MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | May 2024 | Issue #78

THE FUTURE OF MAKING

New skills, new ideas, new technology

- Hackable smart watch
- 3D printed gardening tools
- Open source Braille reader

May 2024
Issue #78 £6
Raspberry Pi PRESS
9 772515 514006 78

CASH REGISTER TINY COMPUTER **WEATHER**

Welcome to HackSpace magazine

What does the future hold? It's one of those impossible questions that is so inviting, and sometimes seems straightforward. However, if you try to pin down the future, it becomes slippery. Who had a global chip shortage in their predictions for the early 2020s?

Still, we're trying to pin down a few people with a good insight into the field of maker technology and find out what they think is going to happen. But before you turn the pages and find out what they have to say, have a bit of a think. What do you think is going to happen? Will the future be 3D-printed, or will that remain geeks only? Will AI change everything, or is it just another fad?

What do I predict? I predict that whatever happens in the technological or geopolitical landscapes, people will go on making things, just as they have done for the past couple of million years. Civilisations come and go, technologies evolve, but people are still people and we make tools to solve our problems. It's what we've always done, and I suspect it's what we'll always do.

BEN EVERARD
Editor ben.everard@raspberrypi.com

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

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

GET IN TOUCH

hackspace@raspberrypi.com

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

ADVERTISING
Charlotte Milligan
charlotte.milligan@raspberrypi.com
+44 (0)7725 368887

DESIGN
Head of Design
Jack Willis
Designers
Sara Parodi, Natalie Turner
Illustrator
Sam Alder
Photographer
Brian O Halloran

CONTRIBUTORS
Marc de Vinck, Jo Hinchliffe, Rob Miles, Nicola King, Andrew Lewis, Phil King, Stéphane Godin

PUBLISHING
Publishing Director
Brian Jepson
brian.jepson@raspberrypi.com
Director of Communications
Liz Upton
CEO
Eben Upton

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
+44 (0)1293 312193
hsmag.cc/subscribe
hackspace@subscriptionhelpline.co.uk

recycle

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

HackSpace magazine is published by Raspberry Pi Ltd, 194 Cambridge Science Park, Milton Road, Cambridge, England, CB4 0AB.

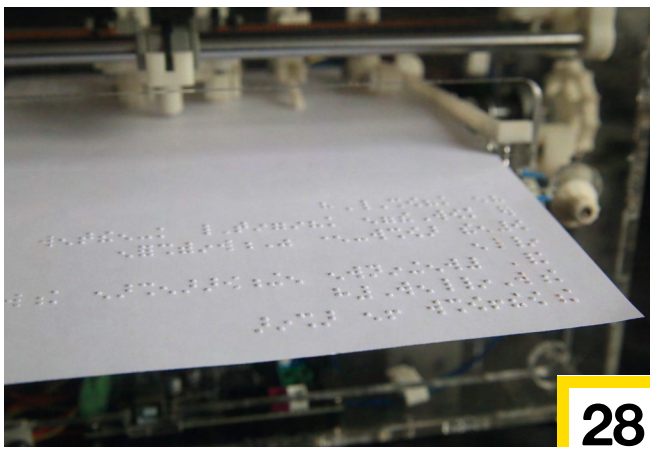
ISSN: 2515-5148.

Contents

48

Cover Feature

18

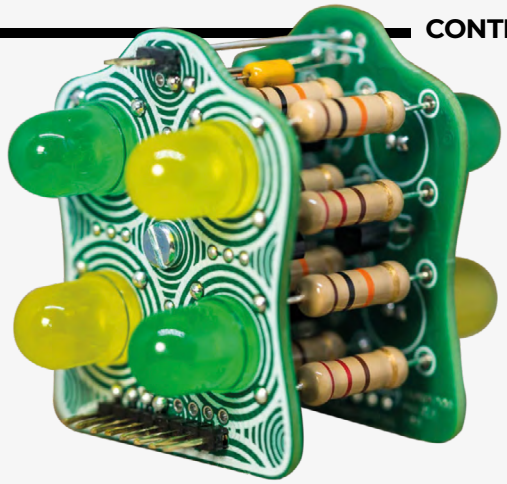


28

Tutorial

3D-printed planters

66 Grow fresh, tasty herbs with a tiny bit of space and some 3D-printed planters



Interview

The Sanjay Mortimer Foundation

Review

Bangle.js 2

53 FORGE

85 FIELD TEST

REGULAR

Desktop flamethrower

By Mellow_Labs

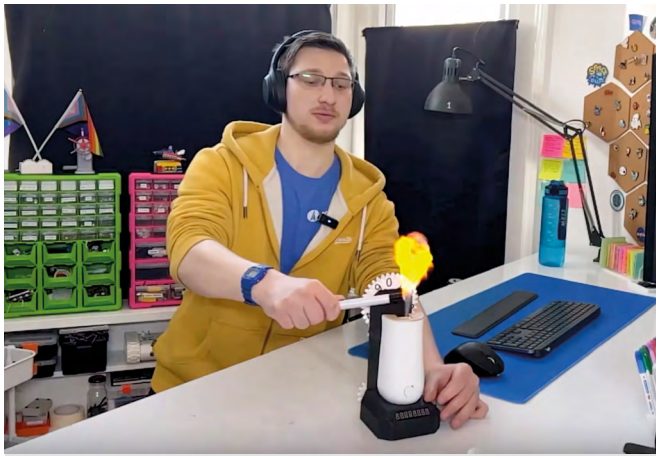
hsmag.cc/Mellow_Labs

File this one firmly under 'do not try at home'. Unless you have the skills, patience, and safety protocols in place to make what is essentially a DIY flamethrower. It's not just a flamethrower, however, it's a subscriber counter that shoots fireballs for every ten new YouTube subscribers.

Maker Tomasz wanted to celebrate his friend Dan's new YouTube channel, so he built him something tangible to encourage him to increase his online following. Rather than gifting Dan with something pedestrian like a numerical display ticking upwards as your follower count grows, Tomasz took inspiration from the name of his friend's YouTube channel: Gears, Code, and Fire. The final product combines all three of the elements: gears, some code, and the aforementioned incredibly dangerous fire.

Before the internet, one of the activities I imagine some people partook in to pass the time was making a flamethrower out of a can of deodorant and a contraband lighter. Tomasz's follower counter is literally the same thing but levelled up with automated gears that switch the lighter aflame and spray the deodorant for you. Fourteen-year-old me did not foresee being concerned with these things so many years later, let alone in a professional capacity, but here we are. ◻

Right
A big gear with numbers 0–9 sits atop the flamethrower. It's controlled by a stepper motor, and the numbers relate to the last digits of Dan's YouTube follower count



3D-printed marble clock

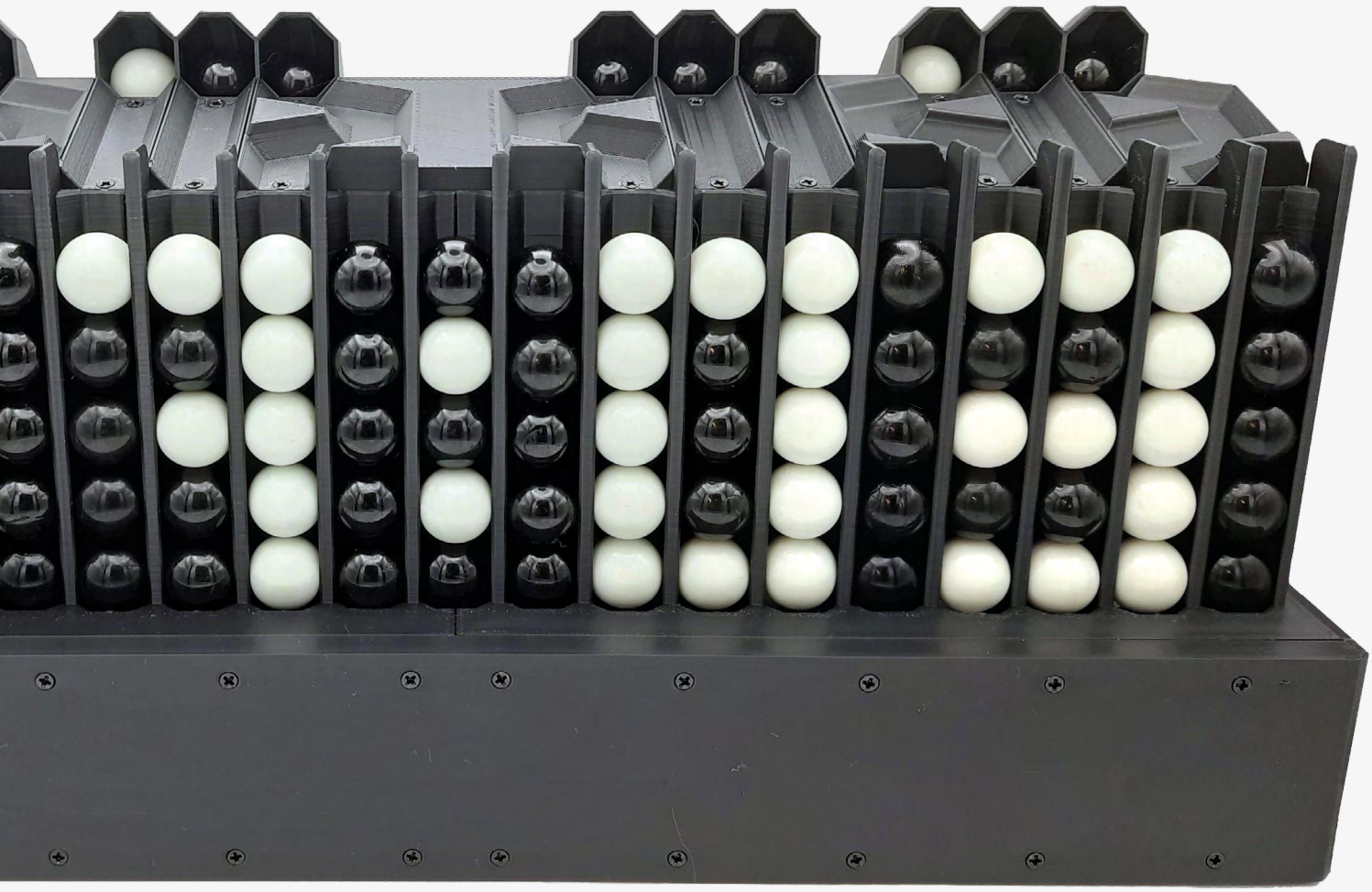
By **Sakati1984**

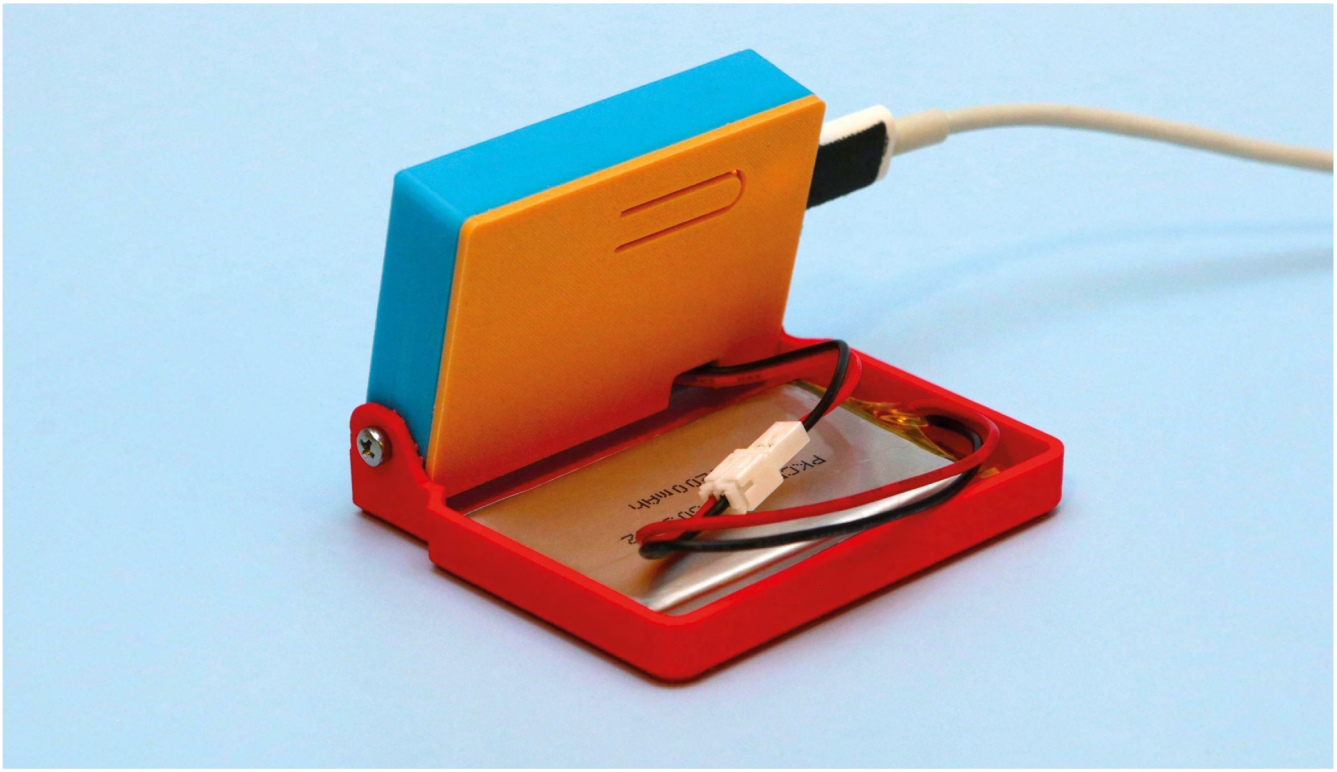
hsmag.cc/MarbleClock

We mastered clocks a while ago. Each iteration on the clock is basically an art piece. Polished brass, Nixie tubes, even fancy Rolex wristwatches – for anyone with a phone in their pocket, they're all obsolete, but they look brilliant. Joining the ranks of useless beauty is this creation from Sakati1984. They've used a Raspberry Pi Pico, 227 marbles, twelve servos, two metal rods, and a handful of other components to create this fully functional clock.

