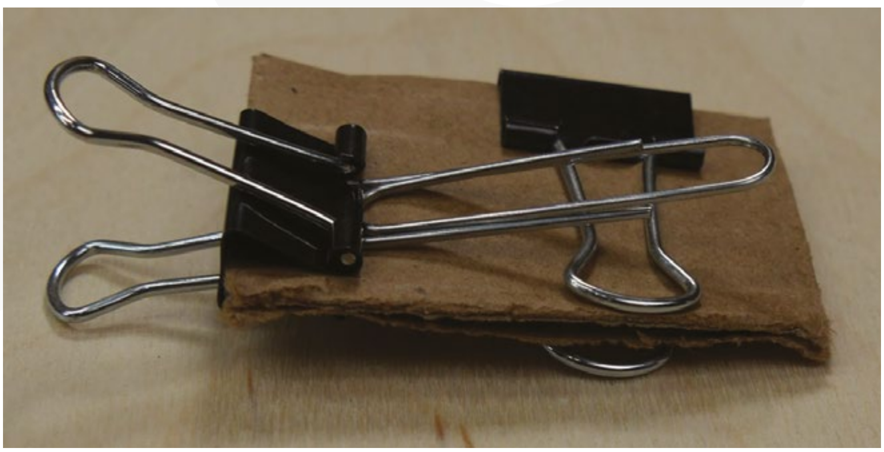
And I'm trying to remember it, but I think what it is is the gun itself has like a little light sensor. And when you pull the trigger, the screen actually goes black for a quick second. And there's just white where the duck is. If the light sensor sees that white, it knows you hit the duck. And if it sees the black, it thinks you didn't hit the duck, if I remember correctly. Don't quote me on that!

HS That's brilliant. Simple and brilliant.

MC It really is. Nintendo was better than a lot of the others because they had games like Duck Hunt. But it really is just that simple, [a] genius idea that I'm sure some guy whose history is forgotten came up with that really launched Nintendo in the US.

HS You know, there's some kids now in your school district who are going to come up with the next big, cool thing.

MC That's my goal. It's a long-term goal, but I hope one day to fulfil it. ◻



Right
A push switch is just one of the many ways humans have domesticated electricity

CUTLERY

Fine metal, cut and bent into new shapes and then polished up, shows off the intricate detail of the original design, making upcycling cutlery a worthwhile pastime, finds **Rosie Hattersley**

Rosie Hattersley

@RosieHattersley

Rosie Hattersley writes tech, craft, and life hacks and tweets **@RosieHattersley**.

Using the right knife, fork, and spoon when dining acknowledges social mores and demonstrates both knowledge of and adherence to such rules. Some of us cringe when faced with unfamiliar place settings and cutlery arrangements but that, of course, is sort of the point, putting the oiks in their place with a visible demonstration of their struggle to understand the codes of a certain class.

If you are happy to dispense with such social niceties and rewrite the etiquette rule book, you might very well have some fancy-schmancy cutlery that could be put to use in all sorts of innovative ways. Among the many craft and art ideas to be found online are key rings, bracelets, candelabra, hooks, smartphone stands, and both picture frame components and subject matter to be framed. Along with some ideas for such items, blogger Dishfunctional Designs offers a video on how to prepare flatware, aka cutlery, for a new use: **hsmag.cc/CutleryPrep**.

With craftsmanship and quality metal at their heart, cutlery sets are a perennially popular wedding gift, and it makes sense to preserve the designs resulted from the skilled shaping of prized sterling silver and steel when upcycling or repurposing unwanted items. Sterling silver is the favoured material due to its non-corrosive, non-reactive properties, with Sheffield the most famous cutlery manufacturing centre, employing thousands until the industry was undercut by China and other overseas production centres.

> "The concept of using a knife, fork, and spoon while dining is a Georgian invention"

Needless to say, the traditional cutlery set of western dinner tables has existed for a matter of hundreds, rather than thousands, of years. Forks are a relative newcomer to the cutlery set, whereas knives for hunting, slicing, and dicing date back to Palaeolithic times or earlier. The concept of using a knife, fork, and spoon while dining is a Georgian invention, making it less than 400 years old.

ANTIQUE SPOON PLANT MARKER

Unusual markers and stakes lend themselves to all sorts of uses, and the straightforward concept of flattening spoons in a vice, or with a metal press, before stamping out a message or label is pretty sound. Since stainless steel is relatively resistant to discolouration, you could create an attractive herb patch following Shrimp Salad Circus' suggestion of using the upcycled spoons to mark out which herb is which. Other ideas that employ flattened and stamped spoon bowls include personalised coat hooks, with the stem of the spoon bent upwards and nailed to a length of wood to hold a jacket or hat.

Project Maker
Shrimp Salad Circus

Project Link
hsmag.cc/SpoonPlantMarker

Left Stamping letters on steel is very satisfying

ULTIMATE SURVIVAL TOOL

Project Maker
Funtogether

Project Link
hsmag.cc/UltimateSurvivalTool

"This survival tool, pounded from the dead remains of an old spoon, is a very useful object, but its usefulness depends on the imagination of the user", explains maker funtogether, as he elaborates on how it's up to the user rather than the tool to prove the upcycled spoon's worth. He begins by chopping off all but the bowl end and the first inch of the stem using a hacksaw. The stem is then bent back on itself to form an enclosed hook. Next, the bowl is hammered flat between two cloths, any blemishes polished out, and its edges filed down to become sharp. String the whole thing up with a shoelace or thick cord, and wear it around your neck, advises the maker, where it can be pressed into service when needed. Suggested uses include a sun reflector with which to send Morse code messages, as a knife or carving tool, a tool for digging up treasure, and a condenser/reflector for helping start a camp-fire, while one commenter declared he could find multiple uses for such an item in his job as a flat roofer. →

Right Multi-tool or dangerous necklace? You decide

Upcycled Ram's Head Sculpture

Everything that David Mcauley makes had its origins in another form, most of it intended for the cutlery drawer. The Alberta, Canada-based artist has a fantastic gallery of sculptures, wearables, and vehicles which Facebook users can most easily browse at **hsmag.cc/DaveMcauley**, since his WordPress website (**theeupcycler.com**) is a little shaky. TheeUpCycler's inventive designs first caught our eye on Instagram, particularly this ram's head fashioned from a combination of "heavy truck differential horns, a natural gas engine crank-shaft, and upcycled cutlery." While his work features plenty of spoons, David often makes creative and playful use of various kinds of old knives too, with owls, eagles, a peacock, and polar bear sculptures on public display, backgrounded by stunning mountain lakeside scenery.

> "TheeUpCycler's inventive designs first caught our eye on Instagram"

Left ⬗ You just need an old cutlery drawer and a welder

Project Maker
David Mcauley aka TheeUpCycler

Project Link
hsmag.cc/TheeUpCycler

JEWELLERY

Upcycled jewellery is perhaps an obvious use for unwanted sterling silver, given the material's popularity when making rings, necklaces, and more. While cute elephant designs created by taking off the head and part of a tine from a fork are fairly common online, Australian jeweller Jen, aka Gitane, seeks out especially beautiful cutlery designs, such as Japanese knives with intricately carved handles, and uses her jewellery designs to showcase the original maker's handiwork. Examples include a brooch made by flattening and spread-eagling the tines of a fork which then cradles a delicate dragonfly. Attention to detail, and an eye for how the repurposed items can enhance whatever they are paired with, ensure her in-person makers' market stall quickly sells out.

Project Maker
Gitane Jewellery

Project Link
hsmag.cc/Gitane

Left Giving silver cutlery a new life

RECLAIMED CUTLERY CHAIR

Project Maker
Osian Batyka-Williams

Project Link
hsmag.cc/CutleryChair

The Reclaimed Cutlery Chair was created from more than 150 knives, forks, and spoons and presented as Osian's final year product and furniture design degree at university. Having been aghast to learn that restaurants often discard their cutlery after less than two years of use, Osian set about collecting their unwanted items, as well as trawling charity shops. His aim was to get restaurants to use unwanted cutlery sets again, albeit in a different form. Once assembled, Osian prepared everything for welding, first bending the fork prongs or covering them with a spoon head so they wouldn't prang an unsuspecting user. He used only fairly blunt knives, laid flat, and kept their sharp edges hidden away. Osian spot-welded every item over a three-week period, ending up with a chair sturdy enough to take the weight of a 16-stone adult. □

Right Knives and forks don't sound comfy, but it looks like they are

FEATURE

IN THE WORKSHOP: Bike Lights

By **Ben Everard**

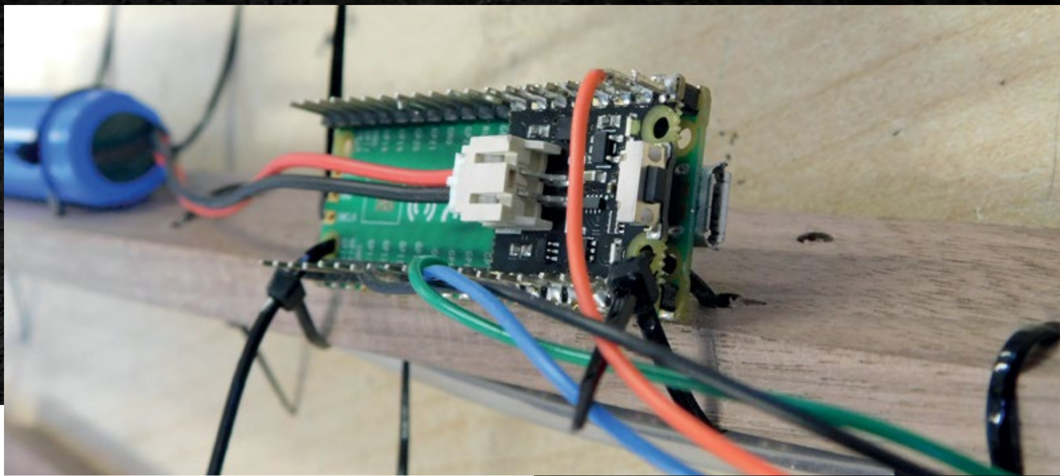
Stay seen, stay safe

There's a project that's been rattling around my brain for quite a few years now – mount a load of LEDs inside a bike wheel so that, as it spins, the wheel lights up. With a bit of mathematics, it should be possible to turn them on and off at the right time to display an image. This is by no means my idea. In fact, there used to be a commercial product that did it, though, as far as I can see, this is no longer available.

The basic idea of a persistence of vision (POV) display is relatively simple. You just need some LEDs that you can control, a microcontroller to control them, and something to sense the current position of the wheel. All of this is eminently doable. The maths to calculate the current position of the wheel might prove to test my rusty high school-level trigonometry, but with enough head-scratching, I think I can do it. The hard part of this project – and the thing that's been putting me off all these years – is building something that is both safe and tough enough to survive a winter of cycling around Bristol.

This year, I've bitten the bullet and decided to go for it. The electronics is simple enough. A Pico has plenty of processing power to do this. A Hall effect sensor mounted on the Pico, and a magnet on the forks of the bicycle, should let it know each time it's performed a rotation (which should let me calculate the speed and, from there, extrapolate the current position).

I've got APA102 LEDs which are, in some ways, similar to the classic WS2812Bs that are almost ubiquitous with maker projects, but the APA102s have



both faster update speeds and faster PWM speeds, both of which should be better for this project.

The final piece of the puzzle is adding battery power. I decided to use a Pimoroni LiPo SHIM for Pico. This solders onto Pico and provides a battery input. It can take LiPo or Li-ion. I went with a Li-ion as it seemed like a safer option.

For waterproofing, I plan on sealing everything with silicone, but that will come later. I wanted to get the electronics working first. I soldered it all up, attached everything to wooden battens that I then cable-tied to the spokes on my bike, and ran a simple APA102 test program.

That all worked just fine. Then it was time to prepare for waterproofing everything. The first thing needed was a way of accessing Pico's USB port. I needed this to apply power, as well as reprogram Pico. Fortunately, you can get waterproof USB extension leads. Or rather, USB leads that end with a USB port that has a waterproof screw cap. The first version ended in a USB Micro-A plug rather than a USB Micro-B plug. It turns out that there are far more variations on the USB standard than I'd come across! Fortunately, this was a quick and cheap part to swap out and back in. However, it wasn't my only mistake.

LiPo SHIM for Pico has a power button. Tap it once to turn on, tap it again to turn off, double-tap it to reset. For almost all projects, this is probably a good thing. However, it's a problem for me because that button will be inaccessible once the device is encased in silicone.

It might be possible to remove the button and solder wires directly onto the pads, which could give me another way of powering the board on and off (as well as putting it in bootloader mode which can be done with a double-tap of the reset button, if you include the right library in your code). However, the pads were tiny and I don't think the leads will stay soldered in place for very long. I need a better solution.

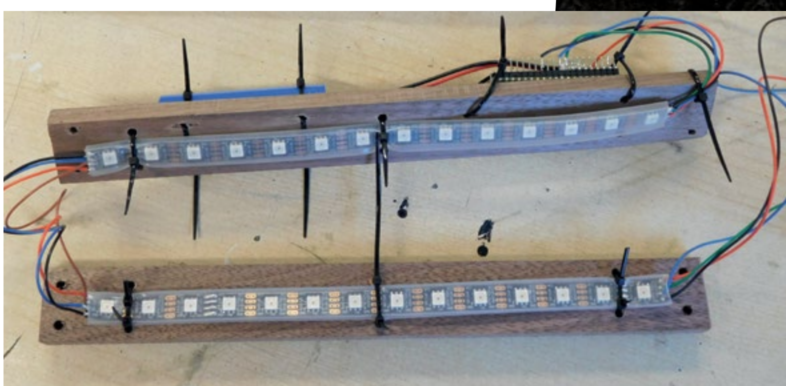
I have to find a way of mounting a power button somewhere

And that's where we are now. I'm awaiting delivery of a Pimoroni LiPo Amigo Pro which, if I have read the spec sheet correctly, should allow me to charge the battery from the USB port (if I connect 5V in on the Amigo Pro to VBUS on Pico, and the power out on the Amigo Pro to VSYS on Pico), and have an externally mounted power button.

This does mean that I have to find a way of mounting a power button somewhere, and buttons were something I had hoped to avoid because, while you can get waterproof buttons, they tend to be only half waterproof. That is, they're designed to be mounted on a box where the outside is exposed to the elements, but the inside is dry. That, however, is next month's problem, and I'm taking it one step at a time. □

Above That pesky start button means I have to rethink the parts

Below The walnut offcuts should be rot-resistant and last in wet weather



FORGE

HACK | MAKE | BUILD | CREATE

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

Light up the room

How to control large numbers of LEDs

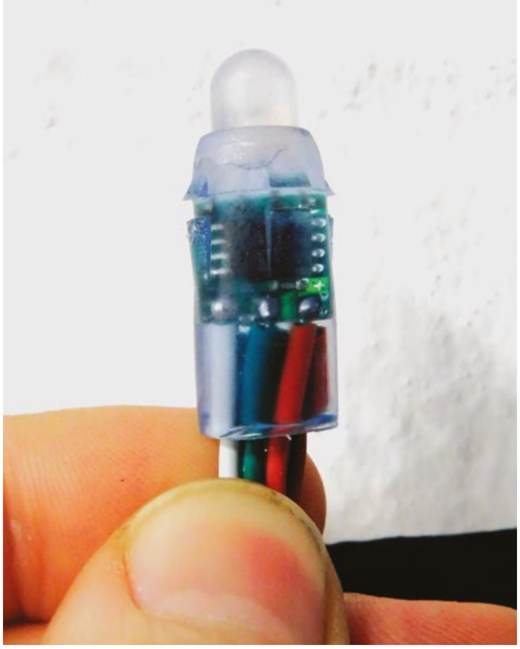
Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Who doesn't like a wildly unnecessary number of LEDs? Actually, it turned out that one person in Bristol Hackspace – where we've been testing out this technology – doesn't. Matt Glasspole, we send our apologies. For the rest of us, let's dive in to how to control a mind-bending quantity of dots of light in near-perfect synchronisation.

When it comes to hacker-friendly colourful LEDs, it's almost always WS2812Bs and their cousins WS2811s. These are cheap, widely available,



Right We're big fans of chains of WS2811 LEDs. They're easy to use, moderately waterproof, widely available, and fairly cheap

and easy to use. The only difference between WS2812Bs and WS2811s is that the former have the control IC and LED in the same package, while the latter have an external chip to control the LED. They both use the same protocol and can be used interchangeably.