Rather than selecting balls on the fly to represent the characters, the machine relies on the balls being in the right order, making this useful for telling the time, but not for displaying live data. The refresh rate is also seven seconds, so there's no point adding a seconds counter – for that, you can just look at your smartphone.

Right
This clock is louder than thunder, and is thus unsuitable for bedroom use





IoT battery charger

By **Clarke, Rembor & Ruiz²**

hsmag.cc/BatteryMonitor

The internet is cluttered with pointless devices, promising convenience but offering little more than outdated software, security vulnerabilities, and subscription services offering an umbilical link to companies that can put up prices when they feel like it or even go bust.

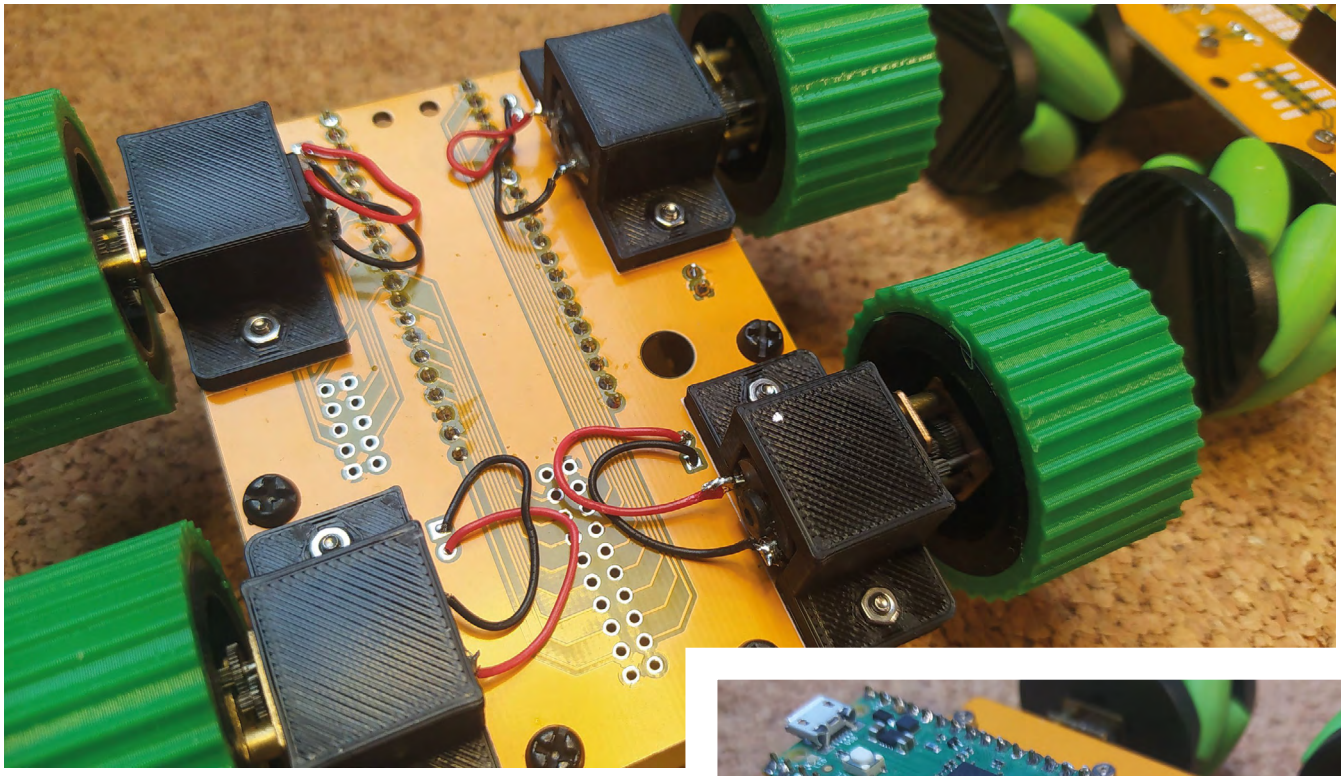
This IoT battery charger by Liz Clarke, Kattni Rembor, and the Ruiz brothers avoids all these pitfalls. For one, it's built using open-source hardware and software, so should Adafruit vanish tomorrow, it'll still work. And in contrast to many other IoT devices, it's actually useful.

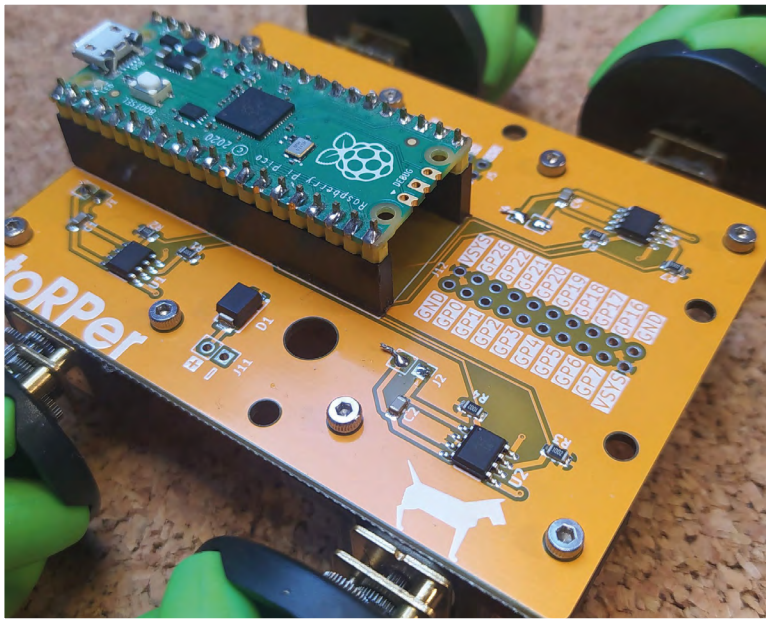
This is a pretty simple build, beautifully executed: it's just an Adafruit ESP32-S2 Reverse TFT Feather (a Feather word with a built-in TFT screen) and a MAX17048 LiPoly battery monitor chip. The chip detects your battery's charge, and the Feather displays it on screen. So far so good, but the IoT element is where this build shines. Using Adafruit IO, the device can let you know when the battery is charged, via a text message or an email. ◻

Right ◈
You get an email or a text message to let you know that your battery is full


3.80 V
%

REGULAR





StoRPer

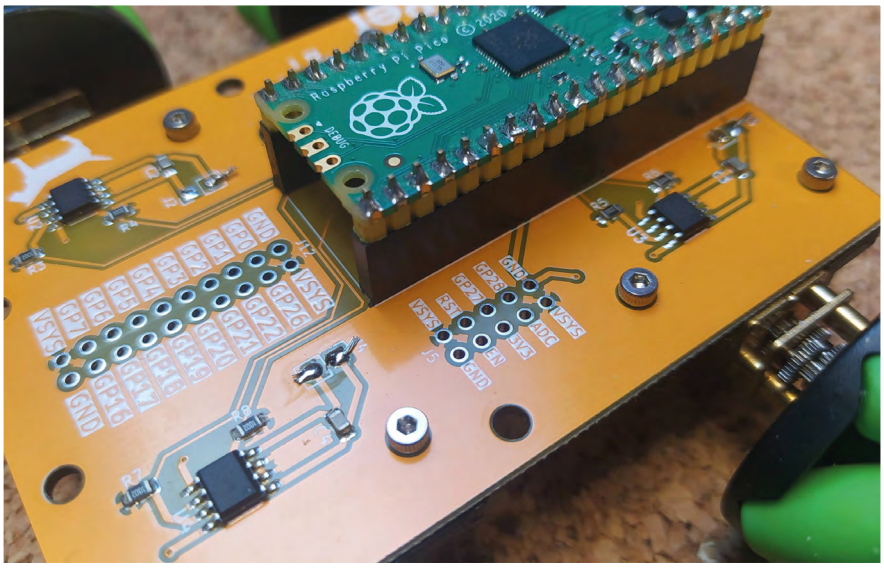
By **Jo Hinchliffe** hsmag.cc/Storper

As a child, Jo Hinchliffe used to love playing with Stomper 4WD cars – small, battery-powered vehicles that could climb seemingly anything. Now, as an adult, he's made his own version, powered by Raspberry Pi Pico and for sale on Tindie at pocket money prices.

At just $16, this is a cheap way to get into robotics. It's not a full kit – you'll need to supply your own N20-style motors, wheels, and your own Raspberry Pi Pico which gives you programmability, elevating StoRPer from a 4WD vehicle to a robot.

The PCB includes four motor drivers, giving users the flexibility to drive the robot how they see fit. You could, for example, keep it simple and attach one motor per wheel. Or you could take advantage of the ability to control each wheel and opt for mecanum-style wheels. Or you could choose to add one motor per side to control two wheels, freeing up the other two motor drivers for google eyes, gun turrets, or whatever else you want – the choice is yours.

We like this project, and not just because it's by one of our favourite humans (and the author of our KiCad PCB design tutorial series). We like it because, even though it's cheap, it's got loads of potential for experimentation. ◻



Left
This project has been on Jo's workbench for ages – we're glad he's finally got it out the door

REGULAR

Universal remote control

By **Armerdan**

hsmag.cc/UniversalRemote

Reddit user Armerdan needed a device to replace an old Logitech Harmony 512 universal remote. Like an absolute genius, they decided to model the replacement device on a '1950s sci-fi death-ray blaster-type thing'. If you squint hard enough, you can make out that this device, which is based on an Arduino Pro Micro, uses an enclosure from a guitar effects pedal. What might escape you at second glance is that the antenna is made from a couple of transparent plastic yo-yos that have been drilled out, topped off with a drawer knob.

The maker isn't 100% happy with how this device looks (even though it looks awesome!). The joystick, while functional, isn't in keeping with the aesthetic they're going for, so in the near future, that'll probably be replaced with a miniature red ball on a silver pole. We reckon it's perfect as it is, though – if you watch the original *Star Wars* film closely enough, you'll notice that all their sci-fi weaponry is bits of broom-handle held together with sticky tape. In comparison, this is a paragon of design harmony.


Power switch to turn the TV on and off
Joystick: directional control for navigating menus
ON
OFF
Rotary encoder volume control. Push to mute
Coloured buttons activate macro functions to select input sources

HiPi.io

HIGHPI PRO

The new case from the HiPi.io team

- Rapid tool-free assembly and disassembly
- Large internal volume for HATs
- Compatible with Pi 2/3/4
- Multiple lid options
- Passive & Active Cooling options
- Secure microSD card cover
- VESA mount support
- Molding-configurable output ports customizable for volume orders
- Printed logo for your branding

Available at these great Pi stores:

PiShop.us PiShop.ca PiShop.mx The PiHut Pi-Shop.ch Welectron.

Contact your favorite Pi store if it's not listed here

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

FEATURE

THE FUTURE OF MAKING

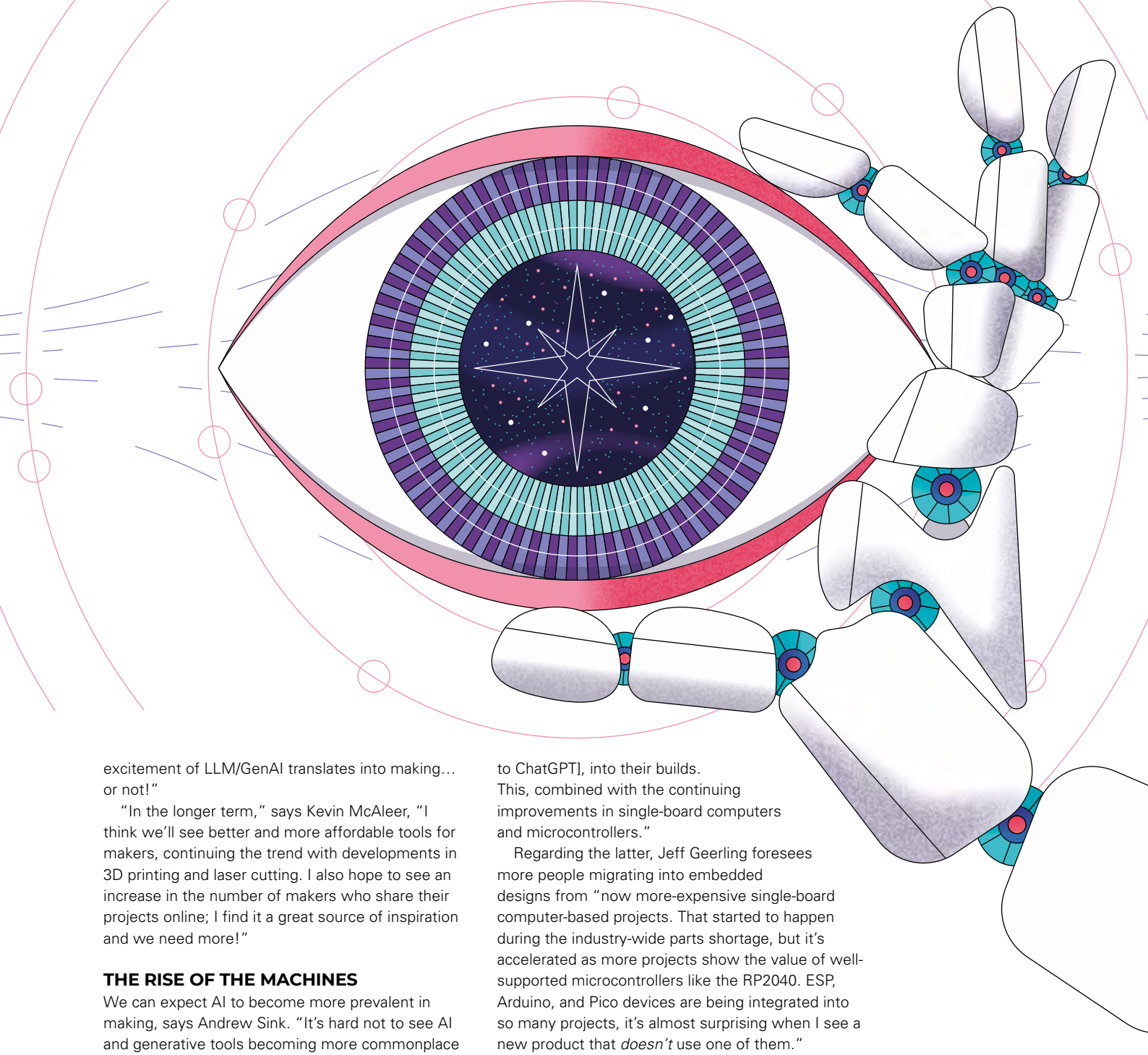
What lies ahead for the world of making? We ask the experts...

Words by Phil King

Predicting the future is fraught with problems – we're still waiting for flying cars to become the norm. Still, we'll give it a go… We've consulted some movers and shakers on how things may change in the makersphere over the next few years. Will making become more mainstream? What trends are we likely to see in the fields of SBCs, microcontrollers, robotics, and 3D printing? What does the future hold for open-source hardware? And will the emergence of AI affect how makers go about creating projects?

Looking ahead towards the next five to ten years in the makersphere, the experts we consulted were able to point out some emerging trends.

First of all, on a cautionary note, Limor Fried hopes there won't be another chip shortage, "as that really set folks back and damaged a lot of the industry: there's recovery, but we know of a lot of maker companies that are still struggling or closed shop. The flip side is we're starting to see more chips and technologies come out, although a lot of it is focused on AI – it will be interesting to see how the



excitement of LLM/GenAI translates into making… or not!"

"In the longer term," says Kevin McAleer, "I think we'll see better and more affordable tools for makers, continuing the trend with developments in 3D printing and laser cutting. I also hope to see an increase in the number of makers who share their projects online; I find it a great source of inspiration and we need more!"

THE RISE OF THE MACHINES

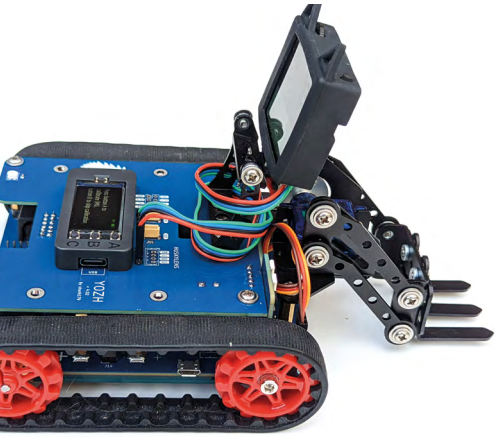
We can expect AI to become more prevalent in making, says Andrew Sink. "It's hard not to see AI and generative tools becoming more commonplace as the barrier to entry (cost, required programming skill, etc.) continues to lower. Personally, I am excited to see new software tools and workflows enabled by these advancements, especially where niche hardware and software disciplines overlap."

Kevin adds: "I think we'll see an increase in the sophistication of the kinds of projects people are building where they combine AI tools, such as Ollama [the locally hosted, offline alternative to ChatGPT], into their builds. This, combined with the continuing improvements in single-board computers and microcontrollers."

Regarding the latter, Jeff Geerling foresees more people migrating into embedded designs from "now more-expensive single-board computer-based projects. That started to happen during the industry-wide parts shortage, but it's accelerated as more projects show the value of well-supported microcontrollers like the RP2040. ESP, Arduino, and Pico devices are being integrated into so many projects, it's almost surprising when I see a new product that *doesn't* use one of them."

The lower costs and greater user-friendliness associated with microcontrollers will also draw more people into the world of making, reckons Jeff. "These platforms are beginner-friendly. Making the technology more approachable with things like full MicroPython support means more makers can build features that were very difficult just a few years ago. I see an influx of makers who don't have much of a coding background." →

TUTORIAL



"It will be interesting to see how the excitement of LLM/ GenAI translates into making... or not!"

MAINSTREAM MAKING

Above
Programmable robots are cheap enough these days, or if you want a challenge, you can use 3D printing (or bend sheet metal) to build your own

Above Right
Limor Fried – an electrical engineer and the founder of Adafruit Industries (adafruit.com). She is very influential in the open-source hardware movement

This brings us neatly onto the subject of whether making will become more mainstream over the next few years.

"I think it will continue to grow," says Matt Venn. "We see a lot of creativity with mobile phones and apps, as more people have access to tools, but I do think that there is something special about physical making. Holding something in your hands that you made is special, and as the barriers to that continue to drop, more people will get involved. Having access to the tools, equipment, and time is a privilege that still many people don't have, so as well as appealing to people from different backgrounds, we also need to support them when they are interested."

"Making will be more mainstream because of accessibility and community," agrees Jeff. "Kids especially are at the leading edge of 'mainstream' maker culture, for example CrunchLabs build boxes, or Raspberry Pi's deep penetration into classroom STEM projects. That will have long-term effects as kids are exposed to the tech, just like my generation was inspired by video games and the early internet."

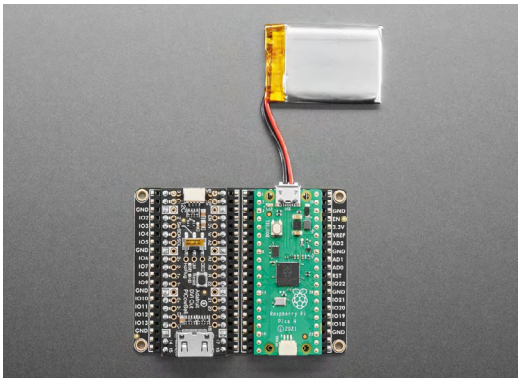
Limor concurs: "We think that making gains a lot of STEAM (ha!) from being introduced in schools, so as we saw schools returning back to in-person instruction and workshop projects, we'll also see more makers. I'm a strong proponent of meeting kids where they are: if they have a background that they love medicine, teach them about how to build a pulse sensor. If they love animals, show them how to craft a 'smart' chicken coop monitor or an automatic pet food bowl. Fashion lovers can drape themselves in LEDs! Skateboarders can use accelerometers to analyse their air time."

Kevin points out that making is already more mainstream than it was ten years ago. "I think that's a combination of events like MakerFaire, magazines (like HackSpace), and the availability of parts from companies like Pimoroni and Adafruit. The ecosystem for makers is there for makers to be part of and to learn from other makers too, particularly with social media and YouTube…." He has one caveat: "I think for making to become even more mainstream, it needs to lose a bit of its nerdy image. Making is for everyone!"

LOWERING BARRIERS

Not only is making likely to become more mainstream, it will continue to have an impact on mainstream technology, says Jude Pullen. "Through their enthusiasm and experimentation, makers can create unexpected breakthroughs in technology that shape our future. It spreads the 'bets' on where the next revolution or evolution will come from… The makerspace communities often 'tinker' with future tech long before it reaches wider adoption."

He gives the example of bio-hacking communities, "where even school kids can sequence a plant genome using only ~$1000 worth of hardware. This is kind of mind-bending when you think about it. Another space I'm watching is edge computing, but this is going to be governed by privacy and security. So this is sometimes the 'friction' between makers' can-do attitude, and industry still needing to be compliant and safe."



Left
When it comes to open hardware, more and more projects are being OSHWA certified, such as this Adafruit Proto Doubler PiCowbell for Pico

Far Left
Matt Venn – science and technology communicator and electronics engineer who runs the Zero to ASIC chip design course – zerotoasiccourse.com

"I do think we'll see a lot more work in the ASIC / semiconductor space"

BETTER BOARDS

In the field of single-board computers and microcontrollers, Jeff says that software support and community are paramount. "Companies like Raspberry Pi build 'stickiness' to their platforms with software and support. The companies that get that, and invest in software and hardware-adjacent community development, will lead the next few years of growth."

While he reckons there's still headroom for more performance in the boards, "I think on the hardware side, there will be more competition on integrated chip features and efficiency, like with ESP devices adding built-in Matter/Thread support."

"I think we'll continue to see innovations at all levels," agrees Matt. "I think the RP2040 is a remarkable chip that no one expected. We also see more FPGA fabric incorporated (e.g. the latest Microchip parts). We'll have to see whether people go for PIO or FPGA. One thing where Raspberry Pi excels is in documentation. They know it's as important as the hardware if you want adoption in the maker space."

He also thinks there's still room for speciality devices such as the Daisy (**electro-smith.com**). "They aren't exactly doing anything new, but they're very successful because they have a niche, and they are the easiest way to do embedded music."

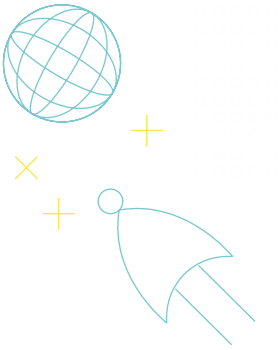
Limor predicts lots more use of microcontrollers for IoT and home automation. "We're also seeing more folks get comfy with the idea of an interpreted language on a microcontroller: with tons of processing speed and RAM, why not convert that into ease-of-use! And RISC-V is also gaining attention; it's great to see a competitor to Arm."

The use of SBCs in industry will continue to grow, she adds. "Why design when you can pop in a ready-to-go pre-built and pre-certified machine?"

Cryptographic chips, such as in the Arduino Nano 33 BLE, will make boards more feasible for industrial and commercial use, says Jude, where data security is vital. "I think it's a clear sign of Arduino's journey catering to makers/amateurs, but now wooing those who also need a more professional architecture."

He gives the example of an insulin monitor/dispenser that was shown to him when he worked for Sugru: "The idea was certainly solid, but the execution was not robust enough for us to publicly champion. Some ten years later, this is something many startups are working on – so it's not the originality; it's the security, the 'real-world' stuff, that is the key to success." →

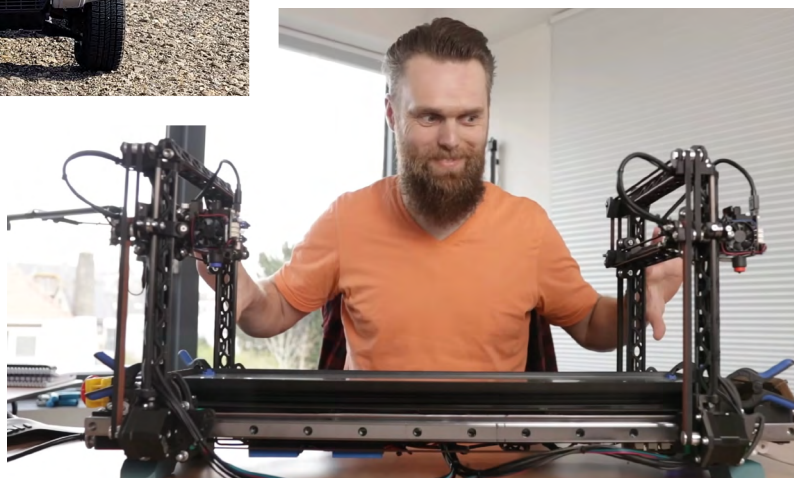
"We will probably see a lot more hobby robots becoming full products"



Above
Kevin McAleer – a hobbyist robotics fanatic who likes to build robots and share videos about them on YouTube and his blog (kevsrobots.com)

Above
Better CAD tools could help makers to create more complex 3D printing projects, such as this 7.4:1 scale model of a Mitsubishi Pajero by Casadio Design

Right
Makers are pushing the boundaries in 3D print technology, such as with Jón Schone's TIME machine, which is in effect two 3D printers in one



3D PRINTING

"In 2024, makers can purchase a desktop filament 3D printer that is fully assembled and ready to print for under $300," notes Andrew. "It's hard to predict when the market saturation of desktop 3D printing will occur, but we're reaching a point where makers are more likely to have used a desktop 3D printer than not, something that wasn't always the case.

"The exciting next step for the field of desktop 3D printing is the continuing trend of makers who are pushing the boundaries of existing technology and sharing their work to inspire a new generation of makers," he continues. "For instance: Jón Schone's TIME (True Independent Multiple Extrusion) 3D printer (**hsmag.cc/SchoneTIME**), which essentially adds an entire second 3D printer to an existing 3D printer, is a novel development that will no doubt be adapted in ways that are difficult to predict. It's an exciting time for desktop 3D printing, and the future is brighter than ever for this technology."

Advancements in 3D printing hardware will be matched by those in software, reckons Jude. "Although there will be more detail, more materials, new processes (FDM, SLA, SLS, etc.), and hopefully better safety awareness (do please get extraction – even with PLA!). I think it'll be software that creates the biggest impact in innovation: CAD software will reasonably start to have 'feedback' loops. For example, photogrammetry is now so advanced that if you scanned the resulting 3D print you created, the 3D printer might be able to 'look' at the difference between what you got and what you designed – and offer to create a second improved attempt."

DIY robotics

In the world of robotics, Kevin McAleer thinks we'll see more sophisticated homemade robots: "Once a technology is open-sourced and documented well, it tends to be picked up and enhanced by the community. If you compare this decade with say the 1980s, you'll find robots being created all over the world in coder dojos, maker spaces, schools, and colleges."