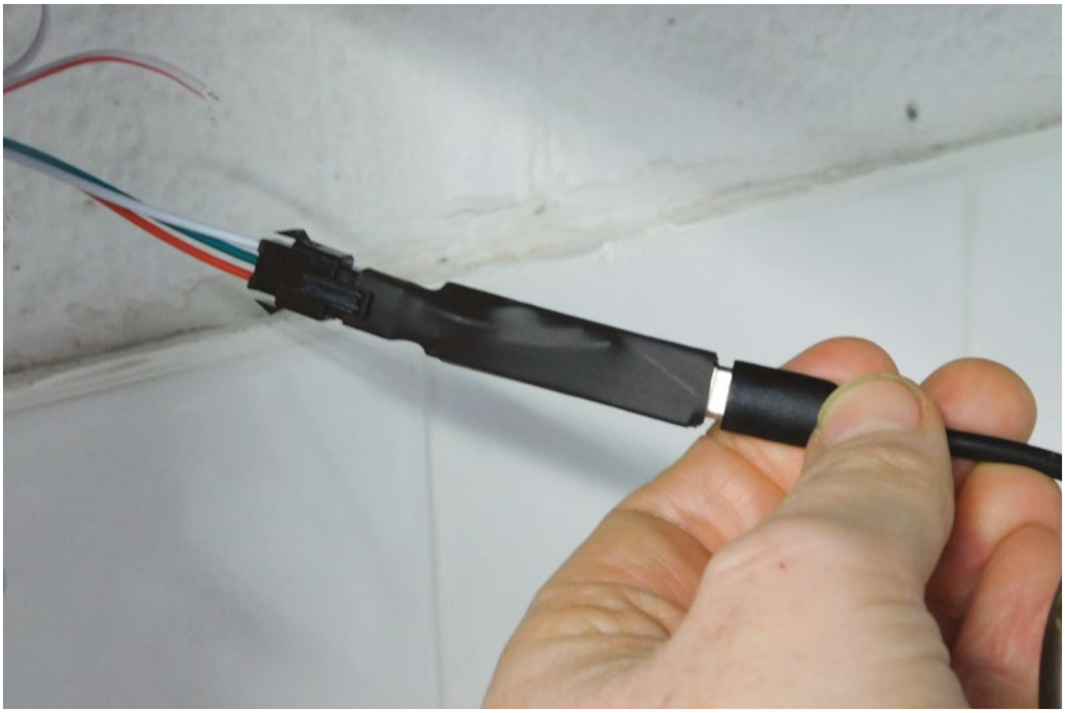
You can control these LEDs (which are commonly known as NeoPixels) from almost any programming language on almost any microcontroller. If you're looking to control a single string of LEDs that's a smallish size (less than 100 or so), then this is a great choice. However, if you want more LEDs, it gets hard to manage with just one microcontroller. If you have lots of LEDs on a string, the refresh rate slows down, and it's hard to spread your LEDs out. Sooner or later, you'll probably reach the inevitable conclusion – you need more than one microcontroller.

You can just run more than one independent microcontroller running different effects on different LED strips. However, thanks to a few bits of open-source software, we can also get our microcontrollers talking to each other to create effects that span multiple separate controllers.

BRINGING IT TOGETHER

WLED is the first piece of the puzzle. This is a firmware for ESP32 and ESP8266 devices that is, frankly, fantastic. You can get up and running quickly and easily without having to compile anything or edit strange config files.

First, you need an ESP device with at least one exposed GPIO pin, as well as power and ground. Almost any should work. Our setup uses the ones we had to hand, including a SparkFun smôl and D1 mini.



Left One of our controllers is a SparkFun smôl ESP32. We soldered on a connector and wrapped it in heat-shrink

Plug your ESP device into your computer's USB port, and head to **install.wled.me**. Here, you can press Install to install WLED on your device. If your board doesn't show up in the Connect box (which should pop up in your browser after you press Install), you might need to unplug it and plug it back in. If it still doesn't show up, then you may need to install a drive for it. Click on the No Device Found link for more information.

Each WLED instance will automatically create its own Wi-Fi hotspot to which you can connect to manage the device. However, it's usually easier to connect WLED to an existing network if there is one available (this will also let you manage multiple devices at the same time). Once the web app has finished installing, it'll give you the option to configure Wi-Fi, so you can set this up (if you don't do it now, you can still do it later by connecting to the device's access point, and then configuring external Wi-Fi in there).

Once you have all your WLED instances flashed, make sure you can connect to them all. The easiest way of doing this is via the WLED mobile phone app (by Aircoookie on Android and Christian Schwinne on iOS).

Power up all your devices and now they should all appear if you scan for them in the app. This will give you the IP address of each of the strips. →

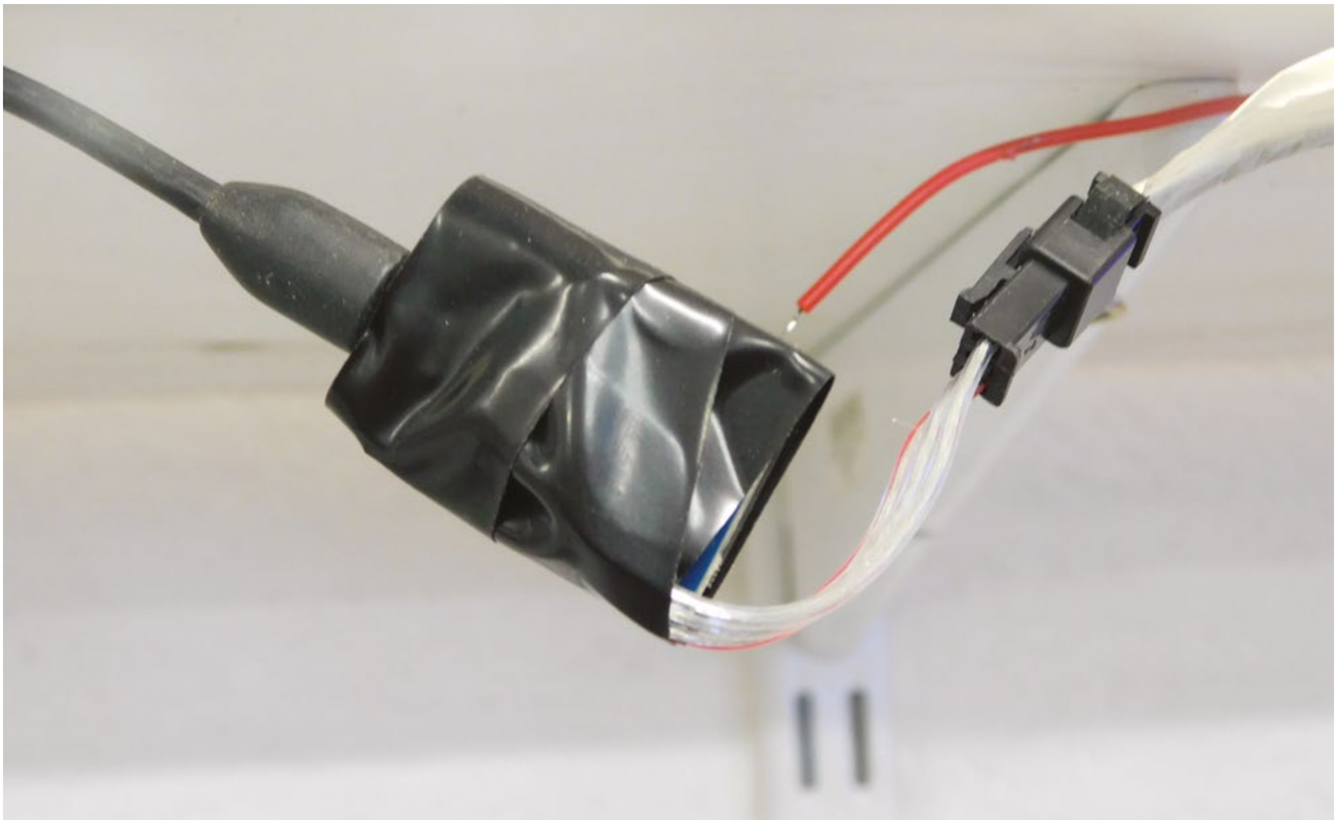
> "Sooner or later, you'll probably reach the inevitable conclusion **– you need more than one microcontroller**"

UNIVERSES

The underlying protocol we're using – DMX512 – comes from the days before it was common to add huge numbers of LEDs to things. Way back in 1986 when the protocol was first used, it was for controlling big expensive lights. It would probably have seemed laughable that anyone would have wanted to control more than 512 of them, so a hard limit of 512 lights per universe is encoded in the protocol.

However, fast-forward to 2022 and LEDs are cheap, widely available, and easy to use. Not only that, but it's common to pack three of them into a single package (one each of red, green, and blue), so a single universe can contain 170 RGB LEDs. Fortunately, you can have multiple universes – each one just has its own number.

This historical anachronism doesn't need to constrain us too much, since it's mostly hidden behind the scenes. If you add more than 170 LEDs to a WLED instance, it will simply roll over to the next universe. By starting each strip at the start of a universe, it's an easy way of addressing each strip.



Above
D1 Minis are cheap ESP8266 controllers that work well with WLED. We've wrapped this one in electrical tape to give it a bit of protection, but we should also remove those stray power leads!

OTHER OPTIONS

There are a few ways to control your LEDs. If you don't like xLights, you could try the following:

• **LedFx** – This is an app that can run on either smartphones or regular computers, and is mostly used for creating sound-reactive light shows. Install it on your phone or computer and it'll use that device's microphone to configure the LEDs. Effects include equalisers and general flashing. It's well worth playing with as it's really easy to use and creates some impressive effects.

• **PyDMXControl** – Write Python scripts to control your lights in any way you like.

• **Falcon Player (FPP)** – We've not used this very much, but as far as we can tell, it works well for playing sequences and putting multiple sequences together into longer performances. It can work with sequences created in xLights.

• **WLED** – You can control multiple WLED instances from a single WLED instance. This basically lets you act as though all your LEDs are in one long chain.

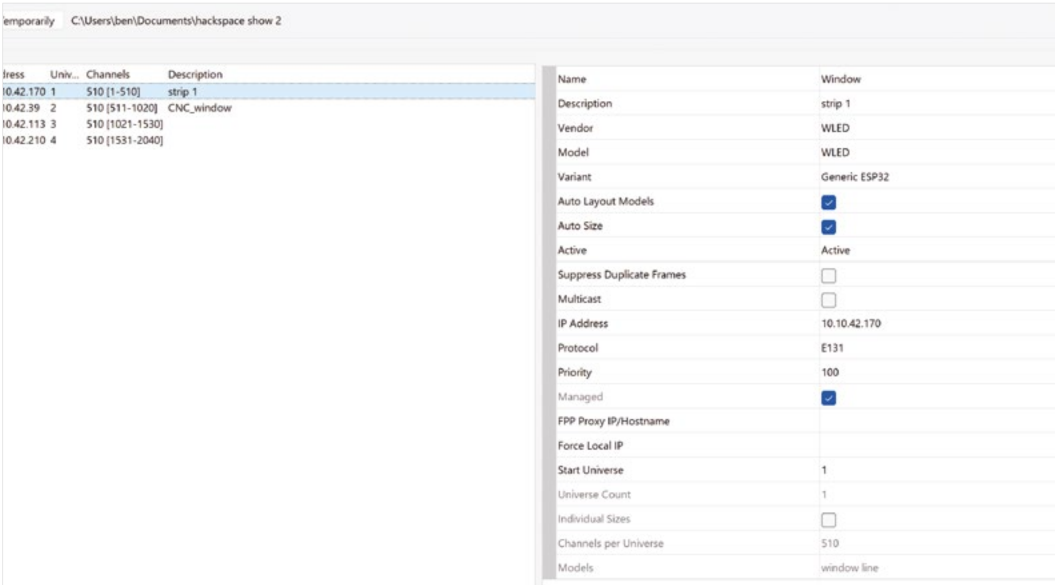
Using these, you can access them from a computer or any other device on the network – the WLED configuration is a web page hosted on that IP address, so just point your browser there.

If you don't want to use the app, you need to find out the IP address for each strip. If you have access to the admin panel of your Wi-Fi router, you'll be able to see it there.

We've now got our microcontrollers set up, and it's time to add the LEDs.

You'll need to solder connectors onto your microcontroller. You can either do this directly by cutting the connector off the LED strip and soldering this onto the pins, or you can add a connector to your microcontroller. Unfortunately, there's no standard for whether 'data in' on the strip is a pin connector or a socket connector, so you have to look carefully at your LEDs and see which way your strip is wired.

Either way, you'll need three connectors – 5V power, ground, and data. By default, data is on GPIO2 (which is often labelled D4, but check the pinout of your board for details). You can use any data pin, but not all support hardware WS2812B.



Left You don't need to fill in all the controller fields. Here are the key ones from ours

On some it has to be 'bit-banged', which may give worse performance. However, we've been using a bit-banged strip of 50 LEDs without noticeable problems.

At this point, you can power on your ESP board, and if you've used GPIO2, the LEDs will come on. If not, you'll need to set the GPIO pin. Open the IP address of the board either in a web browser or the WLED app, and in config > LED Settings, change the GPIO number and the number of LEDs that you have.

> **If you don't want to use the app,** you need to find out the IP address for each strip

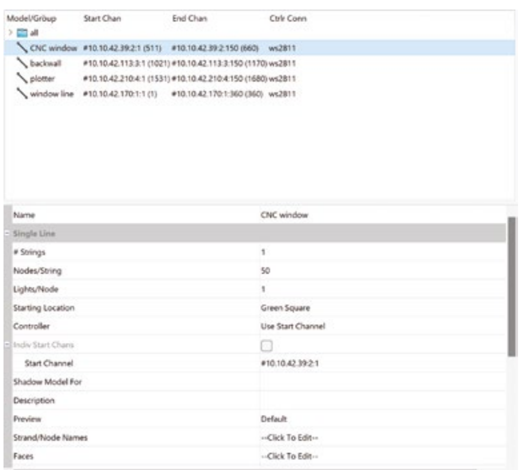
You can use this interface to set colours or select patterns, which is a great way of controlling a single strip of LEDs, but let's up the stakes and control them all from one place.

XLights is a bit pernickety to set up, but once it's set up, and you've got your head around the layout and workflow, it's pretty easy to use. First, you need to download it from **xlights.org**. It's also a good idea to install the QM Vamp Plugins from the download page as this gives you much more powerful options for timing your sequences to the beat.

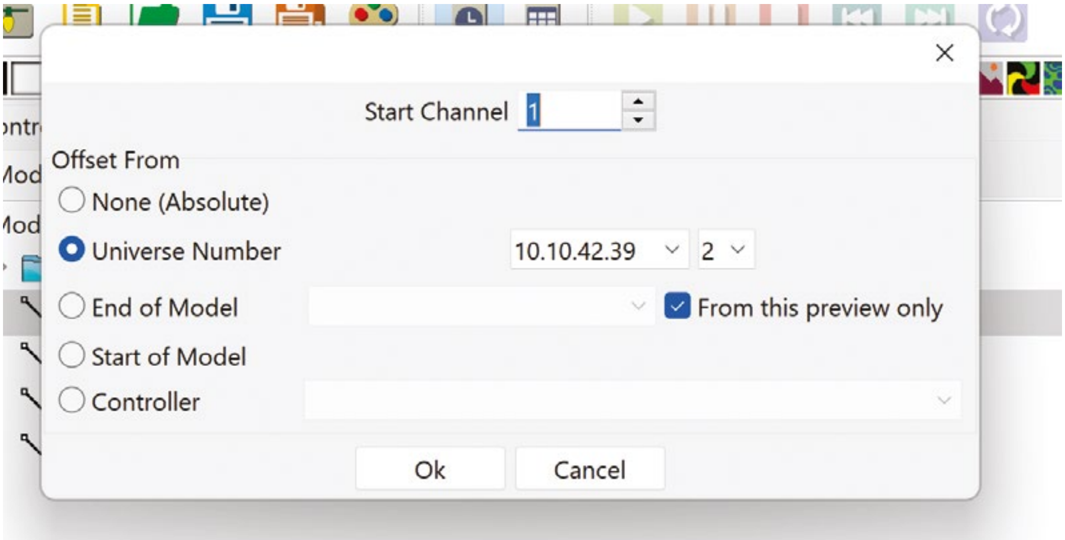
There are three parts to the setup: the controllers, the layout, and the sequencer. First, we'll set up the controllers, and as we do so, we'll set the universe numbers in WLED.

For each WLED instance, you'll need to do the following:

- Set the universe number in WLED > config > sync. The first WLED controller should be 1, the second should be 2 (or higher if you have more than 170 LEDs on controller 1 – see 'Universes' box).
- In xLights, go to the controller, then tap or click Ethernet to add a new networked controller. The key things you need to add are Name, Description, IP Address, and Start Universe. →



Left The layout settings can be a bit confusing. Here's one of ours

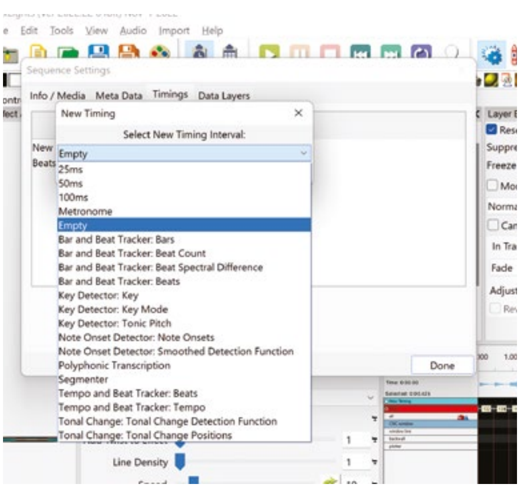


Right If you click '…' in Start Channel, you should see this box that lets you select the IP and universe number

Once you've added each controller, it's time to organise the layout in the Layout tab. The window will be split in half with a big black rectangle on the right-hand side. This is where you can draw your LED strips. If you want, you can take a photo of where they are and include it here, then draw them in place, but we preferred to leave it black and do it as a top-down plan.