He is also hopeful that the hobby will become less expensive, as "the parts for robots can currently be an obstacle for some."

He doesn't envision robots becoming widespread in the home yet, though. "I think we're still a long way off from having robot butlers or humanoid robots in general, and that's for a number of reasons, including cost, complexity, and more practical aspects such as battery power and safety concerns." The news is brighter on the DIY robotics front, however. "We will probably see a lot more hobby robots becoming full products – Otto DIY [**ottodiy.com**] is a good example of that."

Home-built robots are likely to become smarter, too. "From an SBC and microcontroller perspective, I think we'll see a lot more power and capability in this space, particularly around AI add-ons to make, say, object detection a simple add-on, freeing up the main SBC board to do other work. Large language models have seemed to come out of the blue in the last two years. I suspect we'll see a lot more development in this area." To this end, he thinks we'll also see more AI-related add-on boards for SBCs.

Above
DIY robotics is set to grow in popularity and scope. Expect to see more hobby robots become full products, such as the ultra-customisable Otto DIY

"I see an influx of makers who don't have much of a coding background"

Left
Jeff Geerling – software developer and YouTuber whose channel shares a wide variety of tech videos: hsmag.cc/JeffGeerling

OPEN HARDWARE

On the subject of open-source hardware, Limor is optimistic. "It's looking good – there's a lot of contributors to OSH and OSS; the OSHWA certifications list goes up every week (**hsmag.cc/OSHWAList**) – it'll probably hit 3,000 registered projects this year. We're seeing more work from folks on 'democratising' documentation and automation: using third-party low-quality manufacturing to iterate and share designs with the manufacturing documentation 'cooked in'. And, of course, KiCad 8 is an exciting milestone. Having a top-to-bottom open-source toolchain is great for the longevity of open-source hardware."

Jeff notes that "open-source hardware is always a hard nut to crack. The time and money involved in building great hardware products is exponentially greater than software, so I don't see open-source hardware as a movement ever approaching the same level of adoption as with software."

He still thinks developing open hardware has value, however. "For makers especially, building or adopting an OSHW design can be a great boon to your own work. Unless you have a strong desire to become an entrepreneur, building completely open designs or working with others on existing designs amplifies your abilities, just like with software." →

FEATURE

Hack and repair

Makers have long taken an interest in hacking and customising existing commercial products. The right to repair is also an important concept, especially with regard to being eco-friendly, but some manufacturers are resistant.

"Having just written a series on RS DesignSpark on repairability, called The Fight to Repair (**hsmag.cc/FightToRepair**), this has been on my mind a lot lately!" says Jude Pullen. "I think industry will improve its repairability of its products when it establishes new business models; e.g. Apple is perhaps not wild about making a phone last twice as long, if it means it halves its profits, so it needs a spares/upgrades/etc. model to 'fill the gap' left by rampant consumerism."

Returning to the subject of makers, "their role I think is best when they are provocateurs. Ken Pillonel is a great example of how he hacked an iPhone to have a USB-C connection. I interviewed him as part of my series for DS (**hsmag.cc/FightToRepairEp9**), and he does not of course believe he single-handedly made Apple 'change their ways', but I think he was certainly an agitator that made it hard for Apple to try to virtue signal. And although a USB-C plug is just one part of the puzzle, it was a way to say 'try harder' to a company that deep-down lives or dies by how 'cool' it is deemed by the early-adopter community."

Jude thinks there is a need and place for "keeping companies honest, with a mix of humour/satire and innovation/provocation to not greenwash or only tackle the trivial."

"Makers can create unexpected breakthroughs in technology that shape our future"

THE IMPACT OF AI

Above Right
Jude Pullen – creative technologist, prototyping expert, engineer, life hacker, and product designer (judepullen.com)

Jeff notes that many makers are already using AI tools "to assist with design, whether it's in coming up with an initial code structure, a 3D model, or even imagery or prose for documentation! For some things, especially mundane aspects where machines can cut out some of the boring work, this is a huge boon."

Andrew agrees that "AI-assisted tools are quickly becoming a fast way for makers to solve problems that would have historically been a blocker for them. For instance: a maker who is struggling to understand a line of G-code is able to receive a detailed analysis using a generative tool and deepen their understanding of the syntax, as opposed to searching for hours for a solution."

"AI is so early," says Limor, "nobody could see the emergent behaviour from massive LLMs like ChatGPT. Personally I love using AI for help with coding Arduino libraries (**hsmag.cc/AdafruitAIDriver**), and Python. I check what's being generated and credit the AI with help; I like to share the whole chat I used to do the code creation. It saves me tons of time. I'm not yet using it for hardware design explicitly – it's great for repetitive/rote work, but it's mistake- and hallucination-prone – I wouldn't trust it to design a board!"

Jude adds, "I see AI tools like this as just 'tools', and potentially as 'hallucination machines' – their outputs are not clear, rational, or robust, but like a dream, they can spark something. They will improve, but right now, this seems enough of a reason to give them a place in our 'toolkit'."

DON'T BELIEVE THE HYPE

Jeff doesn't see AI replacing human creativity in the making process. "Despite all the AI hype, I see it as a tool – just like a good IDE, a well-stocked workbench, or a search engine – and not as something that replaces the creative work makers do."

Andrew observes that it's "difficult to define a point at which a maker is no longer 'making' a thing. Artist Sol LeWitt used sets of written instructions to create artworks as far back as the 1960s (i.e. vertical lines, not straight, not touching, covering the wall evenly), which are arguably an analogue version of using an AI-assisted tool to generate the same completed work. Is that considered making? Good news: it's up to you!

Design your own chip

Matt Venn thinks that more makers will embrace designing their own custom chips, like those created in his Zero to ASIC course (**zerotoasiccourse.com**). "We've gone from 0 to 1000s in a couple of years; we'll see where it goes. The downsides are the long wait times to get hardware back, but the upside is that people can do things they've never been able to do before. With Tiny Tapeout 6, we've added mixed signal support, which is a good answer to 'why not just do it on an FPGA?'."

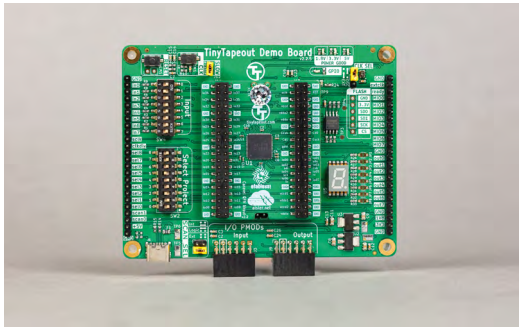
He says the process is "already as simple as designing and ordering a PCB, but the trade-off with simplicity is depth. You could create a chip with an AND gate on it, and order it within 30 minutes from **TinyTapeout.com**, just as you could make a PCB with a single LED on it."

For the time being, at least, he thinks open-source chip design will be of most use for academia and training, self-guided learning. "It's hard to find a low-volume use case that only an ASIC can fulfil."

As for the future of open hardware, "I do think we'll see a lot more work in the ASIC / semiconductor space. The old factories won't go away; they'll get cheaper. And as more people learn the tools, we'll see novel applications that didn't make sense before. As an expert recently told me, the professional tools are crap because the number of users is so low. The open-source tools enable faster innovation and bug fixing so they'll improve quickly – although it's unlikely they'll ever get the billion-dollar investments the closed-source ones benefit from."

"We're reaching a point where makers are more likely to have used a desktop 3D printer than not"

Left
Andrew Sink – senior applications engineer at a leading additive manufacturing technology company, he shares his 3D printing adventures on YouTube – hsmag.cc/AndrewSink



BRIGHT FUTURE

One thing our experts agree strongly on is that the future is bright for the makersphere. "I'm incredibly optimistic about the future of making," says Jeff. "For many people, making is a compulsion, and outside of the freedom of expression being stripped away, there's nothing that will stop it!"

"There are so many cool projects out there," adds Kevin. "The community, suppliers and educators all help to create an environment for makers to find their happy place and to share parts of their soul with others in the form of robots, code, projects, builds, and art."

Finally, Jude notes that so many more of the world's problems are "being solved by small groups of people… the global community can do more than ever before to shape a better future." □

SUBSCRIBE TODAY

GET SIX ISSUES FOR JUST:

£30 UK / **€43** EU / **$43** USA & Canada

SUBSCRIBER BENEFITS:

- **Get every issue of HackSpace magazine delivered to your door**
- Learn from hackers and makers in our in-depth tutorials
- Early access to the PDF edition
- **Get a free Raspberry Pi Pico W**

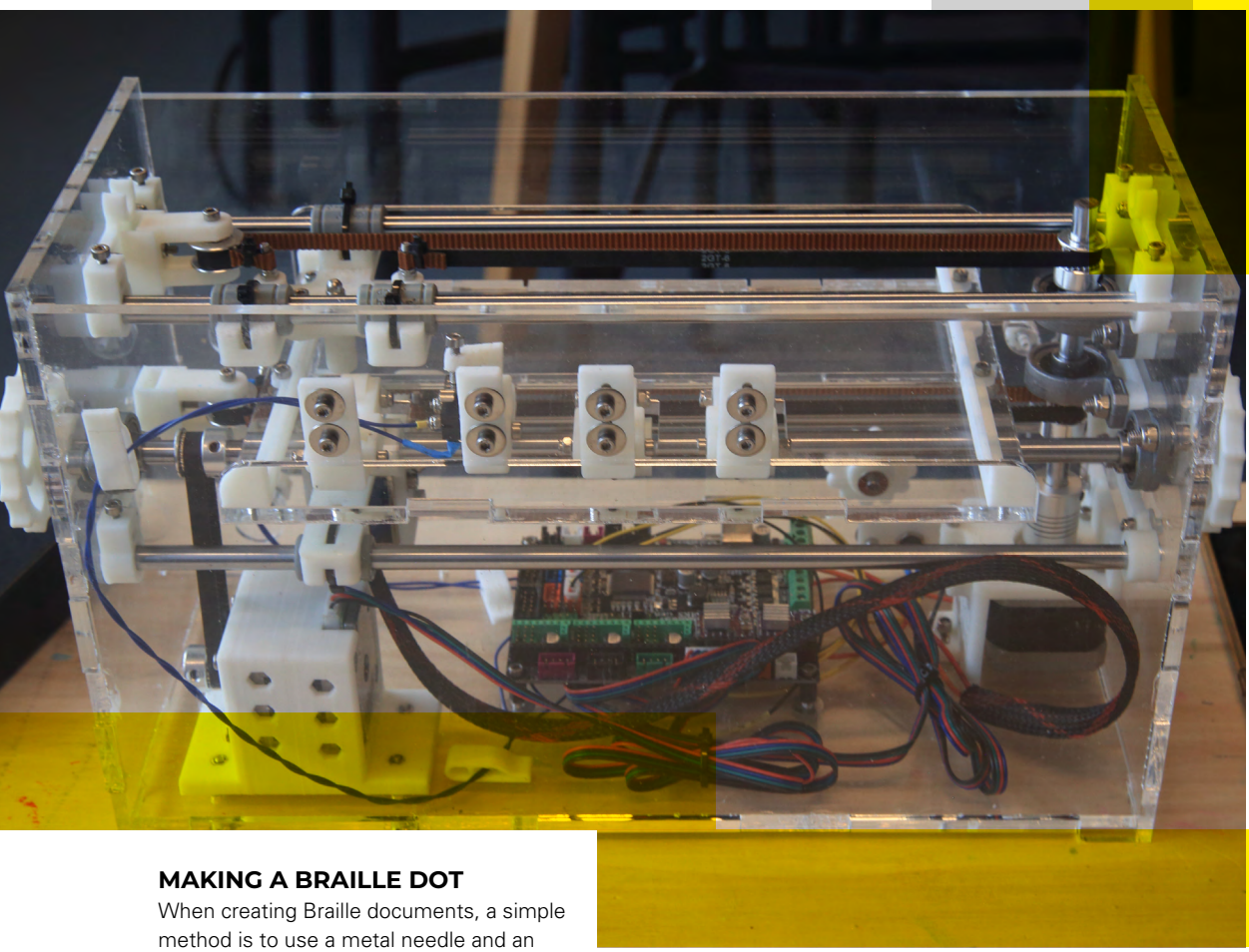
hsmag.cc/subscribe

HOW WE MADE

By **STÉPHANE GODIN**

AN OPEN-SOURCE BRAILLE EMBOSSER

Imagined in 1829 by Louis Braille, Braille is a tactile alphabet for blind and partially sighted people. Made of a grid of six dots, each Braille cell corresponds to a letter or an escape code to specify a capital, number, or punctuation mark. As each cell consists of six dots, there are 64 combinations available. Due to the limited number of combinations, each country has adopted its own Braille standard. The usage of computers and information technology shows the need for a better standard with more combinations. Since the start of the 21st century, we also have eight-dot Braille standards. At this time, eight-dot standards are used only on numerical devices, such as Braille displays.



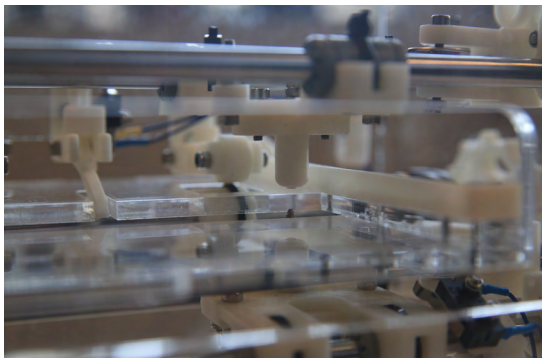
Left
Clear acrylic lets you see what's going on inside

MAKING A BRAILLE DOT

When creating Braille documents, a simple method is to use a metal needle and an elastic material, such as a mouse pad, placed behind the paper. You create the Braille dots by pressing the needle gently onto the paper, causing it to deform. However, the dot will have a blurred edge. We started this project by using an anvil, and found that using a 2 mm hole in a flat 3D-printed part produced better results. This method created a dot with a nice relief over the paper and a sharp edge. Unfortunately, the 3D-printed parts proved to be unreliable, as the material fell apart after a few dots. After testing, we found that the anvil must be made of metal. We also found that the hex hole of an M3 grub screw is perfect for our needs. As a result, all of our designs are now made with a 30 mm grub screw. The hex hole serves as the anvil shape, and we glue and lock a hex nut onto the other end to adjust the depth of the Braille dot.