BUILDING A PLAN

You can draw each strip in place using the tools directly above the black preview window. We used the line tool as we were using LED strips, but if you're using some pre-built shapes, then you can use other tools.



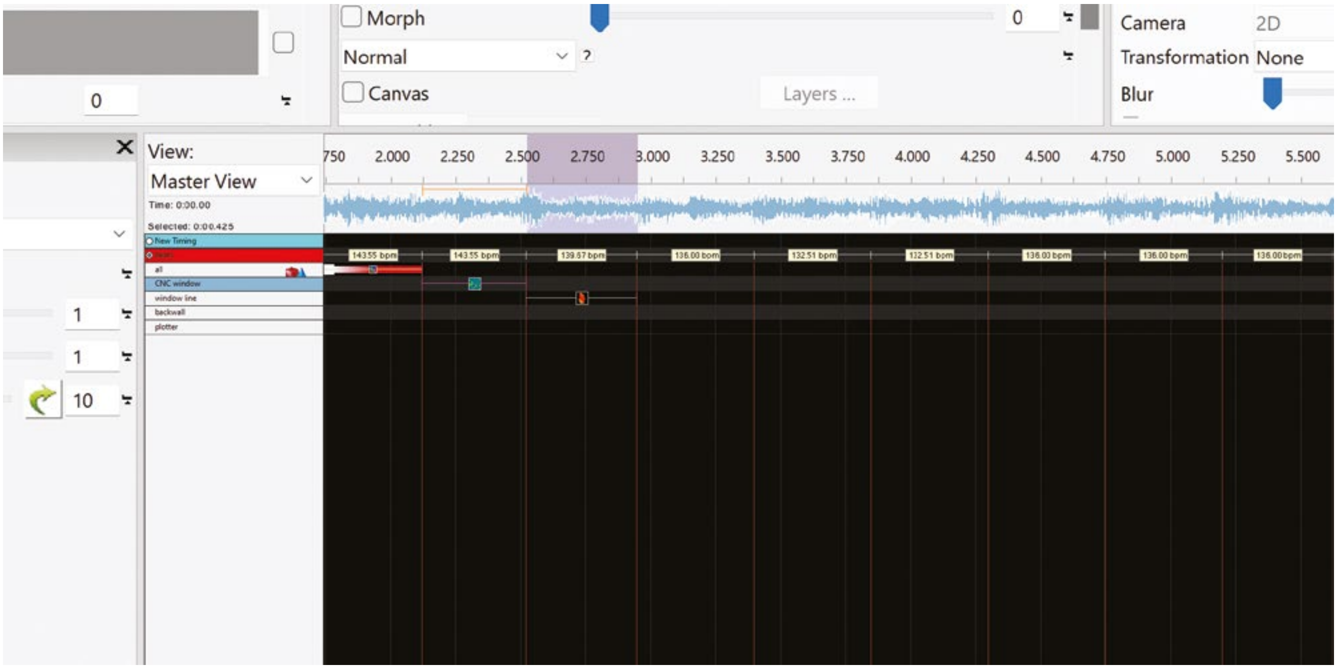
Right Getting the patterns in time with the music is a big challenge, but the VAMP plugin gives you some options

Once you've drawn a line on, you can fill in the details in the bottom-left box. You'll need to enter the name and nodes per string (this is the number of LEDs); in Controller, select 'Use Start Channel', and then click '…' in the Start Channel box. Here, select the universe number and enter the IP address and universe number for the controller.

Once you've done this for all your strips, it's almost time to start designing the sequence. However, before we do that, you'll need to create a group that they can be part of. Right-click in the Models box and select 'Add Empty Group'. You can then right-click on the models for each line and select 'Add to existing group' and then select the one you've just created. Once that is done, it's time to create a sequence.

> **Once you've added each controller,** it's time to organise the layout in the Layout tab

Click on the yellow paper icon to create a new sequence, and you'll get an option to create a musical sequence or an animation. They're both identical except that a musical sequence, unsurprisingly, includes music. You can upload a track and it will time the sequence to the music.



Select whichever one you like – they both work in very much the same way.

If you've selected a musical sequence, you'll now need to select the track you want to play, then you can click through to create the blank sequence.

The key difference between musical sequences and regular ones is that with musical ones you probably want the lights to flash in time with the beat, and that means working out where the beat is. To do this, you'll need to have installed the QM Vamp plugins.

Go to File > Sequence Settings > Timings. Click New and select an option from the list. We found that 'Tempo And Beat Tracker: Beats' worked quite well here, though bear in mind that these probably won't be perfect.

At this point, you have an empty sequence with timing blocks. You can start to add effects. These are the colourful squares along the top. Drag and drop these into your sequence wherever you like.

You can change how an effect looks by opening the Effect Settings box – click on the cog wheel icon.

Once you've got a few effects in place, it's time to see what it looks like. Click on the light bulb to connect to the LEDs (it will have an arrow in it when it's connected), then hit the play arrow and you should see your LEDs flickering on and off.

We've gone through quite a lot of setup to get here, but it's now that the real fun begins. There's still quite a bit of work to get a good-looking light show, but it's just a case of finding the effects you like and putting them in the order you like.

DMX, ARTNET, SACN, AND E1.31

You may see the protocol for sending LED data referred to as DMX, DMX512, sACN, or E1.31. While these have subtly different meanings, they're usually referring to the same thing. DMX512 is a system for controlling lighting systems such as those used on stages. The standard specifies both the format of the data sent and the physical network for wiring everything together. You can control LEDs like this and DMX512 lighting controllers are available, as are USB to DMX adapters. However, this is professional-level gear and usually priced as such. As hobbyists, it's usually easier and cheaper to send everything over a general IP network, such as Wi-Fi.

Architecture for Control Networks (ACN) is a range of protocols that run on top of IP networks for controlling things like lighting or sound. Streaming-ACN (sACN) is an extension to this to transport DMX512 data; sACN is also known as e1.31.

ArtNet is similar to sACN in that it's a way of transporting DMX data over an IP network (such as Wi-Fi). There are a few differences in how it handles network discovery and how data is routed to the right place, but for small networks, the difference is minimal.

Don't expect this to be a particularly quick process – especially if you've got a lot of LEDs – but you can see what's happening as you do it, and it's fun. All that's left after you've done this is to drive your friends and neighbours potty with Christmassy jingles with accompanying flashing lights.

Above Once everything's set up, just drag-and-drop your way to a light sequence

Part 01

Pico W IoT with Anvil: Mood lamp

Create a colour-changing RGB LED mood lamp controllable from a web app

MAKER

Phil King

Long-time contributor to *The MagPi*, Phil is a freelance writer and editor with a focus on technology.

@philkingeditor

You'll Need

- Raspberry Pi Pico W
- Anvil account (free tier) **anvil.works**
- RGB LED
- 3 × 330 Ω resistors
- 4 × male-male jumper wires

With built-in wireless connectivity, Pico W is ideal for creating Internet of Things (IoT) projects. To make it even simpler, Anvil has released a Pico W IoT toolkit to enable you to connect easily, and securely, to web apps you create on the Anvil platform. There's a drag-and-drop web UI builder, built-in database, user authentication, email integration, HTTP APIs, and more – and it all talks to your Pico W.

By using Anvil's special UF2 firmware image, you can code programs on Pico W in MicroPython as normal, with a few extra lines to connect to Anvil, then create a web app on the Anvil site which can send and receive data to/from your Pico W program. In this tutorial, we'll be showing you how to control the colour of an RGB LED using sliders in an Anvil web app.

01 Install the firmware

To be able to link your Pico W to the Anvil framework, you'll need to use a special firmware file. Go to **magpi.cc/anviluf2** and download the latest 'complete' UF2 file. (If you want to avoid overwriting any existing files on Pico W, use the 'firmware-only' version.)

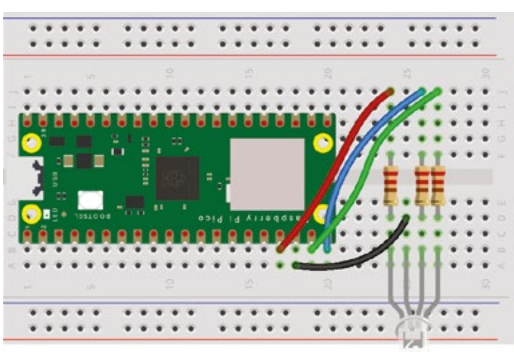


Figure 1: The wiring diagram for the mood lamp circuit using an RGB LED

As usual, hold Pico W's BOOTSEL button while connecting it to a computer via USB, then drag the UF2 file to the mounted 'RPI-RP2' volume. Once it's copied across, Pico W will automatically reboot and reappear as a volume called 'Pico W'.

02 Connect to WiFi

With Pico W connected to your computer, open up the Thonny IDE and make sure the Python interpreter (shown at the bottom right) is set to 'MicroPython (Raspberry Pi Pico)'. The **main.py** file on Pico W will run automatically, so you will need to stop it by pressing the Stop icon. Open the **boot.py** file and enter your wireless router's SSID (name) and password at the top.

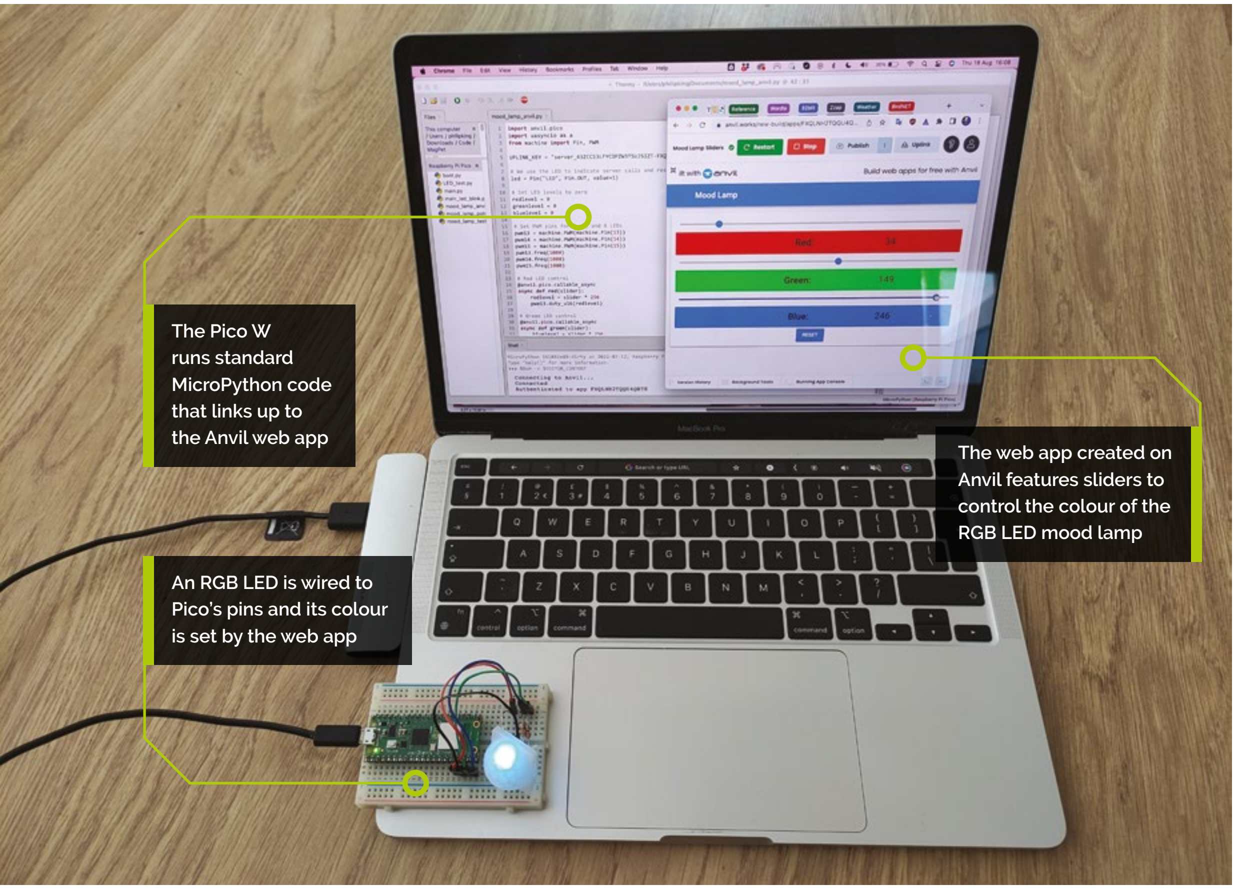
```
WIFI_SSID = "<put your network name here>"
WIFI_PASSWORD = "<put your wifi password here>"
```

Now when Pico W reboots, it'll automatically connect to your wireless network.

03 Wire up the circuit

Place your Pico W on one end of a breadboard, as in **Figure 1**. Add your RGB LED at the other end with each leg placed in a different row. It has four legs, the longest of which should be the ground connection – use a jumper wire to connect that to a GND pin on Pico W.

As usual with LEDs, you should use resistors to limit the maximum current to avoid possible LED burnout. We've used three 330 Ω resistors placed over the central dip of the breadboard to connect the LED's R (red), G (green), and B (blue) legs to jumper wires connected to GPIO pins 13, 14, and 15 respectively on Pico W.



The Pico W runs standard MicroPython code that links up to the Anvil web app

An RGB LED is wired to Pico's pins and its colour is set by the web app

The web app created on Anvil features sliders to control the colour of the RGB LED mood lamp

Note: Our RGB LED's legs were in the order R, GND, B, and G, but yours may differ.

04 Test the circuit

To make sure everything is connected correctly, we'll run a simple program on Pico W. In the Thonny IDE on your computer, make sure the Python interpreter (shown at the bottom right) is set to 'MicroPython (Raspberry Pi Pico)'.

Open a new file and add the code from the **RGB_test.py** listing (overleaf). Run it and the RGB LED should turn red, green, and blue in turn. If the colours are in the wrong order, you will need to swap over the relevant jumper wires.

05 Write the code

Now let's write the MicroPython code for our RGB LED 'mood lamp', as seen in the **mood_lamp_sliders.py** listing. The top line, `import anvil.pico`, enables Pico W to connect to Anvil's servers. The second, `import uasyncio as a`, sets up an asynchronous scheduler for running concurrent functions. We also import the Pin and PWM classes from the machine library so we can control our RGB LED from the GPIO pins.

Further down is a line starting `UPLINK_KEY =`. This is where you will later need to paste the uplink key for the web app you create on Anvil so your Pico W program can link to it. At the bottom of the program is a line to connect using it.

> "Connect easily, and securely, to web apps you create on the Anvil platform"

Next, we create three variables to set the levels of the red, green, and blue parts of the RGB LED. We then set up the PWM pins as in our test code.

06 Call the decorator

In our code, we have three functions: one each for controlling the red, green, and blue parts of the RGB LED. Before each, we add a 'decorator' and also add `async` at the start of the line defining the function; for instance: →

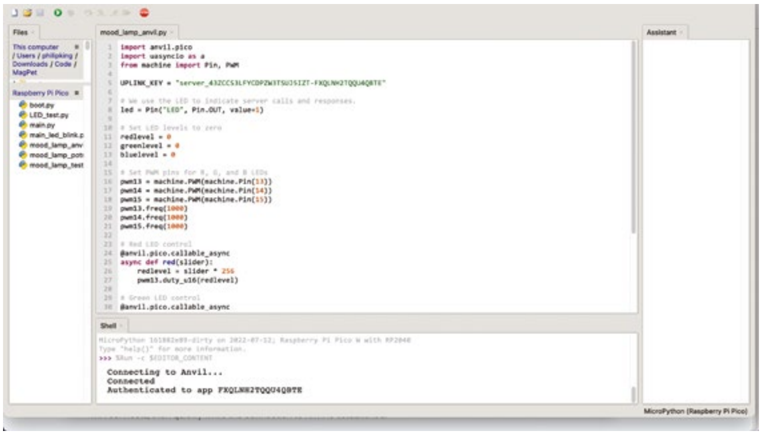
THE MAGPI

This tutorial is from in The MagPi, the official Raspberry Pi magazine. Each issue includes a huge variety of projects, tutorials, tips and tricks to help you get the most out of your Raspberry Pi. Find out more at **magpi.cc**

Top Tip

Autorun code

To make your Pico W code run automatically upon bootup, save it as **main.py**. If you already have a **main.py** file, you may want to save that under another name first.



When run, the Pico W MicroPython code will connect to the linked Anvil web app

```
@anvil.pico.callable_async
async def red(slider):
```

This lets our connected Anvil web app know that this function is available to call from the web.

We'll be sending a value to it, set by a slider in the web app, which will set the level of that colour. As we're using values from 0 to 255 for the slider, we multiply it by 256 to turn it into an unsigned 16-bit integer (0 to 65535) for the PWM level setting.

RGB_test.py

> Language: **MicroPython**

DOWNLOAD THE FULL CODE: magpi.cc/github

```
from machine import Pin, PWM
from utime import sleep

# Set PWM pins to control R, G, and B LEDs
pwm13 = machine.PWM(machine.Pin(13))
pwm14 = machine.PWM(machine.Pin(14))
pwm15 = machine.PWM(machine.Pin(15))
pwm13.freq(1000)
pwm14.freq(1000)
pwm15.freq(1000)

# Loop to light R, G, B LEDs in turn
while True:
    # Red
    pwm13.duty_u16(65535)
    sleep(1)
    pwm13.duty_u16(0)
    sleep(1)
    # Green
    pwm14.duty_u16(65535)
    sleep(1)
    pwm14.duty_u16(0)
    sleep(1)
    # Blue
    pwm15.duty_u16(65535)
    sleep(1)
    pwm15.duty_u16(0)
    sleep(1)
```

07 Design the web app

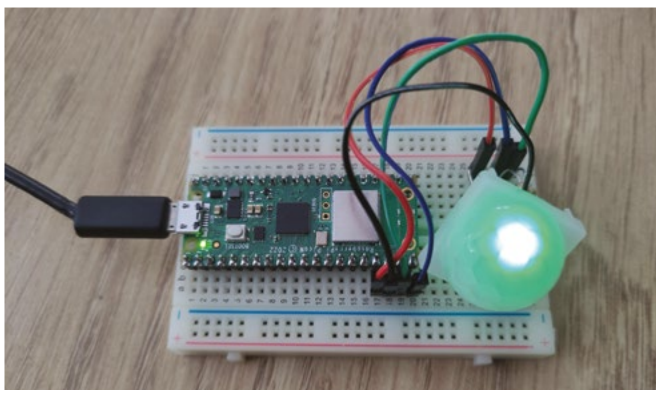
Go to **anvil.works** and sign up for a free account. We want to use slider controls for our lamp, which aren't available as a standard Anvil component, so we'll open up a demo app containing the Sliders library: go to **anvil.works/library/slider** and click 'Open in Anvil'.

The Sliders demo app only has two sliders and we want three, for red, green, and blue. From the right-hand Toolbox panel, drag a Slider custom component to just above the Reset button. Now select an existing 'Value:' element, press **CTRL/CMD + C** to copy it, and paste it under the new slider with **CTRL/CMD + V**. Copy and paste a '1' component to the right of it.

Click on each 'Value:' element in turn and change the text for it (in the right-hand Properties panel) to 'Red:', 'Green:', and 'Blue:' respectively. Set each '1' text element to '0'. Also, set the 'Blue:' element name to 'label_6, and the accompanying '0' element name to 'label_7'. For each slider, set the 'slider_max' value to 255.

08 Enable server uplink

Click the '+' button at the bottom left and select Uplink, then click Enable next to Server Uplink. This will generate a Server Uplink Key (ignore the client key) for the app which you



The cover from a PIR sensor makes a decent diffuser for the light from the RGB LED should paste into line 5 of your MicroPython code on Pico W so it can connect to it.

09 Adjust web code

We need to alter the web app code so the setting for each slider is sent to the red, green, or blue function in your Pico W program. Click on the Code tab and add the following lines (indented) to the `slider1_change` function:

```
    anvil.server.call_s("green",self.slider_1.
level)
    pass
```

Note that slider 1 is the middle one, which we've assigned to green. Similarly, for `slider2_change`, add the following two lines (indented):

```
    anvil.server.call_s("red",self.slider_2.
level)
    pass
```

Add a similar new function for slider 3:

```
  def slider_3_change(self, **event_args):
    self.label_7.text = self.slider_3.level
    anvil.server.call_s("blue",self.slider_3.
level)
    pass
```

Finally, add these lines to the `reset_btn_click` function in the web code:

```
    self.slider_3.level = 0
    self.label_7.text = 0
    anvil.server.call_s("red",0)
    anvil.server.call_s("green",0)
    anvil.server.call_s("blue",0)
    pass
```

10 Run both apps

We're now ready to roll. First, run your Pico W program in Thonny. You'll see messages in the Shell pane to show it connecting. Once it has, run your Anvil web app. You can now move the sliders to adjust the red, green, and blue components of your RGB LED to alter its shade.

To make it a little more effective, you may want to diffuse the LED's light. We simply placed the translucent plastic cover from a PIR sensor over it, which works fairly well. You now have a web-controlled mood lamp! Next time, we'll be reading Pico W sensor data in an Anvil web app. □

mood_lamp_sliders.py

> Language: **MicroPython**

DOWNLOAD THE FULL CODE:

magpi.cc/github

```
001. import anvil.pico
002. import uasyncio as a
003. from machine import Pin, PWM
004.
005. UPLINK_KEY = "<put your Uplink key here>"
006.
007. # We use the LED to indicate server calls and responses.
008. led = Pin("LED", Pin.OUT, value=1)
009.
010. # Set LED levels to zero
011. redlevel = 0
012. greenlevel = 0
013. bluelevel = 0
014.
015. # Set PWM pins for R, G, and B LEDs
016. pwm13 = machine.PWM(machine.Pin(13))
017. pwm14 = machine.PWM(machine.Pin(14))
018. pwm15 = machine.PWM(machine.Pin(15))
019. pwm13.freq(1000)
020. pwm14.freq(1000)
021. pwm15.freq(1000)
022.
023. # Red LED control
024. @anvil.pico.callable_async
025. async def red(slider):
026.     redlevel = slider * 256
027.     pwm13.duty_u16(redlevel)
028.
029. # Green LED control
030. @anvil.pico.callable_async
031. async def green(slider):
032.     bluelevel = slider * 256
033.     pwm14.duty_u16(bluelevel)
034.
035. # Blue LED control
036. @anvil.pico.callable_async
037. async def blue(slider):
038.     greenlevel = slider * 256
039.     pwm15.duty_u16(greenlevel)
040.
041. # Connect the Anvil Uplink. In MicroPython, this call
     will block forever.
042. anvil.pico.connect(UPLINK_KEY)
```

Cases and covers with a 3D printer

Make cases and covers by 'non-sew sewing'

Rob Miles

@robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com** and follow him on Twitter at **@robmiles**

It is a truth universally acknowledged that someone in possession of a bit of tech will want some cases and covers for it. If only to hide what you've bought from the rest of the household. But some things don't come with cases or covers, especially if they are things you have made from scratch. In this article, we'll explore how we can use 3D-printed components as stiffening and connecting elements for cases and covers.

NON-SEW SEWING

Sewing is great. It is one of the many things that the author can't do very well. Although he is the only person in the household who can operate the buttonhole maker on the sewing machine. But sewing thicker materials is hard work. And it involves sharp pointy needles. But the author is quite at home with nuts and bolts and 3D printing, so he prefers to use those skills instead of spending ages trying to thread a needle.

CUSTOM COVERS

Nothing keeps sunlight and dust off your precious devices like a cover. But they can be expensive to buy – even assuming you can find one that fits. So, why not make your own? You can buy leatherette material in a variety of colours at pleasingly low prices. And it can be bought in a width of 1.5 metres (just under 5 feet), so you can make covers for large as well as small devices. First, you cut to size, then cut out the corners, and with four seams down each corner, you

YOU'LL NEED

- **Leatherette (PU leather)** or cloth for the outer covers
- **Felt or thin cloth for the inner cover**
- **Spray adhesive**
- **A hole punch**
- **A sharp knife**
- **M3 bolts 4 mm long** – dome or countersunk
- **3D printer and filament**

Figure 1 You can use dome bolts for a rounder finish

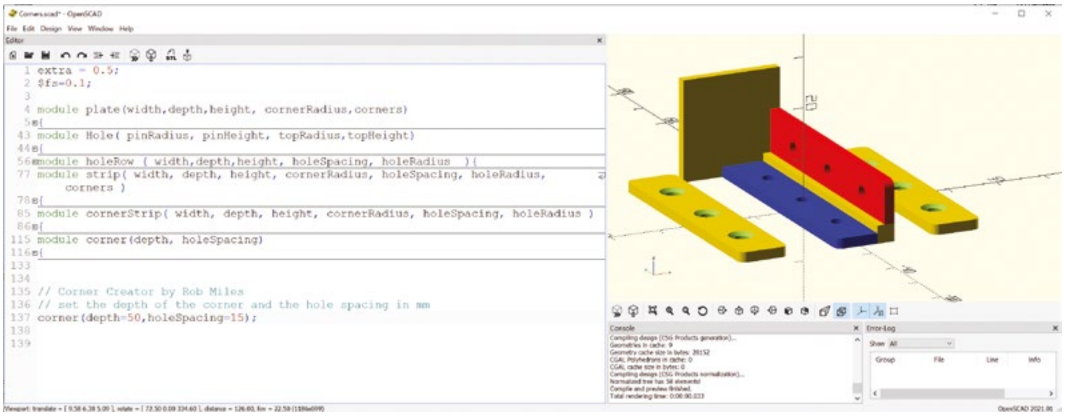


Figure 2
The red and blue elements were given colours to make it easier to see which is which when creating the design

Below
You can also use felt as a liner for the inside of the case, but the author rather liked the bulb material

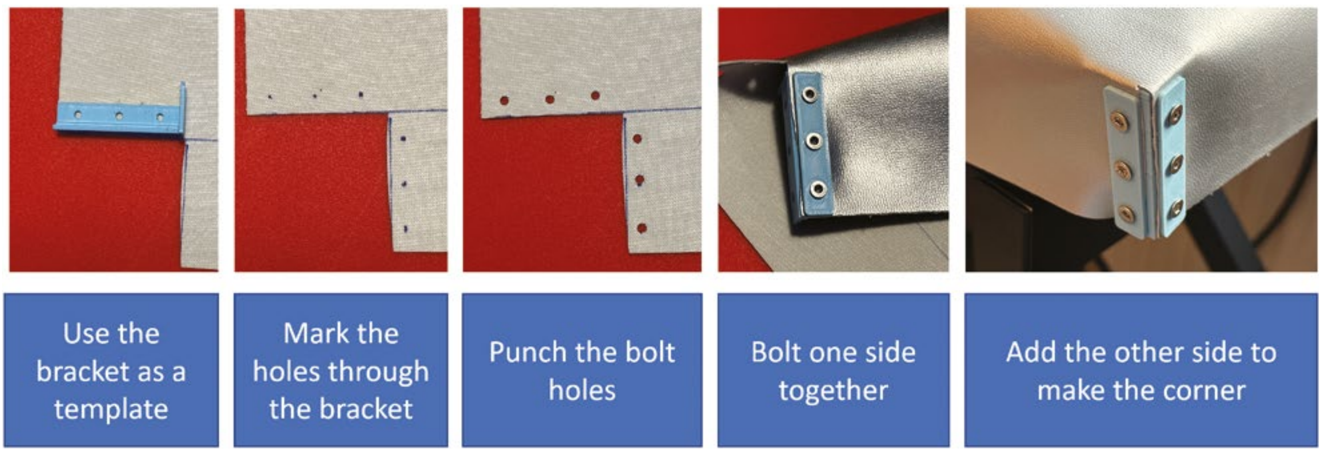
can make a snug-fitting cover. Stitching leatherette is not too difficult, and you can use a hole punch to make the holes if needed. But it can be hard to make your stitches consistent, and the corners that you make have no rigidity. So why not print some corners of your own and then clamp the material into them with bolts?

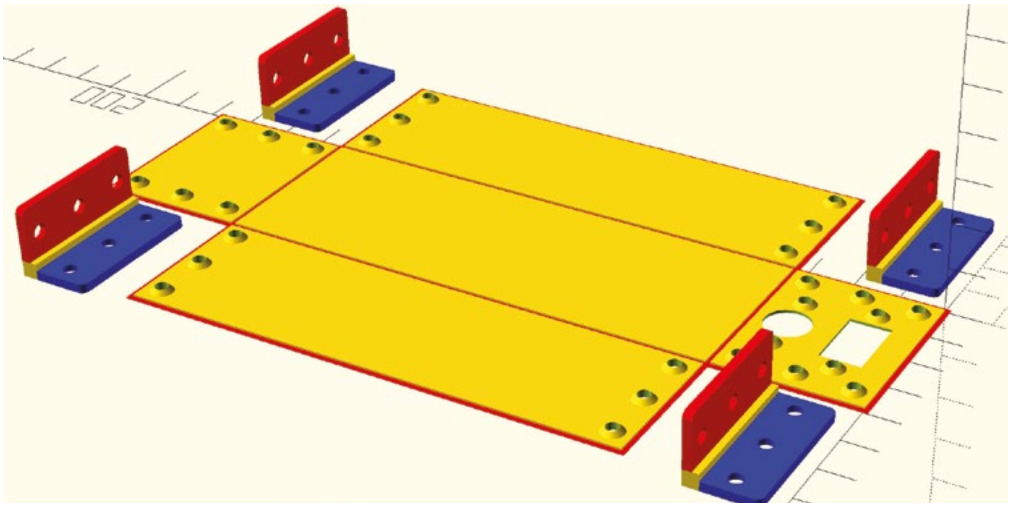
Figure 1 shows what we are making. The corner support is underneath the leatherette and the bolts screw straight into it, through the strip. The author has found that M3 bolts will screw into 3D-printed elements and grip tightly. A hole radius of about 1.45 mm seems to work well on an Ender-3 printer.

Figure 2 shows the designs for the clips for the edges of the covers. They were produced using OpenSCAD. You can find the designs on the GitHub site for this article: **hsmag.cc/covers-and-cases**. You can download the OpenSCAD application from: **openscad.org**. You can create corners of different lengths and hole spacings by changing the call to the corner function in the program:

```
corner(depth=50,holeSpacing=15);
```

Figure 3
The holes in the plates have a larger 'countersink' hole at one side. Make sure you screw the bolt into this larger hole so that the head is slightly recessed





The OpenSCAD program will make the designs for your corner and the brackets. These can be exported as an STL file to be sliced and printed.

Figure 3 shows how you add a corner to a cover. Bolt one side followed by the other. The bolts go through the cover plate, then the fabric, and finally into the bracket. If you want to make something super-strong, you can use longer bolts and put a nut on the end. Cover the corner fitting with insulating tape on the inside to stop the corner support from damaging the edges of the thing it is covering.

A CASE FOR TREATMENT

The party camera was featured in HackSpace issue 57. It takes pictures and sends them to a thermal printer so you can make a printed record of all your party fun. The top surface of the camera exposes a PIR sensor, a camera, an OLED display, and a trigger button. The bottom of the camera contains holes for a USB cable to power the device, and a tripod socket to support it. The author thought it might benefit from a case that just enclosed the bottom of the camera. The case would be lined with cloth and covered in leatherette. It was decided to use the same 'bolt-based' corners as were used for the cover just described, but this time, the corners would be fitted to the outside of the case and be bolted into 3D-printed stiffening panels that made up the case body. This makes for a nice steampunk aesthetic, although the author has probably used more bolts than he needed to. The corners would fit perfectly well with just two bolts.

Figure 4 shows the 3D-printed components for the party camera case. The stiffening material uses one-layer thick 'connecting' elements shown in red, and three-layer thick yellow panels to provide the stiffening. The bolt holes in the panels are surrounded by extra layers to allow them to grip the bolts from the corners. When the case is constructed, the sides are folded up and the corners bolted around the outside to hold them together.

The printable components are created by an OpenSCAD program that accepts the width, depth, and height values and then produces the design for the stiffening materials and the corner pieces. The program will also generate a lid for the case, but this is not needed for the party camera. Extra code was added to cut holes in the end panel for the USB connector and the tripod supports. The code for this is available at the GitHub site for this article: **hsmag.cc/covers-and-cases**.

PAPER TEMPLATES

The best way to get the shape of your cover (particularly if you are covering something which has an angled top) is to make a paper template first and then transfer the shape onto the material you are going to use for the finished cover. Paper is useful because you can always stick an extra sheet on if it turns out you've just cut the wrong part off.

Figure 4 This version of the case design has bolt supports for covers on the end panel for the USB and tripod holes. The cover wasn't added to the finished case because it didn't seem to need them

Figure 5 You can use a 1/8-inch drill bit to cut holes in the outer for the corner fixing bolts. Alternatively, you can cut the outer cover material to shape first and punch holes in it before sticking it onto the stiffener

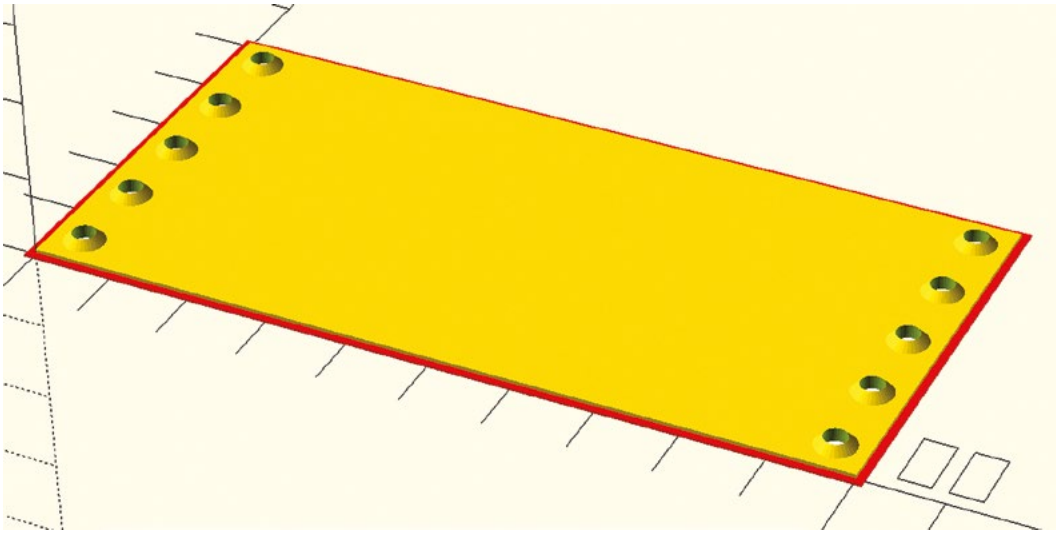


Figure 6 The 'holeSpacing' variable at the top of the program lets you set the spacing of the bolt fittings. The default value is 10 mm. The default margin around each panel is 1 mm. The 'margin' variable at the top of the program lets you change this

CASE ASSEMBLY

Figure 5 shows the sequence for making a case. The lining for the case was made from a scrap of spare fabric. You can also use felt as the lining material. The cloth and the cover were stuck onto the stiffener using extra strong spray-on flooring adhesive, which worked well. The outer cover was wrapped around the edges to finish them. The case was folded so that the raised parts of the bolt-hole reinforcements were on the inside. The outer cover and the liner material were stuck to the stiffener, and the excess material cut away once the glue had dried. The rectangular hole for the USB connector was cut with a sharp knife, and the inner lining cloth was folded back to cover the hole edges. The hole for the camera tripod was made with a hole punch and then opened up with a pen. The original plan was to print and fit covers over the two holes in the bottom of the case, but it turned out that with the corners fitted, it was not possible to add the covers. However, the case material leaves a clean edge when cut, and so the finished result was deemed tidy enough.