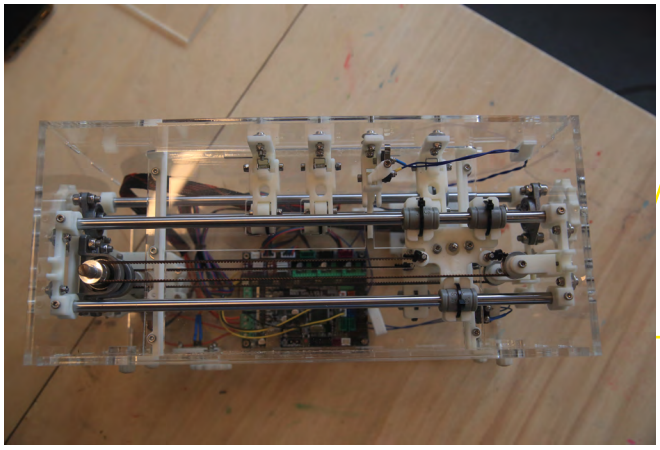
PLANNED DESIGN

Our plan was to create a standard 2D board with a Braille needle on a carriage, building on previous tests conducted on modified CNC or 3D printers. Since Braille dots are always at the same depth, we don't require precise control of the tool depth. Therefore, a simple solenoid is sufficient to manage the up and down positions. A solenoid is much faster than a stepper motor or servo. Just think about a standard CNC with a Braille needle on a solenoid as a tool. As we've seen, a good Braille dot requires a needle and an anvil. Therefore, our Braille tool consists of two parts. The paper is placed between two carriages, with the bottom one using a solenoid to punch dots while the other acts as the anvil. The paper is fed with stepper-controlled rollers to control the position.



Above
3D printing means other people can build this machine

FEATURE



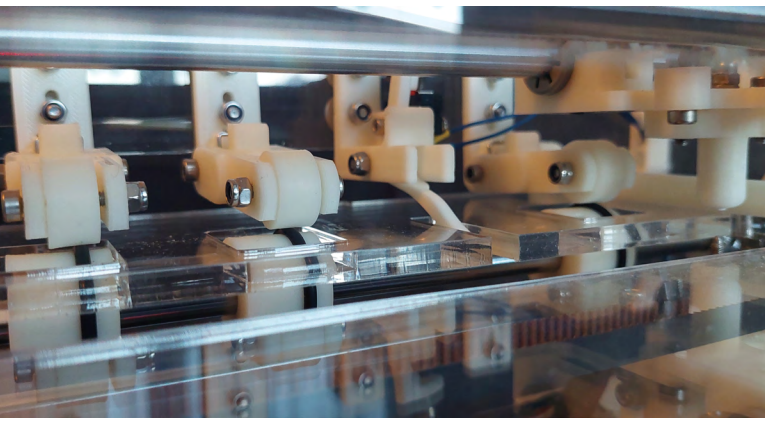
Above
The machine is small enough to fit easily on a desk

PAPER FEED

When handling the paper, the sheet must be securely held between the two rollers. The bottom roller is driven by the Y motor, while the top roller applies the required pressure. If the sheet slips on the rollers, it can lead to inaccurate printing or even cause a paper jam.

We created our first prototype using three rollers and paper pressers. The pressers were all situated on the same 8mm linear rod axis, with two springs at each end to ensure good pressure on the paper. Although this system worked adequately, it was difficult to build and adjust. So, we decided to experiment with three independent pressers and relied on the plastic's elasticity to maintain the pressure. However, the paper feeding system was still tricky to adjust and was not very reliable.

Below
It was hard to get the paper feed to work well, but we're happy with our solution



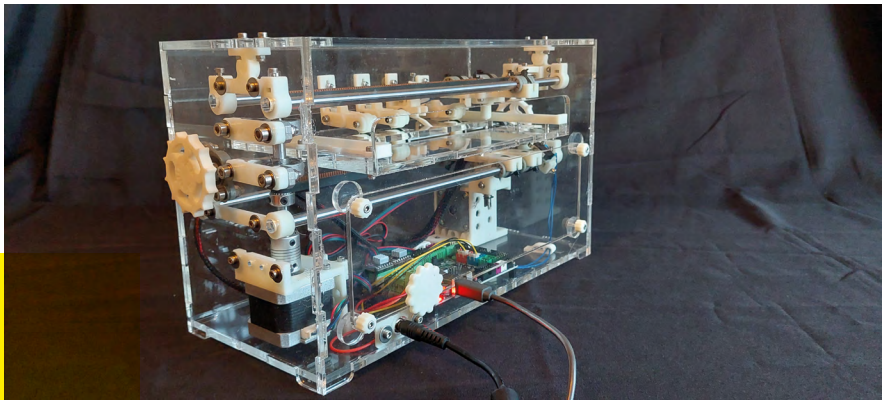
We then decided to try a simpler system using only two rollers, as it is easier to align two points. We had what we thought was a good design that was simple to build and reliable. We made a few prototypes using this design and had some success. However, with only two rollers, you can't use materials like aluminium cans.

We then attempted to upgrade the paper presser by adding springs, one for each presser. We found the GT2 tensioner spring was both affordable and easy to find. So, we redesigned the presser with two plastic parts articulated around an M3 screw, using a GT2 tensioner as a spring. Success! This new presser is more flexible and applies firmer pressure on the paper, resulting in a better paper roll that securely holds the sheets of paper.

DETECTING THE EDGE

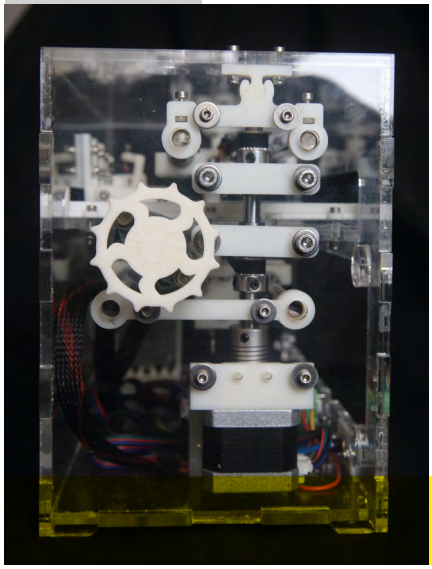
On our board, we need a reference to define an absolute X and Y position on the paper. On the X axis, there is a standard end switch that detects the minimum position of the bottom carriage and defines it as the 0 position for X. The Y axis runs along the paper sheet, so we must detect the start of the paper. Initially, we used a standard end switch with the end switch lever in contact with the paper to detect the top edge of the sheet, defining the top edge as the 0 position for Y. It worked fine, but it was difficult to set up – the paper jammed if it was too close to the end switch and was often undetected if too far. To improve

"WE HAD WHAT WE THOUGHT WAS A GOOD DESIGN THAT WAS SIMPLE TO BUILD AND RELIABLE"



Left
The complete machine ready to run

Below
Maybe I'm biased, but I think this looks good as well as being functional



the design, we used a 3D-printed lever to offset the end switch and a hex screw for fine-tuning the detection position.

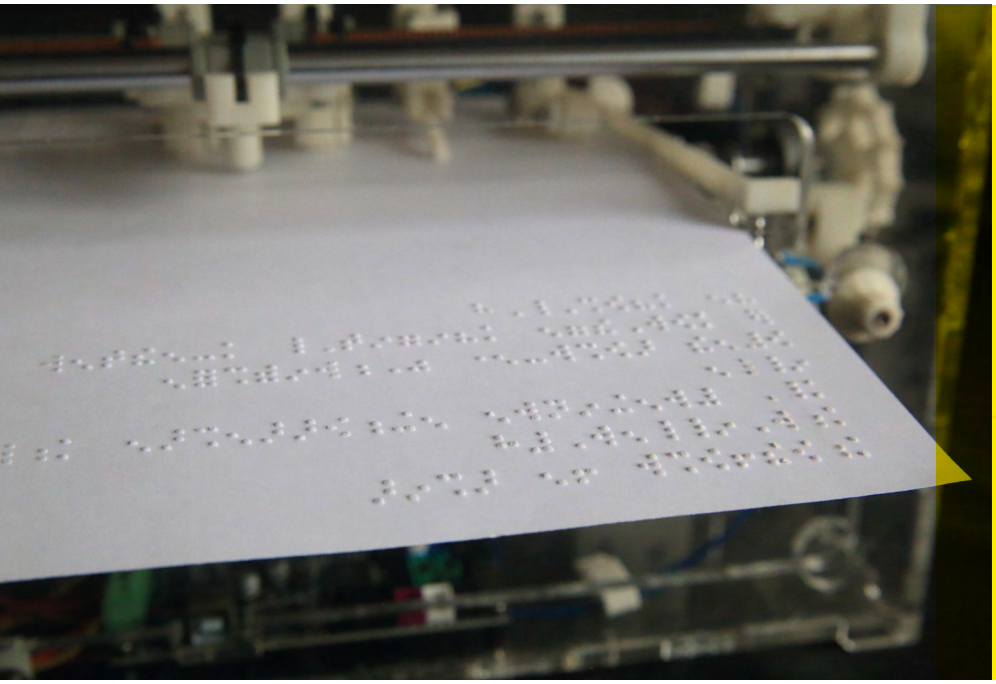
PAPER PATH

Another important aspect of the paper feeding process is the path that the material takes within the device. It's a delicate process, and even the slightest obstacle can cause a paper jam. Initially, we used a plywood or PMMA plate with a 10 mm path for the Braille needle in the centre. However, this setup often resulted in paper jams. As the paper travelled downwards, sometimes one of its corners got caught on the edge of the needle path, causing a jam. To overcome this problem, we experimented with various 3D-printed designs to keep the paper securely on the tray. However, these designs required post-processing, and we found that it's better to let a piece of paper go its own way if it's sticking in the printer. We can lower the tray on the output side, just after the needle's path, and it works incredibly well! The lesson we learned is to never force a piece of paper and instead let it travel through freely.

SOLENOID ALIGNMENT

One recurring issue we encountered was the Braille needle would fail to lower when the solenoid was activated and deactivated, resulting in paper jams or missing Braille dots. The bottom carriage proved the most challenging part to create. The solenoid axis needs to move freely, so we designed a 3D-printed part that acted as a guide for the needle. Though everything seemed to work fine when we tested the Braille needle manually, we found that the needle would often lock in the top position when we used the electromagnet with power. As we were using a cheap electromagnet, we noticed that the position of tapped holes on the side of the solenoid was not accurately aligned with the solenoid axis. In some solenoids, there is a slight angle between the axis and the anvil. This can cause a lateral force on the axis when the needle enters the anvil, which is enough to lock the solenoid in the activated position. We found that the best solution was to make oblong holes for the solenoid fixing screws – this allowed us to finely adjust the solenoid axis with the anvil.

OW



HOW

A WORD ABOUT SOFTWARE

We now have a device that enables us to move a Braille needle along the X axis using a stepper motor, and we can move the paper along the Y axis using another stepper motor. This is perfect for a 3D printer controller board with Marlin firmware. We modified the Marlin firmware to slightly adjust the Y homing. If the end switch is on, detecting the paper, we move the paper backward until the switch is off. If the end switch is off, we move the paper forward until the switch is on. This special homing feature allows the user to easily load the paper into the rollers, and the 0 Y position is automatically set at the top edge of the sheet.

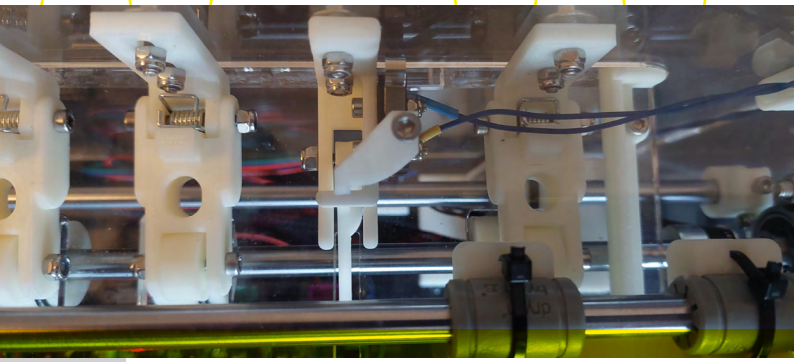
The power output designed for the heatbed is used for the solenoid control. Initially, we used the G-code M3 command to enable and disable the solenoid. This command is typically used for laser control, where M3 S1 activates the laser, and M3 S0 deactivates it. We used the command in the same way to control the solenoid. However, solenoids

Left
The dots left by the machine are easy for people to feel with their fingers

are different from lasers; you can't activate them for too long. Without a cooling system, you must respect an activation/deactivation ratio, which was low with the cheap, powerful solenoid that we used. Initially, we relied on the activation/deactivation command in the G-code file, but users reported burning the solenoid, usually by leaving the software without safeguards. To address this, we made another change in the Marlin firmware. The M3 S1 command still activates the solenoid but only for 50 ms. This is long enough to emboss a dot and ensures that the solenoid is never activated longer than 50 ms.

We now have a G-code-programmable device to emboss Braille dots anywhere on a sheet of paper. To emboss a document in Braille, you can use AccessBrailleRAP. This software translates text into Braille characters, calculates the position of Braille dots on the paper, and then sends the

W WE MA



Left
Paper can be one of the hardest materials to work with because of its tendency to jam and crease

MORE INFO ABOUT BRAILLERAP

BrailleRAP website: **hsmag.cc/BrailleRAP**
Assembling guide: **hsmag.cc/BrailleRAPdoc**

G-code sequence to the embosser. However, you still need to translate the text into Braille. As we discussed earlier, there are different standards for Braille across the world, including variations between the UK and the US. As French lecturers, we started with basic software to translate text into Braille. However, we realised that there were many complex nuances involved in the process. For instance, marking a number is not the same as marking a group of numbers, and marking a capital letter is not the same as marking an entire word in capital letters. These details require careful attention to ensure an accurate translation. While searching online for an open-source Braille transcription library, we found Liblouis. It is a grammar interpreter that enables you to produce Braille translations using text files, removing the need to write code for a Braille translator. As Liblouis is open-source and publicly available, many contributors have provided data files for their own languages. Currently, Liblouis can translate Braille into more than 200 Braille standards. All the major languages of the world are supported, including and some regional ones. So, we stopped trying to code Braille transcription, used Liblouis, and focussed on providing accessible software to use BrailleRAP.

"THERE ARE DIFFERENT STANDARDS FOR BRAILLE ACROSS THE WORLD"

HackSpace magazine meets...

The Sanjay Mortimer Foundation

Honouring the memory of a 3D printing pioneer

Even if you haven't heard of him, if you're into 3D printing you've almost certainly used something created by Sanjay Mortimer, or else have used products based on his ideas. Sanjay was a co-founder of E3D, manufacturer of 3D printer parts, and the decisions he took at the start of his career in 3D printing literally shaped the industry – the standard nozzle for FDM printers is 0.4 mm because Sanjay had that size drill bit in his toolbox at the time, for example.

After Sanjay died in 2021, those who knew and loved him (more on that later) decided to honour his memory by helping neurodivergent people find their place in engineering, an area that Sanjay loved. Since then, the Sanjay Mortimer Foundation has worked with E3D to put on the Sanjay Mortimer RepRap Festival, and has started to fulfil its mission to give neurodivergent young people a leg up into the world of STEM.

We spoke to Teula Bradshaw, chief executive of the Sanjay Mortimer Foundation (**sanjaymortimerfoundation.org**), to find out more about their mission, what they're doing to help more neurodivergent young people into engineering, and the legacy that Sanjay left behind.

WHAT IS **NEURODIVERGENCE?**

The term 'neurodiversity' refers to the natural variation in neurodevelopment across all individuals. We're all different. 'Neurodivergent' describes a person whose neurodevelopment falls outside of the 'typical' range, including but not limited to neurodevelopmental differences such as autism, attention deficit hyperactivity disorder (ADHD), dyslexia, dyspraxia, and Tourette's syndrome, among others.

Various academic studies have shown that neurodivergence is a massive hindrance within the traditional educational setup. The Sanjay Mortimer Foundation (SMF) has sifted through loads of research, most of it from US academia. Kent (2011) showed that high-school students with ADHD have significantly lower grades, and are over eight times more likely to drop out than their peers without ADHD. And according to Berkeley (2010), only 9–20% of students with ADHD graduate from college, compared to 68% of neurotypical thinkers.

High dropout numbers consequently means that neurodivergent thinkers are under-represented at all levels of higher education. And in the UK, the Office of National Statistics has found evidence that neurodiverse people are massively more likely to be unemployed, and that half of those who are employed are under-employed, meaning that they have skills that are being wasted. Something small like access to a bit of kit or training, or a mentorship, can make a huge difference to someone's life.

The SMF is looking for applications from people aged 14–25 in England and Wales who are neurodivergent and who love engineering. It could be 3D printing, electronics, robotics – anything, as long as it fits under the STEM umbrella. They're well aware that mental health diagnoses take time, so you don't need to be officially diagnosed – as long as you're on the waiting list, you're eligible to apply. Go to **sanjaymortimerfoundation.org** to find out more.

INTERVIEW

HackSpace Hi Teula! You're the chief executive of the Sanjay Mortimer Foundation, which begs the question: who was Sanjay Mortimer?

Teula Bradshaw Sanjay was a teacher. He found his love of making through his school and university years. And he also just loved teaching generally – even if you knew him in his later years, when he wasn't a teacher, he was always listening to people. He was always interested in what they were working on; he wanted to help people with their innovation, with their ideas, with their plans. And he always had that really nurturing vibe.

He always had time for people. It didn't matter how busy he was, if somebody had some kind of crazy idea, he would sit down and give them that time to explore it further. So naturally, he was a teacher and a listener.

His friend David Lamb was also a teacher, and they loved 3D printing and messing around with 3D printers. They developed this unique hot end, the V4. They had a go at putting it on the market, it sold out within seconds, and the idea of E3D was born.

The E in E3D stood for education; originally the plan was to try and get 3D printers into schools; they both felt, being teachers in the design and technology field, that there were just not enough 3D printers in schools. Then it just blew off the shelves, it went crazy.

They set up the company, and quickly realised they needed their good friend Josh Rowley, who specialised in engineering and coding, to join them. The three of them set up E3D, and it's been the leading hot-end company around the world for many years now.

A lot of the innovation came from Dave and Josh, but it was Sanjay's baby. His job title at E3D was 'Visionary' – that was his role.

HS I sincerely hope he had that on his business cards.

TB Absolutely! It was on his business card; it was on his email signature… his role was visionary, which just says it all. He thought outside the box. He thought differently to a lot of people; he knew what customers wanted before they knew what they wanted. And he was always ten steps ahead, which really fuelled E3D's growth.

Most of E3D's customers were around the world, particularly the States. And so people knew of him; he would talk at events, and would be on Discord chatting to people all the time, discussing innovations and ideas.

He was such an open person. And that's why he has so much love and respect in the community. He was passionate, shared his ideas, and would listen and help in any way he possibly could.

> "He was such an open person. And that's why he has so much love and respect in the community"

HS How did his neurodivergence affect Sanjay?

TB He had this chaotic, crazy personality, which was lovable – Sanjay was so, so lovable.

But it was also very difficult to work with somebody like that at times. Sanjay wouldn't know when he was supposed to be in a meeting or who he was meeting with. He would come in, in a sort of whirlwind of drama. And he'd put his laptop here and his bag there and his mobile phone there, and he's whizzing around talking to all the engineers and coming up with ideas. We'd have a Teams group and there was a constant theme on the chat: Where is Sanjay? Has anyone seen him? Has anyone seen Sanjay's laptop? Which rabbit hole has he dived down into now?

As funny as that was, it was quite hard to help him structure his day. That's where I came into Sanjay's world; I was brought in to literally shadow him and be in charge of what he was doing, when he was doing it, and who he was doing it with, just to help bring some structure to his day.

That was an amazing role to fill, because it was filled with madness, but a lot of fun. And hopefully, it really helped. And it meant that he was much more effective with his hours and his days.

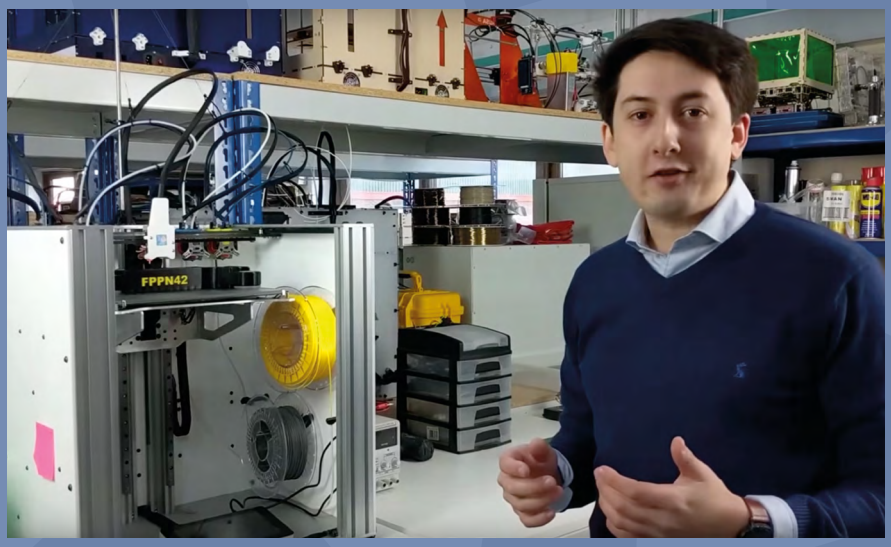
But that was in his later years; as a kid he felt like he was up against it. He didn't sit very well in a traditional educational setting. A lot of neurodivergent people respond really well to what's called strength-based learning, which is when they're learning a subject that they're really interested in. Subjects that they're not interested in, their brain goes off all over the place, they become agitated or fidgety, and potentially get labelled the naughty child at school.

He got that a lot, because he found things slow or boring. And so it wasn't until later that he was allowed to really delve deeper into his making and his tinkering and doing what he really found that he was passionate about.

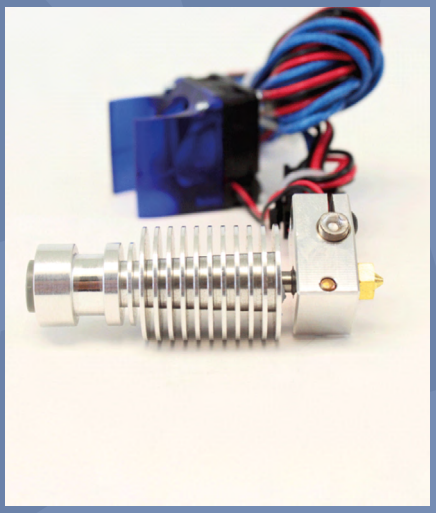
HS And this brings us on to what the foundation is all about.

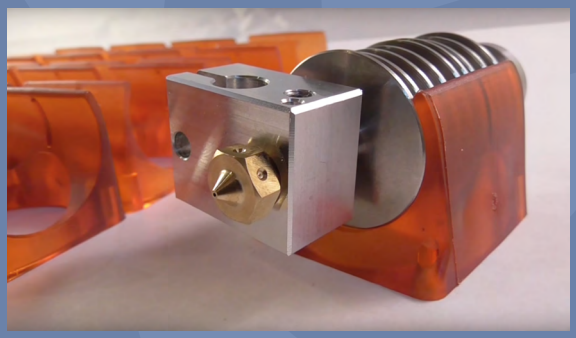
TB It really is about that. It's about giving our Stars the confidence and the self-esteem that they need to recognise that they can go on to thrive in their chosen field.

Often, young neurodiverse individuals have had a number of knock-backs – they've been told, "don't do that", "don't do it like that", or they've just been →



Left
3D printers used to be rubbish. The fact that they are not rubbish anymore is in large part thanks to Sanjay Mortimer





Below
The inaugural Sanjay Mortimer RepRap Festival, held in 2023, was a riot of noise, colour, and sharing. This year's will be in sunny Manchester, and should be an unmissable event for fans of 3D printing

made to feel different to their peers. The Foundation is about celebrating those differences, and helping them to realise that that's OK. They can harness what it is that they love doing, focus on that strength-based learning. The foundation is all about increasing the education in a field that they're really interested in.

We focus on STEM – science, technology, engineering, and mathematics. That's the area that we're in, because there are other charities out there that focus on arts and drama and theatre. We just thought, let's focus on what Sanjay was all about.

HS What sort of form does that take? I'm guessing that you can't give everybody a right-hand man, so you need a slightly more scalable solution than that?

TB Our application process is open at the moment. To apply, you need to be between the ages of 14 and 25. We chose that age range because we had to choose an age range; we'd be spread too thinly if we decided to help absolutely everybody. Below the age of 14 you're still learning all the academic subjects. It's not until the age of around 14, 15, 16 that you start noticing where your strengths lie and what you enjoy.

And then 25 is the cut-off because we felt by that time – and this is a very generalised view – you will have manifested some coping mechanisms. And you're a little bit more established, you've hopefully had some support that you need. That's not always the case, but we had to have a cut-off somewhere and, hopefully, by then you're in employment or you have the support to try and get you there.

We're a charity registered in England and Wales. So you have to be resident in England and Wales. We want to go global, we really, really do – a lot of our support is from abroad. But in the initial stages, the Charity Commission is very strict about regulating a charity, and if it's abroad and much bigger, it makes it very hard for them to regulate us. So the advice I was given was to set up as a UK charity to begin with, establish ourselves, get a few years under our belt, and make sure you're not a big red flag to the Commission. Once you're a well-known entity, you can spread your wings further.

The support we offer is completely bespoke and unique to that Star – we refer to them as our Stars or SMF Stars. So a Star will apply, and they will describe to us the challenges they're having and a particular area of interest. It might be IT; it might be Formula One, you know, any kind of engineering that they're particularly interested in. And then we work with our collaborators who are generally their teacher/tutor, could be their employer, or it could be the maker space owner, somebody who knows them – to come up with the best support possible.

That might be – as an example – funding them through further education. Through a training course, or a degree course. It might be providing them with a resource or a piece of equipment that they can use to further their education in the field that they're interested in. It might be setting them up with work experience or an internship with a company that they would really love to get a foot in the door with.

We've got some mentors on our books, ex-engineers who have got time to mentor these kids. And everyone's different. Our last Star, who is attending Brunel University, has a form of rare neurological fits. She's developing a self-inflating airbag which protects her from injury when she fits.