The case works well. It took a few attempts to get the sizes of all the components right. The outer cover and the inner lining added to the thickness of the edges so that the correct size of the corners could only be determined by printing a new iteration of the corner and testing it. The template took around an hour to print, with each corner taking around 20 minutes.

> "It would take a very long time to create panels **of similar strength just by printing them**"

PANELS FROM OPENSCAD

The OpenSCAD software for this project contains a function that will make panels of any size and add bolt holes and fittings on specified edges.

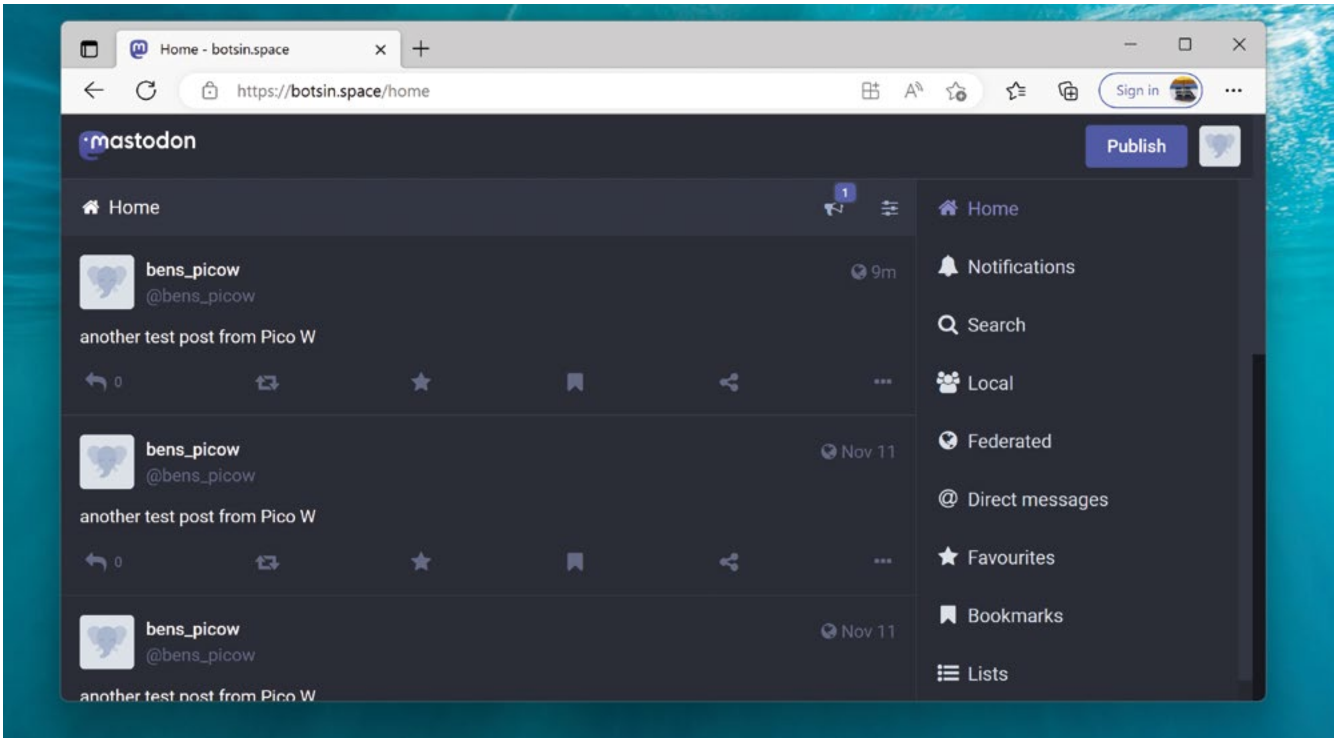
The panel shown in **Figure 6** was produced by the following statement:

```
makePanel(100,50,[1,1,0,0]);
```

The first two parameters specify the width and depth of the panel. The four-element array provided as the third parameter to the call of `makePanel` allows you to specify which sides of the panel should have bolt fittings in them. The first two elements in the array control the sides along the Y axis.

TAKING IT FURTHER

The author was very impressed by the strength and flexibility of the material that was produced by gluing thin 3D-printed layers between an inner lining and an outer covering. The use of single-layer thickness 'margins' to provide folds when assembling a structure also worked very well. It would take a very long time to create panels of similar strength just by printing them, and they would not have the flexibility of the ones that were produced here. It would be interesting to alternate strips of thin and thick stiffening elements to create panels that could be bent to form part of a curved case. By creating joining strips to link panels together, you could also create really large cases.



Pico W with CircuitPython

Getting online with a microcontroller

Ben Everard

@ben_everard

Ben's house is slowly being taken over by 3D printers. He plans to solve this by printing an extension, once he gets enough printers.

Above
The Mastodon social network has an API that's easy to connect to and use from embedded devices

The Raspberry Pi Pico W microcontroller has just got support for CircuitPython. Let's take a really quick look around.

For the most part, using Pico W is exactly the same as using Pico. The pinout is the same, and you'll have exactly the same features available. However, you also have added network connectivity.

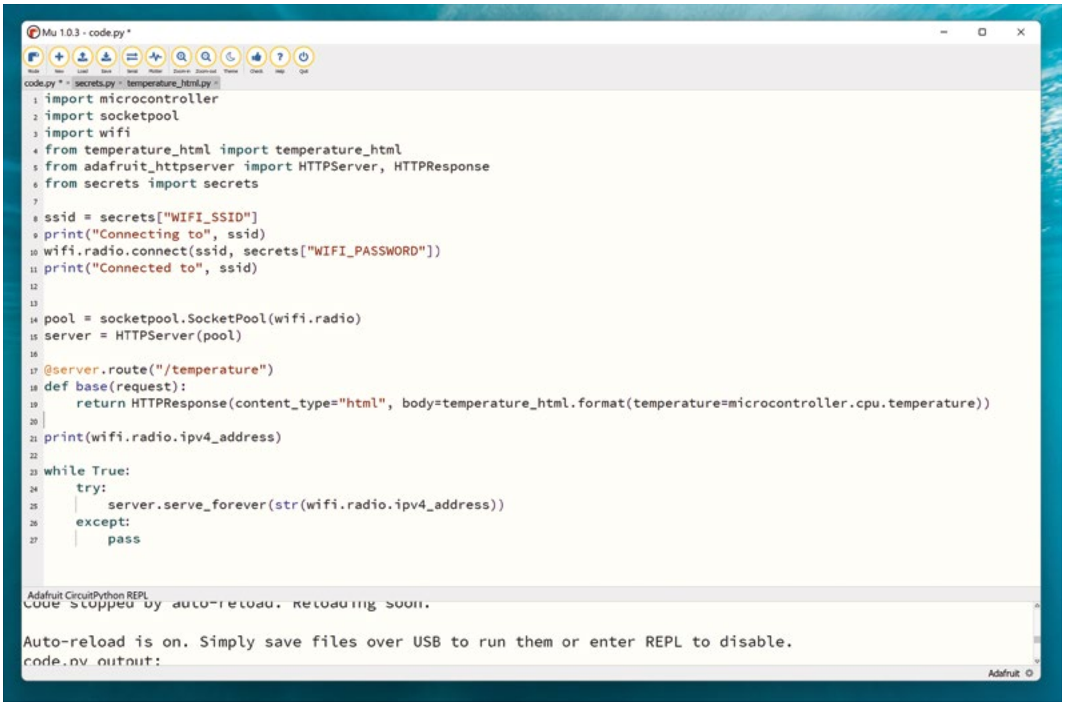
First, you need to get the software. You can download a UF2 file of CircuitPython for Pico W from **hsmag.cc/PicoWCircuitP**. At the time of writing, the latest version was 8.0.0-beta.4. There wasn't yet a stable version for Pico W. This should be fine for most cases, but obviously, a beta version might have bugs in it. The stable version should be available shortly.

In order to get that onto Pico W, you need to hold down the BOOTSEL button, then plug Pico W into your computer (unplug it first, then plug it in). You should see a new removable drive appear, and you can drag and drop the UF2 file onto this. It should disappear and a new CircuitPython drive should pop up.

If you have not used CircuitPython before, then there is a good 'getting started' guide at **hsmag.cc/AdafruitCircuitP**.

For our first experiment, let's try interacting with the social media network du jour, Mastodon. You'll need a Mastodon account for this. It can be a personal account, or you can set one up just for your bot. In fact, there's a Mastodon instance set up specifically for bots – **botsin.space**. Be sure to read the rules so you can properly write your request if you want an account on that instance. If you're looking to set up a personal account, you'll need to pick an instance to join. Head to **joinmastodon.org** to look at the options. Don't worry too much, as you will be able to move your account to a different instance later if you want, and you can interact with people on any instance.

Once you've got an account, you can get an API key. In your edit profile page, go to Development, then click on New Application and give it a name, then



Left
You can build an HTTP server in under 30 lines of CircuitPython, so it's an easy way of adding an interface to your project

press Submit. You'll get redirected to the application list page. Click on the name of your new app, and this should take you to the details page where you'll see the access token for this app. You'll need this shortly.

Posts on Mastodon are short bits of text. Exactly how short depends on your instance, but 500 characters is pretty common. They can also have images or video attached to them, but we won't worry about that at the moment.

Let's look at the code to create a post on Mastodon.

```
import microcontroller
import socketpool
import wifi
import adafruit_requests as requests
import ssl
from secrets import secrets

ssid = secrets["WIFI_SSID"]
print("Connecting to", ssid)
wifi.radio.connect(ssid, secrets["WIFI_PASSWORD"])
print("Connected to", ssid)


pool = socketpool.SocketPool(wifi.radio)
requests = requests.Session(pool, ssl.create_
default_context())

response = requests.post("https://botsin.space/
api/v1/statuses",
                    headers={"Authorization" :
secrets["mastodon_access_token"]},
                    data={"status":"another
test post from Pico W"})
```

As you can see, this uses the Adafruit requests module that can be used much like Python 3's requests module. You just need to make sure that you copy it over the **adafruit_requests** folder from the CircuitPython Library Bundle (which you can download from **circuitpython.org**).

You will also need to create a **secrets.py** file. This will contain all the things that you don't want to be public, including your Wi-Fi password and Mastodon key. →

GETTING DATA

In the first example here, we used a POST request to send data to an external server. In the second example, we set up a server and got data from it. You might be wondering if it's possible to combine the two and use POST requests to send data to Pico W. The answer is: yes, you can.

We don't have space to go through it all here, but Liz Clark has written a guide to this for Adafruit – you can see it at **hsmag.cc/PicoWHTTPServer**.

ESPHOME

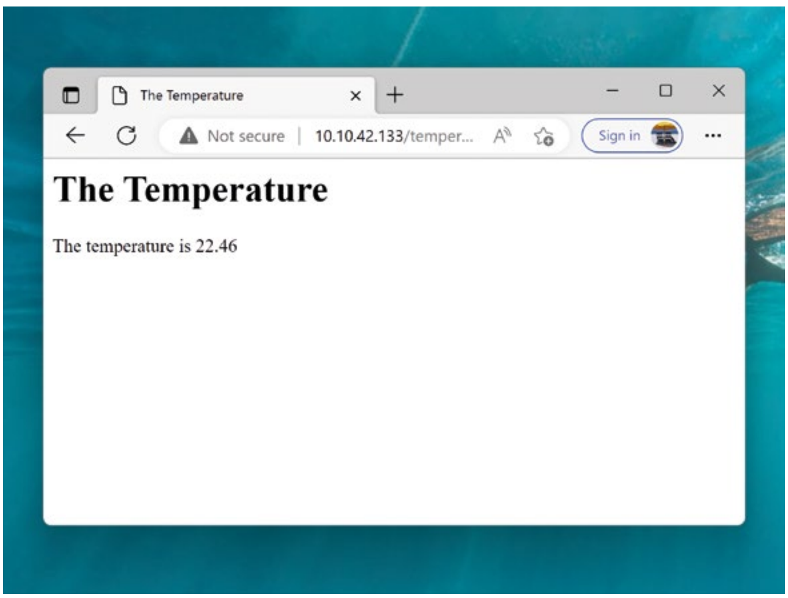
ESPHome is a platform for building and interconnecting DIY Internet of Things devices. As the name suggests, it was originally built for ESP8266- and ESP32-based devices. However, there's now support for Pico W. This gives you a really easy low-code way of linking up sensors, actuators, and other 'thing' devices. You can create configurations in the YAML markup-language and deploy these to devices. Using this, you can very quickly make a network of interconnected things using commonly available maker components and control them from the Home Assistant smart home software.

The format should be:

```
secrets = {"WIFI_SSID" : "<your ssid>",
        "WIFI_PASSWORD" : "<your password>",
     "mastodon_api_key" : "Bearer <your token>"
}
```

In most of these cases, you just need to slot in the appropriate bit of secret information. The Mastodon token is a bit different just because you need to include the word 'Bearer' before the token.

Once you've set this up, you can save the above code as **code.py** on your Pico W, and it will connect to your Wi-Fi network, post the message to Mastodon, and then finish. This is probably a little dull by itself, but you can use it to log information, send messages, or just generally to scream into the void.



Below
OK, this isn't the best-looking website, but it gets the information across

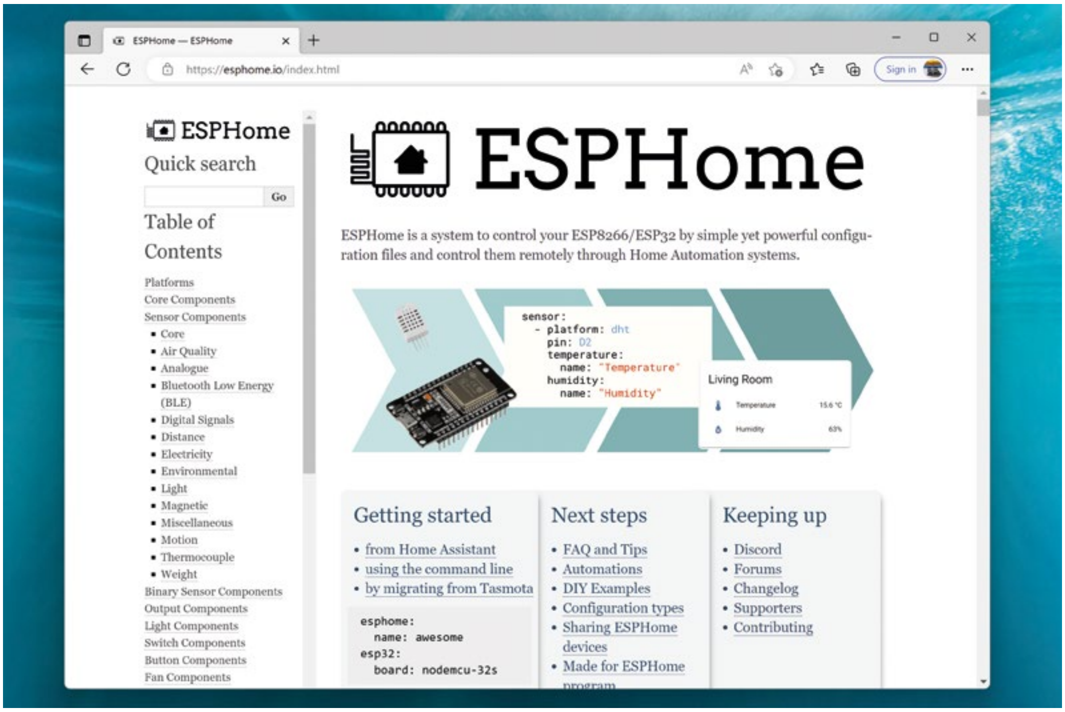
AT YOUR SERVICE

The second thing that we're going to look at is setting up a simple HTTP server. This will let you connect to Pico W and read information about what's going on.

```
import microcontroller
import socketpool
import wifi
from temperature_html import temperature_html
from adafruit_httpserver import HTTPServer,
HTTPResponse
from secrets import secrets

ssid = secrets["WIFI_SSID"]
print("Connecting to", ssid)
wifi.radio.connect(ssid, secrets["WIFI_
PASSWORD"])
print("Connected to", ssid)
print(wifi.radio.ipv4_address)

pool = socketpool.SocketPool(wifi.radio)
server = HTTPServer(pool)

@server.route("/temperature")
def base(request):
    return HTTPResponse(content_type="html",
body=temperature_html.
format(temperature=microcontroller.cpu.
temperature))

while True:
    try:
        server.serve_forever(str(wifi.radio.
ipv4_address))
    except:
        pass
```

There's a slightly unusual bit of syntax in the line `@server.route("/temperature")`. This is called a decorator, and you put them before methods. They're not used that much in CircuitPython, but are a bit more common in regular Python. Essentially, they are a way of wrapping a method up in another bit of code. We won't go into everything you can do with decorators, but in this case we're using the `server.route` method to register the method `base` with the server at the location **/temperature**. This way, anyone who visits **http://<server_IP>/temperature** will see the output of this function.

We use the `HTTPResponse` method (that we've imported from `HTTPServer`) to create this response. This method takes care of most things; we just have to tell it that we can't return HTML and that the



Left The new support for Pico W in ESPHome gives access to a huge range of IoT hardware with minimal coding

body of the response should be a formatted version of **temperature_html**.

When working with web pages, we're going to need to manipulate quite a bit of text. This isn't a huge problem from a processing point of view, but we want to be a bit careful of how we manage our files to ensure that they're not too messy. There are a few ways of doing this, but we've stored the HTML in a string in another file and so are importing that string from the file. We're using Python's built-in string formatting tools to insert data into this HTML string. In its simplest form, this lets us put **{0}** in the string somewhere, and then the **format** method will replace this with the first parameter to format. If you want to insert more bits of text, you can add **{1}**, **{2}**, etc. to the string and use extra parameters to format to inset these into the text. The **temperature_html.py** file is:

```
temperature_html = """
<html><head><title>Temperature</title></head>
<body><h1>Temperature</h1>
<p>The temperature is: {0} degrees C</p>
</body>
</html>
"""
```

HTML is a markup language for creating web pages. We're not going to go into it too deeply here, but very simply, text is encased in tags with angled brackets. A **/** tag closes the brackets. There's a **<head>** section with things like the title, and the body which is the

> "You can use it to build up complex interfaces and **easily get data out of your build without having to physically plug in a cable**"

main text of the page (and not to be confused with the body of the HTTP request which includes the whole of the HTML).

<h1> is heading level 1 (i.e. the largest heading); **<p>** is a paragraph.

With all that in place, we just need to kick off our server with **server.serve_forever(str(wifi.radio.ipv4_address))**. We've wrapped this up in a try-except block in an infinite loop. This is in case there are any problems while it's running. HTTP servers don't have to be particularly complex, but there's a lot of things that can behave strangely in there. The network could lose packets. The client requesting the data might send a poorly formed request, or there could be a bug in our code. This will just attempt to restart the server should anything go awry. It might not last forever, but it will make it a bit more resilient.

There we have it. A simple HTTP server for interacting with a Pico W. You can use it to build up complex interfaces and easily get data out of your build without having to physically plug in a cable – which can make the physical setup much easier. We've looked at the most basic use case here, but it can get as complex as you like. □

YOUR GIFT GUIDE FOR 2023

THE BEST NEW PRODUCTS YOU CAN BUY **FOR CHRISTMAS AND INTO 2023**

£420

G-MASTER GB3467WQSU-B1

Immerse yourself in the game with the curved GB3467WQSU Red Eagle with FreeSync Premium.

Inspired by the curve of the human eye, the 1500R curved VA panel with 165Hz refresh rate, 0.4ms MPRT, and 3440×1440 resolution guarantees superb image quality and a comfortable and very realistic viewing experience.

The height-adjustable stand ensures total flexibility for your perfect screen position. Customise the screen settings using the predefined and custom gaming modes along with the Black Tuner function to give you total control over the dark scenes and make sure details are always clearly visible.

magpi.cc/gmaster

FLIRC SKIP 1S

The Skip 1s Universal Remote Control makes controlling your entertainment center a breeze. With support for 3 different activities that can be activated with the click of a button, a universe of device combinations is at your fingertips. Leave the simplicity to us, and we'll leave the creativity to you.

magpi.cc/skip1s

CYBERPOWER PC

A custom gaming PC from Cyberpower UK is a perfect treat this Christmas season. With hundreds of parts to choose from – including NVIDIA's GeForce RTX 4080 GPUs – its cutting-edge configurator tool lets you build your dream spec from scratch and finish it with a custom engraved side panel.

Order your gaming PC by 12 December, or pick up a pre-built one by 21 December, for guaranteed Christmas delivery, free UK delivery, and an extended return warranty for all holiday shoppers.

cyberpowersystem.co.uk

AOC Q27G2E/BK

The AOC Q27G2E/BK offers a 27-inch VA panel with QHD resolution, ShadowControl and super contrast ratio of 3000:1. Enjoy the most responsive gameplay and fastest battles with its stutter-free Adaptive Sync, 155Hz refresh rate, 1ms MPRT, and low input lag to decrease motion blur and input-output delay. Gaming has never been so fun and intense.

magpi.cc/Q27G2E

£219

£229

AOC C27G2ZU/BK

Unleash your potential with this 240MHz, 27-inch display. Its 240Hz refresh rate, 0.5ms response time, and low input lag enable the AOC C27G2ZU to provide a perfectly smooth performance. With its curved design, height adjustment, and swivel ability, the monitor can be adjusted to individual needs. It comes with FreeSync Premium and G-Sync compatibility.

The C27G2ZU/BK is on a Black Friday promotion at £189, which may continue depending on stock, so pick one up early if you find it on sale.

magpi.cc/C27G2ZU

MINI PUPPER

Mini Pupper is an AI-powered, smart, quadruped robot designed for education. Mini Pupper makes robotics easier for schools, home-school families, enthusiasts, and more.

Utilising ROS (Robot Operating System) and with support for OpenCV and OpenCV's official OAK-D-Lite 3D camera module, this open-source robot is powerful and super-expandable.

mangdang.store

FROM £335

* Prices may vary

Radio communication

Sending messages through the air using magic (and electromagnetism)

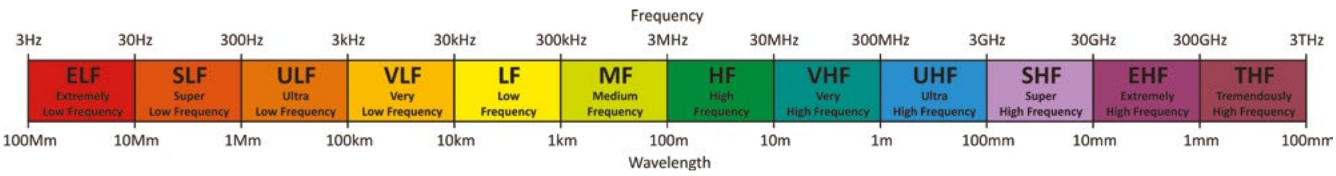
Mike Bedford

Despite loving all things digital, Mike admits to being a bit of a Luddite, vinyl records and all.

With streaming services gradually replacing broadcast radio and TV, it might seem that radio is yesterday's technology, but the reality is quite different. Although the word radio isn't usually used in the same breath, so many of the technologies on which today's world relies depend on radio communication. Included here are mobile/cellular phones, WiFi access points and the numerous devices that connect to them, cordless home phones, car keys, and remote control units for garage doors, central heating and more. What's more, it's quite feasible for you to incorporate radio remote control in your projects, and achieve a greater range than you'd manage with infrared. Here, we introduce radio technology, explain how you can use it legally, and suggest a couple of practical ways you can learn about the technology using a low-cost transmitter and receiver. You'll then be perfectly placed to use these devices together with a Raspberry Pi or Arduino single-board computer, as we'll go on to explain.

INTRODUCING RADIO

Radio is a form of electromagnetic energy, as are infrared, visible light, ultraviolet, X-rays, and gamma rays. What sets these different areas of the electromagnetic spectrum apart is their frequency.



Of all these, radio waves have the lowest frequency, ranging from just a few Hertz through to 300 GHz. Because this is quite a large range, radio is split into several regions, each spanning an order of magnitude. Of particular interest are the following: LF (low frequency) includes the long-wave band, which is still used for analogue broadcast radio; MF (medium frequency) includes the medium-wave band, which is also used for analogue broadcast radio; HF (high frequency) includes various short-wave bands, which are used for long-range, international broadcast radio; VHF (very high frequency) includes FM and DAB (digital audio broadcasting) radio; UHF (ultra high frequency) carries terrestrial TV broadcasts, mobile phones, and WiFi, and is also where the transmitter and receiver modules we're investigating here operate; and SHF (super high frequency) also carries mobile phones and WiFi.

We show each of these areas of the radio spectrum in **Figure 1**, plus those regions which are used for more specialist applications. As well as frequencies, we also show wavelengths which run in the opposite direction, so as the frequency increases, the wavelength decreases. Although we now refer to radio waves by their frequency, at one time the wavelength was used instead. Knowing about the wavelength is still important, though, since it dictates the size of the antenna.

LEGAL CONSIDERATIONS

From the very early days of radio, governments realised that allowing a free-for-all would result in chaos, so legislation was introduced, regulating how radio could be used. Until just a couple of decades ago, therefore, a licence was required to operate any radio transmitter and, as a result, very few people used radio transmitters.

Today, we all use devices that transmit radio waves, for example, mobile phones and WiFi-enabled equipment. This is allowed because they operate in parts of the radio spectrum that have been set aside for licence-free use, but that still doesn't mean there are no regulations. Specifically, the equipment should be low-powered, and it must have been certified by its manufacturer, as adhering to the relevant standard. This means that you can't build your own transmitter for use in a licence-free band, but there are ways of getting around that. Transmitter modules can be certified, allowing them to be incorporated into equipment, and we'll be taking advantage of this proviso later when we see how you can build radio control into your own projects.

> "We all use devices that transmit radio waves, **for example, mobile phones and WiFi-enabled equipment**"

Higher-powered equipment, as used by businesses, for example taxi companies, can only be used with a licence, but the equipment still has to be appropriately certified. Of particular interest to electronics enthusiasts, though, is the amateur radio licence. They're only granted to people who have passed a test about radio theory, but there are several benefits. Radio amateurs can use high power on a wide range of frequencies, allowing them to →

Figure 1
The radio spectrum covers twelve orders of magnitude. Here, we're looking particularly at 433 MHz radio, which is in the UHF region

Left
Radio might conjure up nostalgic images, especially if we use the word 'wireless', but while its use for broadcasting is in decline, it remains key to many of today's most important technologies

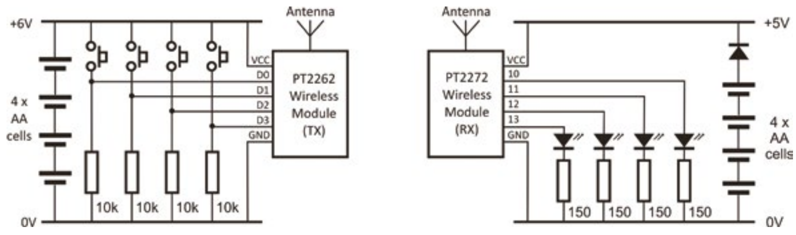
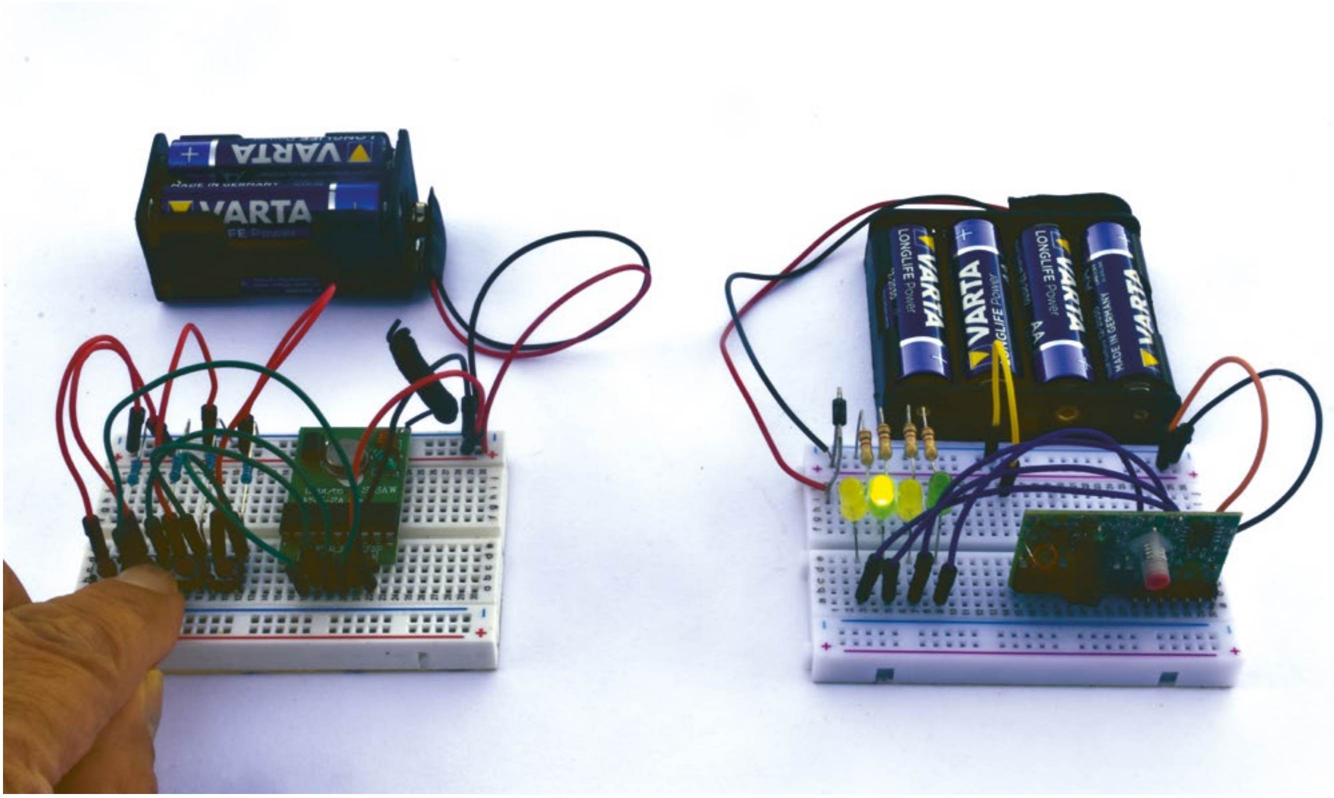


Figure 2
Using 433 MHz transmitter and receiving modules with on-board encoders and decoders couldn't be much simpler



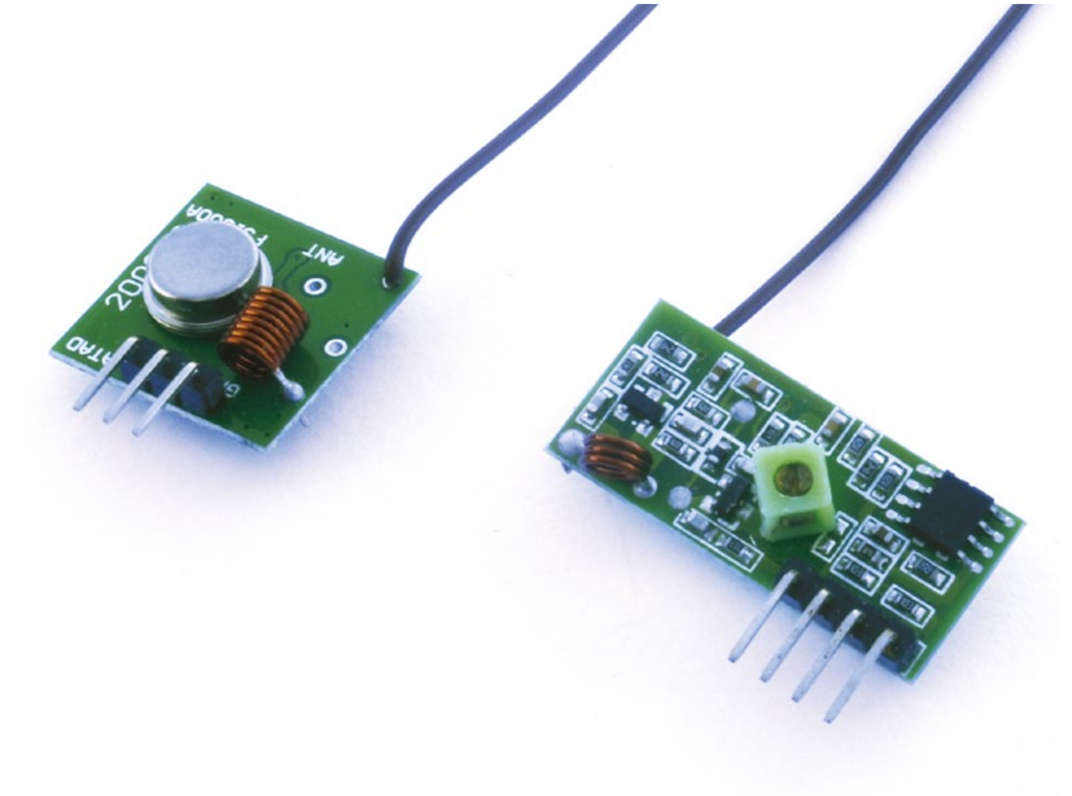
Above To get a feel for transmitter and receiver modules with built-in encoders and decoders, we suggest you add four push-buttons to the transmitter and four LEDs to the receiver

communicate worldwide. And of particular interest, they can experiment by building their own equipment.