There are similar devices on the market like this, but they are really expensive. One was I think, in Sweden, a device that you wear around your neck when you're cycling, and if you fall, it inflates. And the other one is for old people who wear a vest – and if they fall, it blows up. She was desperate to get hold of these two pieces of equipment just to understand and take them apart and research them. The SMF bought these for her so that she could further her research and development on them for her own product.

Another of our Stars received 3D printers, work experience, and a laptop with relevant software to further his education in his chosen field. We are extremely grateful to our Partners, Prusa, E3D, Ask4Support, SolidSolutions and BrendonBuilds for helping us support him.

HS From your website, I learned that people with neurodivergent minds are highly under-represented in engineering degrees. That's a surprise.

TB It is. The evidence shows that neurodivergent people get lost before they make it into engineering careers. They fall out of the system. And this is based on the fact that they have a tough time at school; they feel like they can't achieve, they feel like they're failing. And it's because they haven't been taught in this strength-based fashion. And so they drop out of the system, they don't make it to senior school, or they don't make it to university, and therefore they absolutely don't make it into a career in engineering. That's where that statement comes from.

Another stat is that autistic people have one of the worst employment rates in the country, and they are paid the least. It's an awful statistic.

Applications opened at the beginning of February for this year. And my focus for this year is to try and get more Stars. Last year was very much the setting up phase. And we only got our charitable status in late November 2023. So in theory we've only been operating officially as a charity for three and a half months. Charitable status something that we really, really wanted; it brings us the benefits of Gift Aid; it brings in tax benefits for corporations that are supporting us. In order for us to operate, we need three things: one is our Stars, the people →

we support; two are our Collaborators – they are the teachers, the universities, the employers that are going to help us find the Stars.

Three are our partners; they are people that might do some fundraising initiatives on our behalf. Or they might have us as part of their corporate social responsibility and support us. Or they might be people who provide work experience opportunities or internship opportunities, or provide resources; they might donate printers or whatever it might be. So these are our three focus areas: the Stars, the Collaborators, and the Partnerships that make the whole circle of what we are and how we operate.

HS Last year saw the inaugural Sanjay Mortimer RepRap Festival, held at the University of Oxford. Will that be going ahead this year?

TB I can confirm that we will be holding it in partnership with Manchester Metropolitan University and PrintCity, and we will be holding it in their business school. It's an amazing space. As beautiful as Oxford was, it had some limitations: accessibility was a major limitation. It was all on the second floor, which was logistically a nightmare, because everybody had to cart printers and equipment upstairs.

HS I saw a video of someone carrying a printer the size of a small car up an ornate flight of stairs. It looked like hard work.

TB That's the Voron Phoenix, which is a truly unique printer. Yes, we've taken into account accessibility this year! It's an incredible, modern, beautiful building that's at the heart of innovation. We're being supported by PrintCity, which is Manchester Metropolitan University's 3D printing hub. And yeah, it's going to be a great partnership. I'm doing much more work this year to make it more inclusive, more neurodivergent-friendly. We recognise that it's a noisy event, but we're going to be providing some quiet rooms that you can escape into, and maybe get some involvement from some headphone people, and we'll behaving the speakers in a separate room.

The feedback from last year was that it was just such an incredibly open community and such a lovely friendly vibe. And that's what we very much want to focus on again. There'll be more focus on the SMF Stars, because by just attending, you're supporting them and the charity.

> "It's an incredible, modern, beautiful building that's at the heart of innovation"

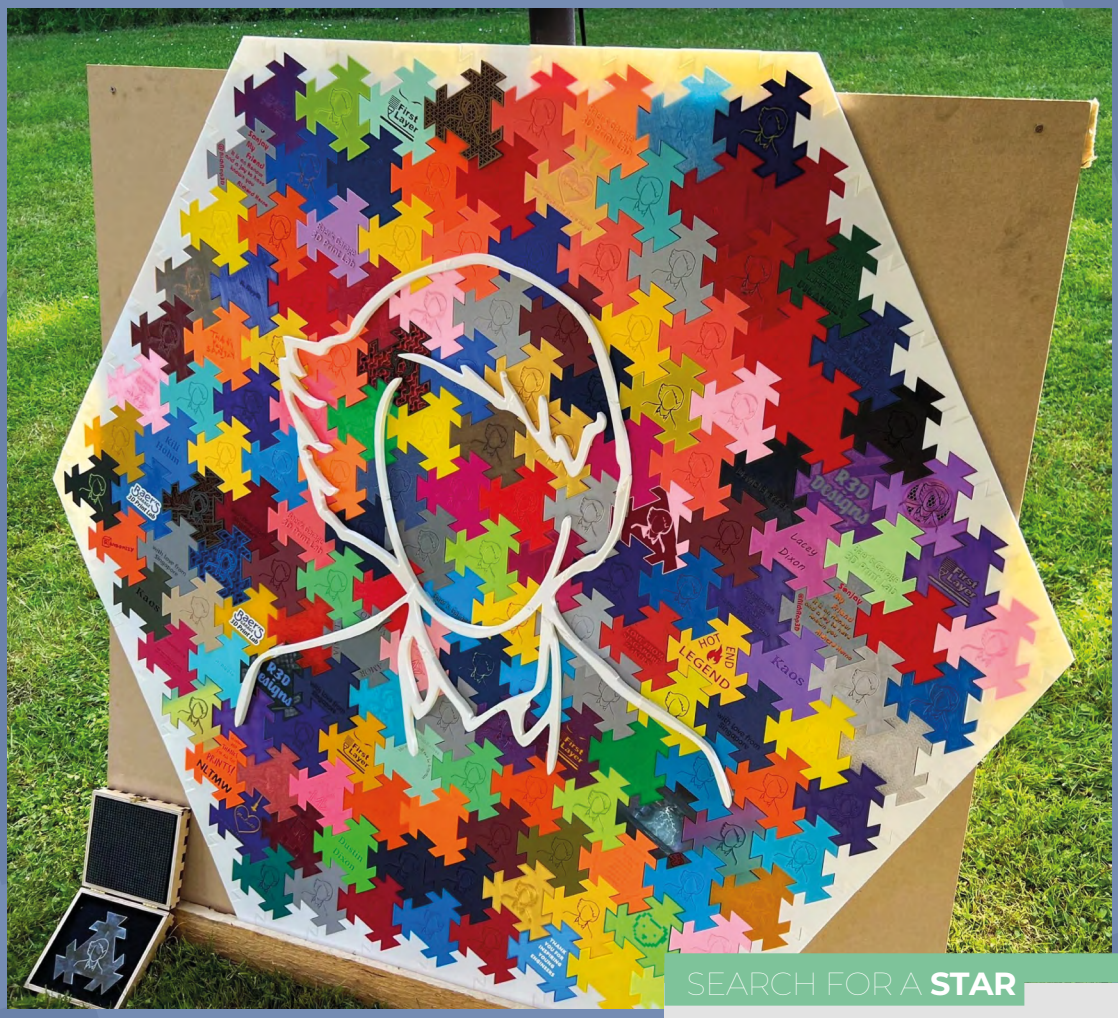
HS I bet you've had a whole load of support from within the 3D printing community. Everyone is just so nice.

TB That's where the majority of our support comes from at the moment, and I am in awe of the community's generosity. On a daily basis I hear about another fundraising initiative – somebody's selling T shirts and the proceeds are coming to the SMF, or Zack Freedman's fabulous Sanjay video, where he's offered to donate royalties to us. It's heartbreaking, but it's fabulous. The plan is to expand to incorporate other engineering areas. But at the moment, because of who Sanjay was, the 3D printing community is where we have most of our support.

HS Who came up with the Smurf pronunciation for SMRRF? That's genius!

TB That's credited to Chris Pelesky, who co-organises the East Coast RepRap Festival. That's ERRF [pronounced 'erf']. Then there's the Rocky Mountains RepRap Festival, which is RMRRF, and now the Sanjay Mortimer RepRap Festival is SMRRF.

We were so tempted to have little blue men all over the venue last year, but we resisted the urge! What I love is that Sanjay would just be giggling. I think he would be seriously pleased and proud of the impact we're having. It's in celebration of him and what he's done for the industry. I hope he's pleased. □



SEARCH FOR A **STAR**

If you know someone who fits the Foundation's eligibility criteria – aged 14–25, resident in England and Wales, neurodivergent, loves STEM subjects – and you think they could benefit from access to equipment, or funding to further their education, gently nudge them toward applying for support from the Sanjay Mortimer Foundation. And remember, they don't have to be fully diagnosed as neurodivergent: it's enough to be on the waiting list.

sanjaymortimerfoundation.org

Arts & Craft

With Raspberry Pi

Technology and creativity are inclusive and Raspberry Pi brings the two together.

By David

THE MAGPI

This feature is from 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**

Many Raspberry Pi projects offer solutions to practical problems. Many builds also perform specific tasks that are either informative or seek to help make life a little easier. Equally, however, there are loads of makers unleashing their creativity and putting Raspberry Pi at the heart of imaginative endeavours: simply because they can.

These makers are producing appealing projects that stop people in their tracks because they either look good or perform tasks that are just sheer, plain fun. These are our kind of makers.

In this feature, we take a look at arts and crafts projects that inspire us. In some cases, they're straightforward but they will hopefully unlock the inventive part of your brain so that you come up with more ingenious ways of taking projects further. In other cases, a project may be difficult to replicate like-for-like, but that doesn't mean you can't be flexible and pursue an alternative path. You may even decide to mash up two or more projects.

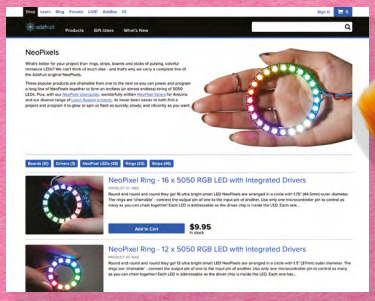
Whatever you do: be bold and innovative. And remember, with Raspberry Pi powering your projects, you already have a head start. The possibilities are endless.

WHERE TO BUY KIT

PISHOP.US

▶ pishop.us

Light up your projects with NeoPixels, screens, and LEDs. You'll find loads on sale at PiShop so take your pick.



THE PI HUT

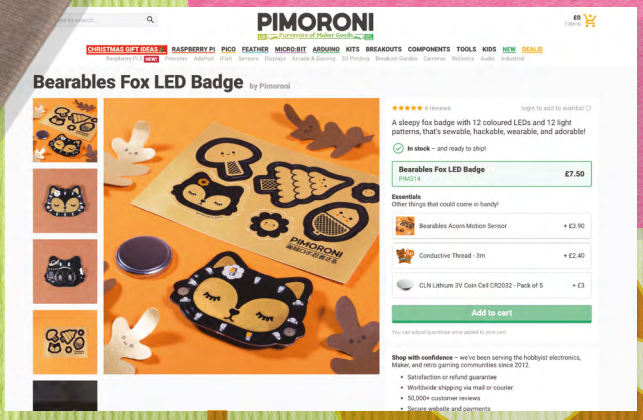
▶ thepihut.com

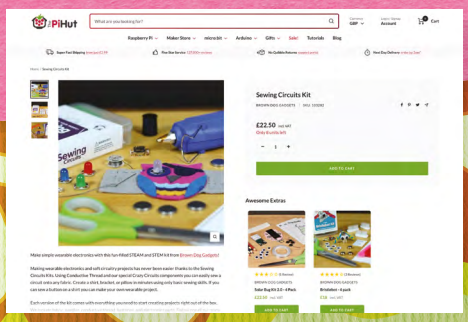
Into making wearables? Snap up conductive thread, woven conductive fabric and more from The Pi Hut.

PIMORONI

▶ pimoroni.com

Find great creative kits such as the Bearables Fox LED Badges and LED circuit stickers at Pimoroni.





Crookes

Maker Motivation

Inspiring projects that will help you to make music, clothes, furniture, and a big impact on other makers!

DIFFICULTY GUIDE

BEGINNER
Projects for everyone

ADVANCED
Projects that require some experience

EXPERT
Projects that require skill

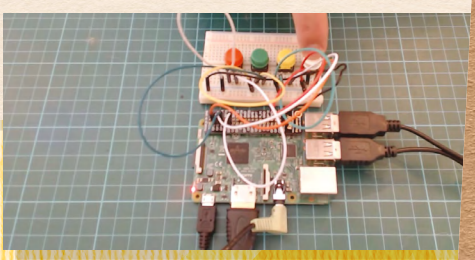
GPIO MUSIC BOX

▶ magpi.cc/gpiomusicbox

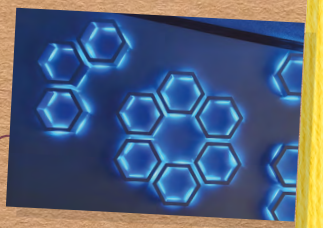
Don't be afraid to make a noise about your arts and crafts project. In fact: if you try your hand at making this GPIO music box, that's exactly what you'll be doing. Connect a set of buttons to a Raspberry Pi computer and you can begin to play sounds using Python. The more buttons you add, the more sounds you can enjoy, potentially turning the project into a fully-fledged instrument. Either that or a sound box that can make rude noises on demand. The choice is yours.

TOOLS NEEDED

- Raspberry Pi
- Breadboard
- Four tactile switches
- Five pin-to-socket jumper leads
- Four pin-to-pin jumper leads
- Speakers/ headphones



HONEYCOMB WALL SHELF



▶ magpi.cc/112

Fancy creating an artistic talking point for your home without going over the top? These honeycomb wall shelves may do the trick. Adopting a hexagonal design, the shelves were designed with a channel, through which 12V RBG light strips could be threaded. Once outputted to a Crealty Ender 5 Pro 3D printer, the shelves could then be connected up to a Raspberry Pi 4 computer from which they are controlled, allowing different levels of red, green or blue to be displayed.

TOOLS NEEDED

- Raspberry Pi 4
- 3D printer
- Four tactile switches
- RGB light strips
- Three IRFZ44N MOSFETs
- Three NPN transistors

RASPBERRY PI BERET

magpi.cc/raspberrypiberet

If you want to get ahead in fashion then you need to get a hat, and this Prince-inspired headwear is sure to inspire. As well as allowing the ring of colourful LED lights to animate at random intervals, you can connect a phone and control the lights from a web browser thanks to Raspberry Pi Pico W's wireless LAN. Any hat would work for this project (and you could even decide to make your own). Just make sure there's enough room above your head for the several feet of wire required for the project.

TOOLS NEEDED

- Raspberry Pi Pico
- NeoPixel LEDs
- MicroPython

Art Advice!

You could turn Raspberry Pi into a live coding music studio using Sonic Pi (sonic-pi.net)

SCREEN DRESS

magpi.cc/screendress

Let's be honest here, creating something that could be worn on the catwalk is certainly no cakewalk. But a Dutch FashionTech designer has shown what is possible if you let your imagination run wild. Her ScreenDress makes use of Hyperpixel 2.1 Round LCD screens costing around £50 each, and they've been designed to look like eyes which change according to the wearer's thoughts (gathered by an EEG sensor strapped to the head). They show how you can bring tech together in a creative way.

TOOLS NEEDED

- Raspberry Pi Zero 2 W
- Hyperpixel 2.1 Round LCD screens
- Unicorn Headband
- 3D printer

SPEECH-TO-TEXT BADGE

magpi.cc/speechtotextbadge

So, you're at a party or a conference, and the person you're talking to can't hear anything you're saying. Rather than resort to pointing, you can make yourself a badge that picks up your words via a USB lapel mic, processes them on a Raspberry Pi Zero with the deep learning speech recognition system Deepgram, and transcribes your speech on a small screen.

TOOLS NEEDED

- Raspberry Pi Zero 2 W
- Touchscreen display
- Mini HDMI – HDMI Adapter
- Battery pack and cable
- USB Microphone

FEATURE

Visual arresting projects

These projects are sure to catch the eye

MATRIX ALBUM ART DISPLAY

▶ magpi.cc/artledmatrix

As much as we love streaming services, we can't help but think we've lost the art of the album cover. With this fab project, you can bring those covers back to life by displaying them on Adafruit's LED matrix. It's a simple enough project which ties in with the music you're playing so check out the maker's GitHub page (**magpi.cc/ftartwork**) to learn how to make it.

TOOLS NEEDED

- Raspberry Pi 3B+
- Adafruit 64x64 RGB LED Matrix
- Adafruit matrix bonnet
- Power supply
- 5v 10A switching PSU
- 3D-printed matrix feet
- Pi mount

SUNRISE LAMP

▶ magpi.cc/sunriselamp

The Sunrise lamp marries technology with an origami technique called snapology. It requires a lot of patience but, if you have the time to learn how to use strips of paper to create shapes from rigid triangles, then you too could create a rather special light. A Raspberry Pi Zero W is connected to a set of NeoPixels and the lights have been programmed to dim at night before brightening in the morning to help ease sleepers from their slumber. There's lots of scope to create some amazing designs here.

TOOLS NEEDED

- Raspberry Pi Zero W
- Plastic case
- NeoPixels
- Paper

POV DISPLAY

▶ magpi.cc/povdisplay

This visually arresting project works by rotating a strip of LEDs at high speed. By syncing their blinking patterns (and working on the principle that humans continue to briefly see an image that has stopped entering the eye), it's possible to create the illusion of a moving image. The strips are connected to the Pico, mounted on a rotating arm and spun at high speed. A wireless charging module prevents the wires from becoming a tangled mess.

TOOLS NEEDED

- Raspberry Pi Pico
- Adafruit DotStar Digital LED Strip
- QTR-1A Reflectance Sensor
- DFRobot Wireless Charging Module 5V
- DC Motor RS-540SH
- Microsoft Visual Studio 2017

RUHACAM

▶ magpi.cc/ruhacam

Photographers don't need to spend a fortune on a cutting-edge brand-name camera, nor do they need to reach for the latest smartphones. With the High Quality Camera for Raspberry Pi, it's possible to produce your own device in any style you want – allowing you to flex your artistic muscles in the creation of a cool case before showcasing your flair for a good photo. This particular project opts for a retro look, but it comes complete with a modern viewfinder. As creator Penk Chen says, "it's literally technology married with liberal arts."

TOOLS NEEDED

- Raspberry Pi Zero W
- High Quality Camera
- 2.2-inch TFT display
- 2000 mAh lithium polymer battery
- Case

Art Advice!

Why not make yourself a light meter too? **magpi.cc/picolightmeter**

AI ART

▶ magpi.cc/creatingaiart

Thanks to artificial intelligence, you can enter a prompt, click generate, and see a picture appear before your eyes. If you fancy experimenting with the technology, try OnnxStream, a small inference library that runs Stable Diffusion 1.5 on a Raspberry Pi Zero 2 W. Sure, it can be slow, but it's super impressive. Check how it works and learn how to set it up at **github.com/vitoplantamura/OnnxStream**

TOOLS NEEDED

- Raspberry Pi Zero 2 W
- OnnxStream

Objet 3d'art

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

Generations of British kids got their first taste of making things through Airfix kits. These cheap, extruded plastic model kits of WWII-era vehicles brought the Hawker Hurricane, Mitsubishi Zero, Messerschmitt, Spitfire, and more into homes up and down the land for pocket money prices. Some painted them meticulously; some threw them out of windows to find out if their model aeroplanes would actually fly. They did not fly!

Stian Ervik Wahlvåg, aka agepbiz, has taken this idea and run with it in the shape of this overly complicated business card. Rather than injection moulding, Stian's used a Bambu Lab X1-Carbon 3D printer to produce a four-colour model of the parts of a toy jeep. For the packaging, he's printed a design onto card, which he's cut out using a Silhouette Portrait 2 vinyl cutter, folded with the relatively primitive but no less satisfying means of a ruler and a craft knife. He's then wrapped the box in cellophane sealed using a PFS-200 Heat Sealer, and heat-shrunk with a Bosch UniversalHeat 600.

Unlike most business cards, this doesn't have any words other than social media handles, but it does tell you that the maker has some pretty advanced skills. □

hsmag.cc/BusinessCard

Letters

ATTENTION ALL MAKERS!

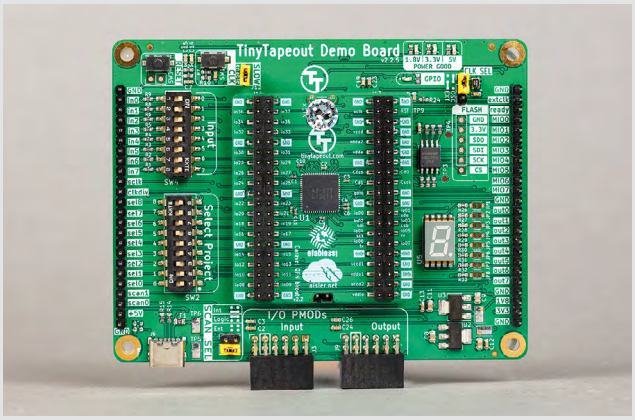
If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

ON TOPS OF THE WORLD

That robot dog in issue 77 – wow. That's a massive achievement for one person. Well done Aaed Musa. I wish I had the dedication to make something like that. Both the hardware and the software look like they must have been a real struggle to get right. Congratulations on an awesome build!

Mark
Telford

Ben says: I completely agree. It's finished off with a level of polish that puts many of us to shame.



TINY TAPEOUT

I don't normally write in to magazines, but Tiny Tapeout just knocked my socks off. I can't believe that it's not only possible for a hobbyist to design a chip (or a part thereof), but reasonably achievable for mere mortals. I can't wait to get started.

Sam
Hull

Ben says: I know, right?! I'm still a little bit in shock that my design is etched in silicon. Yes, my design is simple to the point of being trivial, but it's still made it onto a chip, and it works. I'm now planning what to do next.

AIR QUALITY

The interview in issue 77 sent me down a rabbit hole. Obviously I was aware that pollution was bad, but I didn't realise just how bad it could be on a personal level. Thanks for bringing this up. I'm going to start monitoring it at my house to see how badly I'm affected.

Jane
Stroud

Ben says: Air quality is one of those things that modern life really makes challenging, in a large number of ways. There are some things that are completely within our control – for example, using an air filter when soldering, wearing a dust mask when doing anything that causes dust, and even things as seemingly simple as cooking can put out dangerous levels of pollution in the vicinity that have to be dealt with.

There are also things at a larger level – mainly car and fossil fuel-related – that take more co-ordinated action.

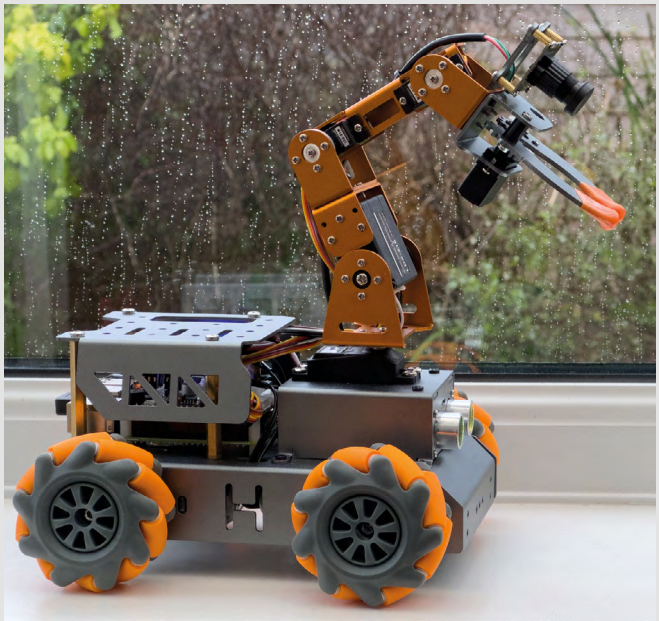
The two things that both areas have in common is that it's much easier to take effective action when you know what's going on. Michael's air droids are excellent and engaging, but your monitor doesn't have to look quite as striking to be effective.

ROS

I've played about with a few robots over the years, and always struggled when the software side started to get complex. I'm not a skilled coder – just a bit of a tinkerer – and it just seemed to get unmanageable very quickly. Rob Miles' tutorial on ROS seems like it might be the answer. I'm going to dig out my old line-following robot and see what new skills I can give it!

Ali
Sunderland

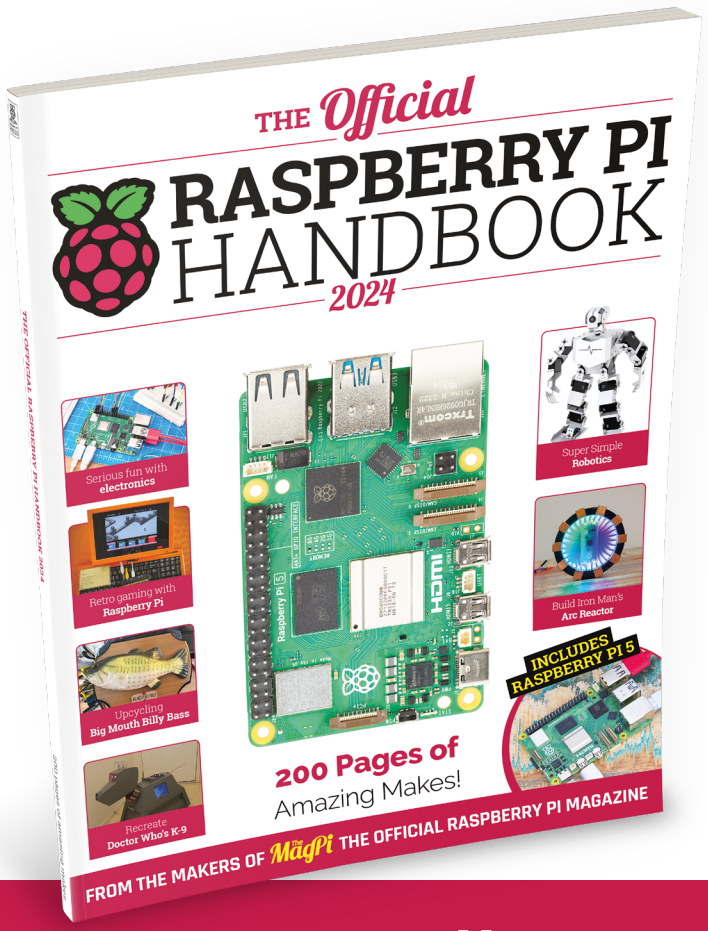
Ben says: Glad you found it useful. Software's a funny thing – so often we start a project worrying about the hardware, but end up finding that the software is the real challenge. The good news, though, is that you've got some hardware to have a hack about on. Getting clued up on ROS shouldn't be too hard, and will add a real skill to your maker armoury.

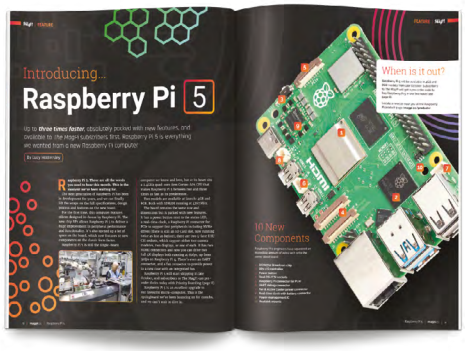


THE *Official* RASPBERRY PI HANDBOOK 2024

200 PAGES OF RASPBERRY PI

- QuickStart guide to setting up your Raspberry Pi computer
- Updated with Raspberry Pi Pico and all the latest kit
- The very best projects built by your Raspberry Pi community
- Discover incredible kit and tutorials for your projects