433 MHZ MODULES

One of the most common licence-free bands for remote control applications is the 433 MHz band, which is available in many countries, and is our main theme here. Transmitter and receiver modules are widely available and very cheap. 433.92 MHz modules are by far the most common, but other frequencies within the 433 MHz band are available, so ideally, buy the transmitter and receiver modules as a pair, but if not, make sure they operate on the same frequency. The transmitter has a data-in pin, and applying a positive signal to that pin causes it to transmit. Conversely, the receiver has a data-out pin, which provides a positive signal when a radio signal is detected. This might suggest that you could add a push-button to the transmitter and an LED to the receiver and you'd be able to illuminate the LED remotely. In practice, it's not that simple. The snag is that, like most radio receivers, the receiver module has an automatic gain control (AGC) that increases the amplification as the received radio signal gets weaker. The upshot of this is that pretty much any signal, however weak, even including noise, will get amplified so much that the module generates a signal on its data-out pin. However, the AGC doesn't act immediately, but is delayed, so it's still possible to differentiate between a radio signal (a digital 1) and no radio signal (a digital 0), so long as the signal is flipping between 1s and 0s sufficiently frequently.

To do that, the data has to be transmitted at a speed that's faster than the AGC's delay, but that alone wouldn't be enough because long strings of 1s or 0s would still cause problems. So the data is Manchester encoded, which means that 1s and 0s are represented either by a high-level signal followed by a low-level or vice versa. As a result, there'll be a transition in the transmitted signal for every bit. Commonly, this is achieved using an HT12E encoder chip on the transmitter and an HT12D decoder chip on the receiver. You can buy these and add them to the transmitter and receiver modules, although an alternative solution is to buy transmitter and receiver modules with the encoding and decoding chips already on-board. They're not as common as dumb modules, but you should be able to find modules called something like PT2262 Wireless Modules (transmitter) and PT2272 Wireless Modules (receiver) for less than £10 for the pair, for example, from **hobbycomponents.com**. This company offers both 315 MHz and 433 MHz variants but, for use in the UK, get 433 MHZ modules, because the 315 MHz versions can't be used legally. These modules actually use the PT2262 encoder chip and the PT2272 decoder chip, which appear to be identical to the HT12E and HT12D, respectively, except that



Left 433 MHz modules will work over a few metres without antennas, but you need external antennas – and straight ones work better than coiled ones – to achieve a good range

the HT12D has latched outputs, while the PT2272 is available with either latched or momentary acting outputs, the version in these modules being the latter. A useful feature of all these chips is that the encoder has four data inputs, and the decoder has four data outputs, so you can control four different devices. To start, to get a feel for these modules, you'll probably use four push-buttons on the transmitter and four LEDs on the receiver. Here, we show how the modules should be wired up to do that, and you can also see how simple it is to patch them up using a pair of breadboards. Note that, if the modules we used are typical, the data inputs on the transmitter are labelled with the bit numbers (D0, D1, D2, and D3), while the data outputs on the receiver are labelled with the pin numbers on the PT2272 chip (10, 11, 12, and 13), but D0 is on pin 10, D1 is on pin 11, D2 is on pin 12, and D3 is on pin 13.

Having patched up the modules, you can see what sort of range you can achieve. With just a 6 V supply on the transmitter, and the helical antennas that are fitted to the modules when you buy them, we only managed to communicate over four metres. However – unlike the receiver module which should not be used with more than a 5 V supply, so you should use a diode, as shown in our schematic (**Figure 2**), to reduce the voltage if you use four AA cells as your supply – the transmitter can be used with up to 12 V. Increasing the supply from

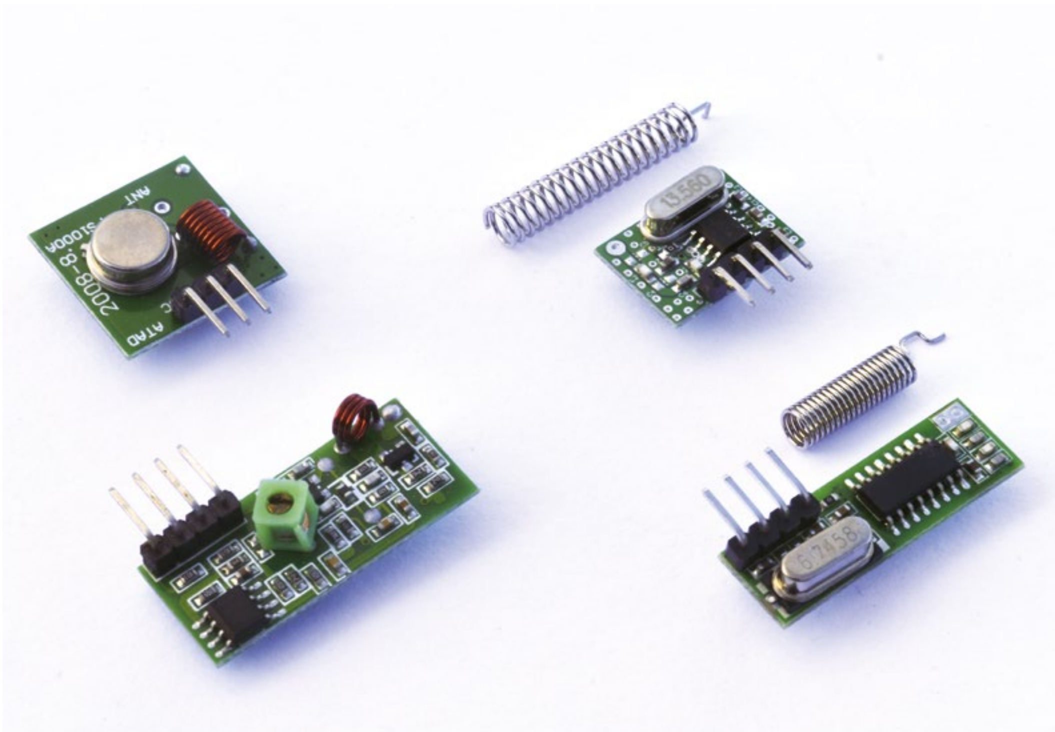
> " With just a 6 V supply on the transmitter … **we only managed to communicate over four metres** "

6 V to 12 V will produce four times the transmitted power, which should double the range. The range could also be improved by changing the original helical antennas with straight antennas made from stiff insulated wire. They should be a quarter of a wavelength long, which is 167 mm at 433.93 MHz.

The other thing to try is using a superheterodyne receiver instead of the inferior super-regenerative design. The latter is the more common, and if the type isn't mentioned, it's probably super-regenerative. In the main, you'll find that the superheterodyne modules are shipped directly from China and have long delivery times, but we did discover **247geek.co.uk**, which ships modules from the UK.

ENCODING AND DECODING

Using the PT2262 and PT2272 modules is very convenient, but you'll probably not learn a lot about the encoding and decoding process, as you would if you were to use unencoded 433 MHz →



Right Superheterodyne receivers (bottom right – top right is the paired transmitter), aren't as common as the super-regenerative receiver (bottom left – top left is the paired transmitter), but they offer superior performance

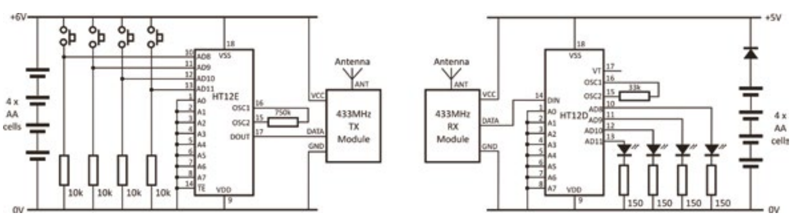
transmitter and receiver modules with separate encoder and decoder chips. So, if you're interested in the behind-the-scenes details, take a look at the schematic shown in **Figure 3**, which is a commonly shown circuit. This uses the HT12E and HT12D chips, which are more commonly available than the PT2262 and PT2272.

Most of it should make sense with little in the way of explanation, but the chips have various features that aren't used in the PT2262 and PT2272 modules we looked at earlier, so let's take a look at those chips. First of all, the HT12E and HT12D both have inputs labelled A0 to A7, which are all shown as connected to 0 V. These are actually address inputs, and if facilities are provided on the transmitter to change the address, it would be possible to selectively transmit to up to 256 different receivers, each of which would need to have a different 8-bit address. We also show a pin labelled $\overline{\text{TE}}$ connected to 0 V on the HT12E encoder. This means that the chip will generate a continuous stream of data so the transmitter module will transmit continuously. However, by keeping that pin at +5 V, and only applying 0 V when it's necessary to transmit, power consumption will be reduced, which is important for battery-powered equipment. You could do that, for example, by generating a low-level signal whenever one of the four push-buttons is pressed. Moving to the HT12D decoder, you'll notice that there's a pin labelled VT which is unconnected. This produces a signal whenever a valid transmission is received and could be connected to an LED, for example.

If your project requires 433 MHz modules to be connected to a Raspberry Pi or an Arduino, the simplest solution would be to use modules with on-board encoders and decoders. However, if you want to reduce the chip count, there's an alternative to using encoder and decoder chips, that becomes available if your project uses a single-board computer. That solution is to implement the functionality of the encoder and decoder chips in software. And the good news is that the necessary functionality is available in various libraries and open-source software projects, and guidance on using these resources appears to be widely available.

And finally, and if you get bored with 433 MHz modules, and have a project that needs greater data throughput, you might be interested to know that there are also 2.4 GHz transceiver modules – that is a transmitter and receiver combined – which are intended exclusively for use with single-board computers.

Figure 3 This standard circuit shows how 433 MHz transmitter and receiver modules can be used with the HT12E encoder and HT12D decoder chips. With an SBC, though, the encoding and decoding could be done in software



HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

PG 94

MAKE YOUR OWN UNO

Learn to solder with Arduino

PG 88

BEST OF BREED

The bits you need to build a home arcade

PG 96

CROWDFUNDING FISHES

Paper lanterns to make at home

ONLY THE BEST

Arcade controllers, parts, and accessories

Build your own arcade

By **Marc de Vinck** @devinck

Custom arcade builds have been a popular project for many DIY electronic enthusiasts. Personally, I have built over 15 different arcade machines over the years. Yes, I counted! From full-size cabinets to tabletop versions, and even a few that can fit in your pocket. It's a fun project for beginners and advanced builders alike. You can easily build one that does not require any soldering from a complete kit, or you can go all out and design a 100% custom build, including designing your own PCBs. It's one of those topics that has a robust marketplace and community, making it easy to jump in, no matter what your skill level.

In this round-up, I'll be looking at a wide variety of kits and components, all focused on building your own arcade system. To make your build easier, some are complete kits – one even includes the elusive Raspberry Pi 4 – while there are also some basic components and break-out boards. Something for everyone! And, just like the custom keyboard-building scene that has taken over the DIY community lately, there are almost limitless accessories that allow you to get a truly unique build.

Picade vs Adafruit LED Arcade Button 1×4

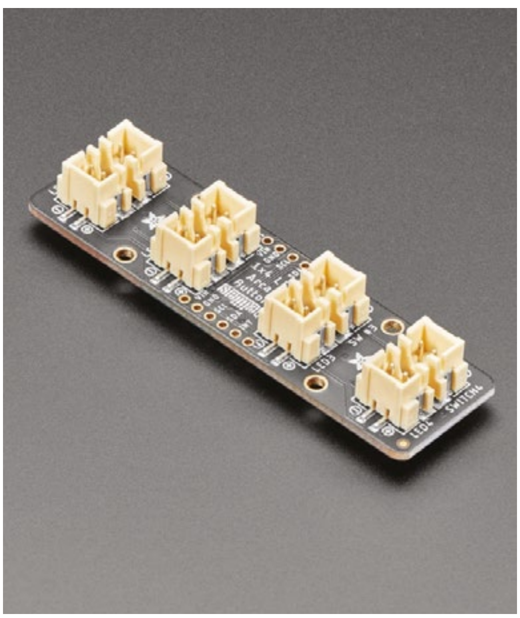
PIMORONI $191 - $324 | pimoroni.com

ADAFRUIT $9.95 | adafruit.com

If you want to get into retro gaming and the idea of putting everything together and having a polished and complete system seems daunting, then the Picade from Pimoroni might be a good place to start. If you opt for the basic kit, you'll have to source a few parts like a Raspberry Pi, power supply, and SD card. But, you won't have to design, cut, and figure out how to build the enclosure. And, in most instances when building a retro arcade, the enclosure is the hardest part! If you go for the new 10" deluxe kit, you'll get everything needed to build the complete desktop arcade machine, even the difficult-to-source Raspberry Pi! And as you all know, finding a Raspberry Pi is not easy nowadays!

Left
Arcade machines don't have to be huge

Below
Lots of buttons with only a few GPIO pins



Save pins and time with the Adafruit LED Arcade Button 1×4. This handy little board may look simple at first, but it has a few interesting surprises hidden on the back of the PCB. This board allows you to add multiple illuminated arcade buttons to your next microcontroller project thanks to the on-board microcontroller programmed with seesaw firmware.

Simply connect the buttons via the JST XH sockets, and the board will not only supply 5V power to the button LEDs, but it will also do all the controlling via I2C. This will save you from having to use a dedicated pin for each button or dealing with stepping up 3.3-volt power. You can also connect several of these boards together, 16 boards in total, and you won't need any more pins on your microcontroller. This board really does simplify adding multiple buttons to a custom microcontroller-based project. Head over to the website for additional technical details about how to use the Adafruit LED Arcade Button 1×4. →

VERDICT

Picade

A kit that includes a Raspberry Pi 4!

10/10

Adafruit LED Arcade Button 1×4

A simple way to connect arcade buttons.

9/10

BEST OF BREED

SparkFun micro:arcade kit for micro:bit v2.0

SPARKFUN ◆ **$52.50** | **sparkfun.com**

Left ◆
From microcontroller to games machine

The folks over at SparkFun love gaming. So much so, that they even came up with a kit for the micro:bit. The kit features a custom-designed carrier board that includes a classic D-pad style layout of buttons and two additional A/B buttons. The kit also includes four classic arcade buttons, a joystick, battery pack, and a wiring harness to hook it all up.

You might think this is a bit overkill for a micro:bit, but they have done a good job of giving examples of simple games that can be made on the LED matrix of the micro:bit. The kit does not require any soldering, but you do have to supply your own micro:bit.

VERDICT

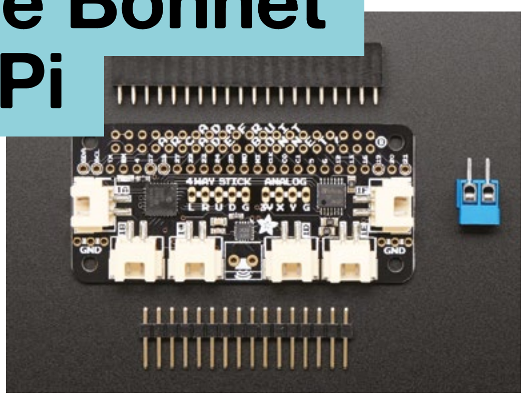
SparkFun micro:arcade kit for micro:bit v2.0

There is even a kit for micro:bit.

8/10

Adafruit Arcade Bonnet for Raspberry Pi

ADAFRUIT ◆ **$14.95** | **adafruit.com**



Left ◆
Add buttons to the tiniest Raspberry Pi

The Raspberry Pi is a great system for both large and small arcade builds. The Adafruit Arcade Bonnet for Raspberry Pi is the perfect match for those smaller form factor builds. The board is the same size as a Raspberry Pi Zero and makes your build easy. You will have to solder on the header pins and speaker connector, but once you've done that, it's all plug and play.

The Bonnet has a few extra features too. There is an on-board 3W digital speaker output and some circuitry that includes an I2C-GPIO converter for all the buttons, reducing the required number of pins. It also allows for old-school gated joysticks or analogue joystick input. And, although you do have to do a little soldering at first, when it comes to assembling your arcade system in the final enclosure, having all these components plug and play via the JST connectors makes that step a lot easier. Just don't forget to pick up some compatible wires that are also available at Adafruit.

VERDICT

Adafruit Arcade Bonnet for Raspberry Pi

Great for small builds.

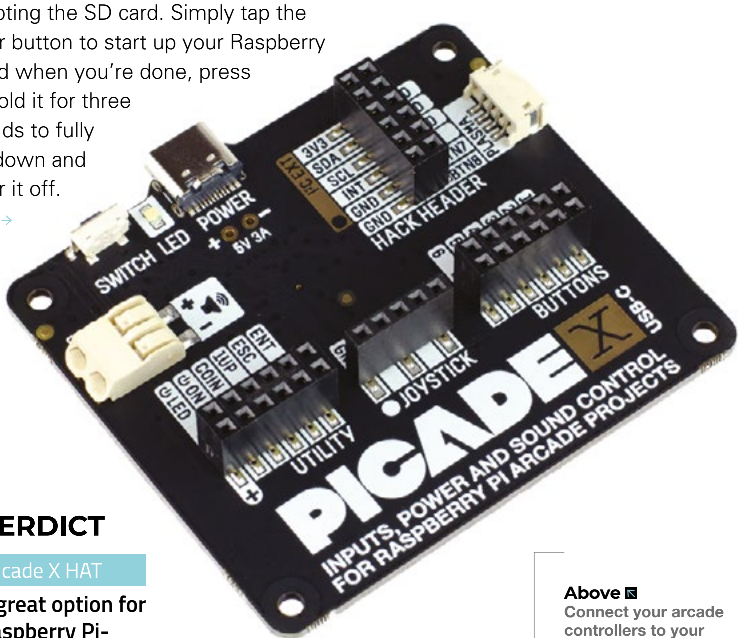
9/10

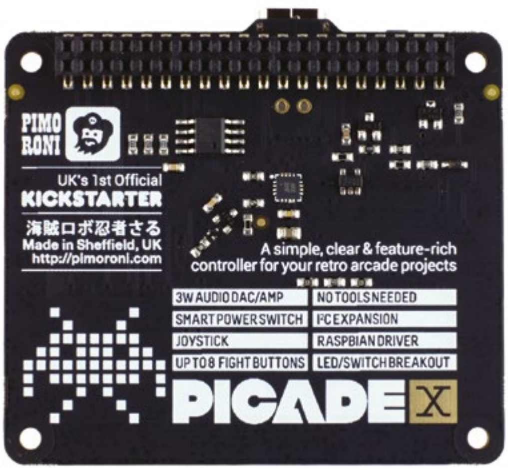
Picade X HAT

PIMORONI ◈ **$15.64** | **pimoroni.com**

Want to scratch-build a no-solder Raspberry Pi arcade system? Then check out the Picade X from Pimoroni. With this HAT, you can easily add buttons and joysticks to a Raspberry Pi board. It features a 3W amplifier, soft power switch, and lots of female DuPont connectors for a no-solder arcade build. Just keep in mind, you'll also want to pick up the matching Picade wiring loom to make all those connections.

The soft power switch allows you turn your Raspberry Pi on and off without the risk of corrupting the SD card. Simply tap the power button to start up your Raspberry Pi, and when you're done, press and hold it for three seconds to fully shut down and power it off. Nice!





VERDICT

Picade X HAT

A great option for Raspberry Pi-based gaming.

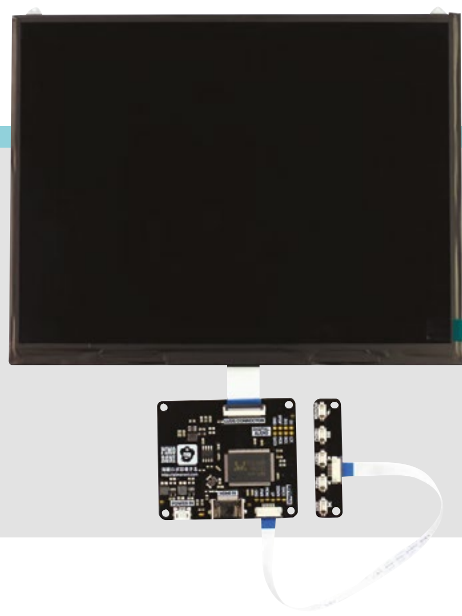
10/10

Above Connect your arcade controllers to your Raspberry Pi

HDMI 10" LCD SCREEN KIT

PIMORONI ◈ **$118** | **pimoroni.com**

Most people will simply repurpose an old TV for their casual arcade building needs, but others will want something a little more custom. Specifically, something that has a small bezel and is ready for a custom cabinet. And that's where the HDMI 10" LCD Screen Kit from Pimoroni comes into play. The kit comes with a 1024×768 resolution display and a custom-designed PCB for power management and breakout buttons for easy access to the display settings. If you are going to build a desktop arcade system, be sure to look at this kit.



BEST OF BREED

Picade Console

PIMORONI ◆ $88.50 | pimoroni.com

The Picade Console from Pimoroni takes all the great features of the popular Picade X HAT, and couples them with all the required buttons and joystick to make a complete arcade system housed in a beautiful and compact retro-style case. You can pick up all the components separately, but the real value here is in the case to house everything. It features black powder-coated MDF panels and an acrylic top, giving it a truly authentic look and feel.

> "It features black powder-coated MDF panels and an acrylic top, giving it a truly authentic look and feel"

And for an added special touch, the instructions are printed on the back of a fun retro-style poster that would make a great addition to your game room. Just keep in mind that you do have to supply your own TV and Raspberry Pi. And as of right now, Raspberry Pis are tough to source. If you already have one, however, this will make for a fun build! □

Below Arcade machine and controller all in one

VERDICT

Picade Console

Fun and compact retro gaming.

9/10

COIN ACCEPTOR - PROGRAMMABLE 4 COIN TYPE

ADAFRUIT ◆ $39.95 | adafruit.com

If you are going to go all out and build a custom arcade system, you might as well be compensated for the good times! With this coin acceptor from Adafruit, you can start collecting some money to help offset the build cost. Or, maybe just the extra money to help replace all the snacks and drinks your friends inevitably will be taking since your house is the coolest place in town.

Make Your UNO Kit

Arduino's first soldering kit

ARDUINO ◈ **€ 55** | **arduino.cc**

By **Ben Everard** @ben_everard

We've tested out a lot of soldering kits over the years, and there's a very diverse range available. Some are simple and fun, to give people their first taste of soldering. Some are complex and designed to push your skills. Some are a way of getting short-run hardware without the added expense of someone assembling it for you. Where does the Arduino Make Your UNO Kit fit into all this?

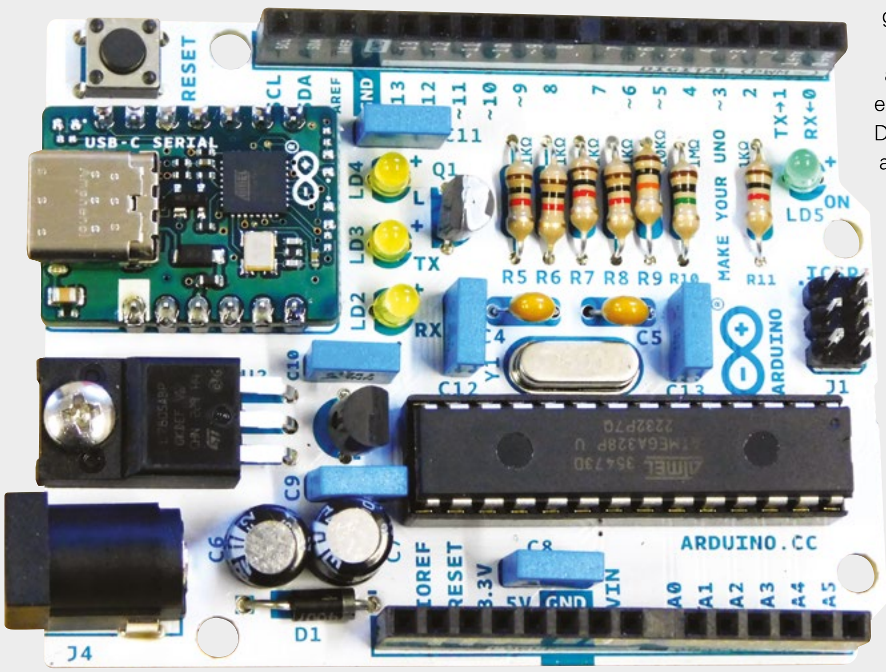
There's no particularly complex soldering – it's all through-hole, with no small components or parts too close to each other. The one bit that could be difficult (the USB-to-serial adaptor) comes as a pre-soldered module. However, there is quite a lot of it. An experienced solderer can probably do everything in an hour or so, but it could take considerably longer if you're not familiar with components or the process of soldering.

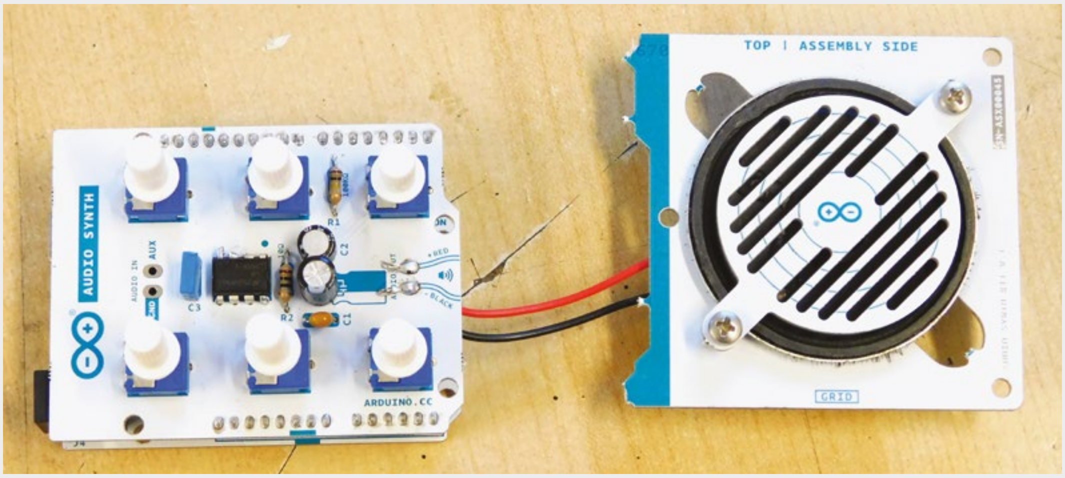
For beginners, this does come with a couple of practice PCBs that you can try soldering. They're much smaller and not really needed, so it's not a problem if you mess them up. If you're a complete beginner, going through everything in this kit will build you up to a reasonable skill level. You'll have plenty of practice with standard components, and probably be ready to try some more challenging soldering.

It is really well-documented – there's a 3D guide that shows how everything fits together. However, despite all this, we did make an error, and we're not convinced it's our fault! Let us explain. The kit contains two practice PCBs, a DIY UNO, and a synth shield. The synth shield also has a speaker holder. All these are contained in separate PCBs that are on two panels. Each PCB is designed to snap out, and that's exactly what we did at the start – and this matches the 'finished' photo on the DIY UNO guide. However, if you snap them all out, you can't then fit it all together in the cardboard case that comes with the kit. This doesn't stop it all working, but means you can't get the tidy look of the finished product that's shown on the website. It's not the end of the world, but a bit more guidance on this at the start would be appreciated.

The synth shield is basically six potentiometers and an amplifier. Five of these potentiometers connect to the analogue inputs on the Arduino UNO, and

Below The final UNO can be used just as any other Arduino UNO





Left You can make some noise using the Mozzi library

Below All the surface-mount components come pre-soldered

the sixth is a volume control that also has quite a big impact on the tone of the sound. The amplifier is connected to a PWM-enabled pin (D9), and filtered through a capacitor, so you can create simple sounds using the tone library, or more complex ones using Mozzi (a library for creating sound on microcontrollers, see **hsmag.cc/Mozzi** for details). At the moment, there's still a relatively limited set of sketches available to play that are specifically designed for this shield, but you can of course program your own. Mozzi is well-documented and, if you have a bit of Arduino experience already, you should find it reasonably straightforward to get started.

> "If you have a bit of Arduino experience already, you should find it reasonably straightforward to get started"

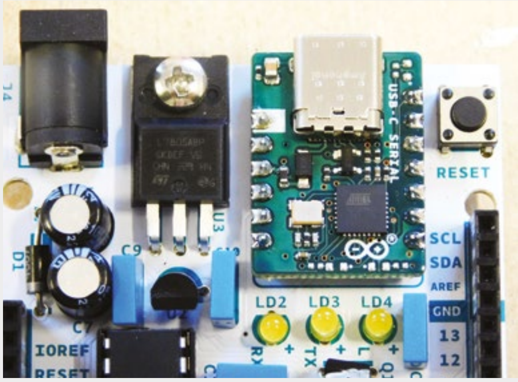
The synth shield is great fun to play with. We'd appreciate a bit more hackability – particularly adding connectors, so you can add additional hardware to unused GPIOs. For example, buttons to trigger the sound generation. Arduino shields can be stackable, but this shield doesn't have stackable headers. You could add a proto shield underneath this to add additional connections, or you could solder onto the joints for the headers. Even without this, though, there's plenty of fun to be had with the synth shield.

You don't have to use the synth shield. Your UNO will act just like any other UNO, and you can use it for almost any microcontroller project you like. By modern standards, it's not particularly small, powerful, or connected, but that's often not a problem for a microcontroller.

There's a lot of attention to detail in this kit that makes it good to use. For example, the through-holes only have copper exposed on one side of the PCB, which means you can't solder components on the wrong side – something this author has done on multiple other soldering kits. The 3D soldering guide is really easy to follow and shows exactly what goes where.

This is also – and this is an important point – a really good-looking PCB. The silkscreen shows the outline of the component combined with the schematic in a way we've not seen before (yes, we know they use the American symbol for resistor rather than the EU one, but we're not going to hold that against them). The result is informative, helpful for people getting started with electronics and, to our eyes, beautiful. It helps that it's done in Arduino's teal solder mask rather than the traditional green.

At 55 euros, this is an expensive kit. However, it's the most excited we've been about an Arduino release for a while. It's a great introduction for new people coming to soldering, it's a fun kit to play with, and you've got perhaps the most common microcontroller board available to continue your adventure however you wish. For your 55 euros, which is approximately the same in pounds and dollars at the moment, you can get a lot of enjoyment.



VERDICT

A fun kit to help you learn to solder and make some noise.

10/10

CROWDFUNDING NOW

Koi Lanterns

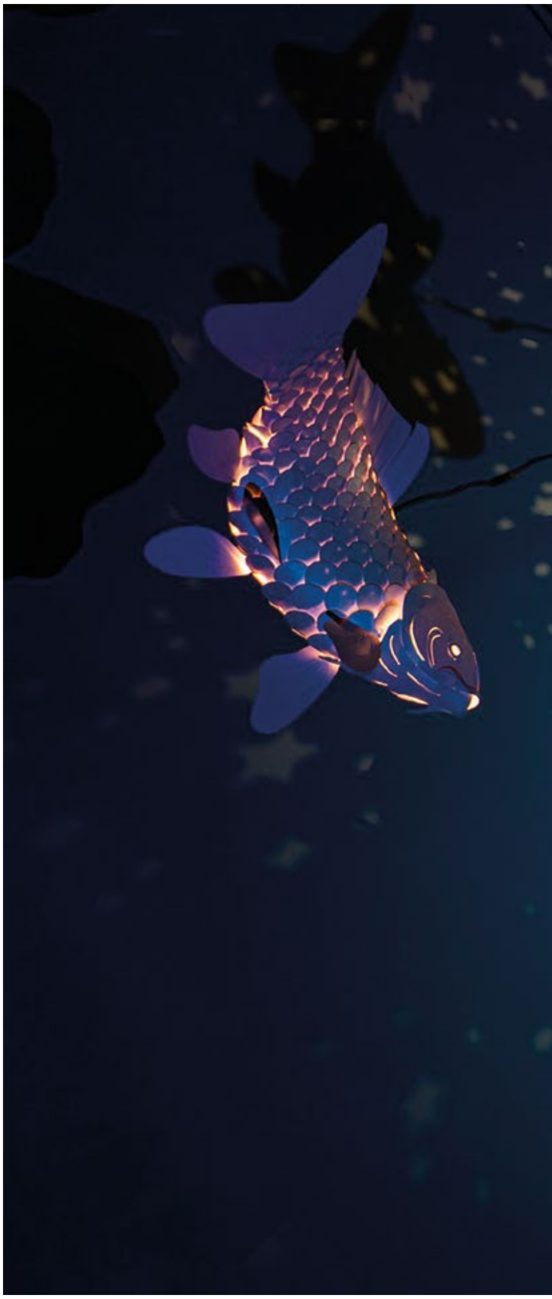
A new way to show off LEDs

From $29 | **hsmag.cc/koi** | **Delivery:** May 2023

Paper is a fun material to work with. It's cheap, ubiquitous, and easy to shape. Yet it's something we often don't think about for fabrication. However, there's a long history of making things with paper – most notably origami, but there are others. The Koi Lanterns by Yuumei Art will come as laser-cut shapes that fit together without glue or tape, to create poseable fish models.

Paper and LEDs work really well together. Here at HackSpace mag, we're big fans of using thin paper as diffusion material, but these Koi will use thicker card and let the light shine through the gaps. The texture on the surface of paper makes for a very soft light that illuminates gently. These kits come with LEDs, but it looks like a great place to flex your maker muscles with some colour-changing options. A gentle flow through the spectrum of colours could look wonderful.

The build looks quite involved. There are a lot of small parts that need to slot together gently, and it would be easy to accidentally crumple and fold. Although we've not had a chance to try this out, it looks like assembling it could be quite a meditative process.



Right We absolutely love the effect of LEDs shining through paper

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.

next month

issue #63

ON SALE 19 JANUARY

FLYING MACHINES

ALSO

- LEDS
- SOLDERING
- 3D PRINTING
- CIRCUITPYTHON
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe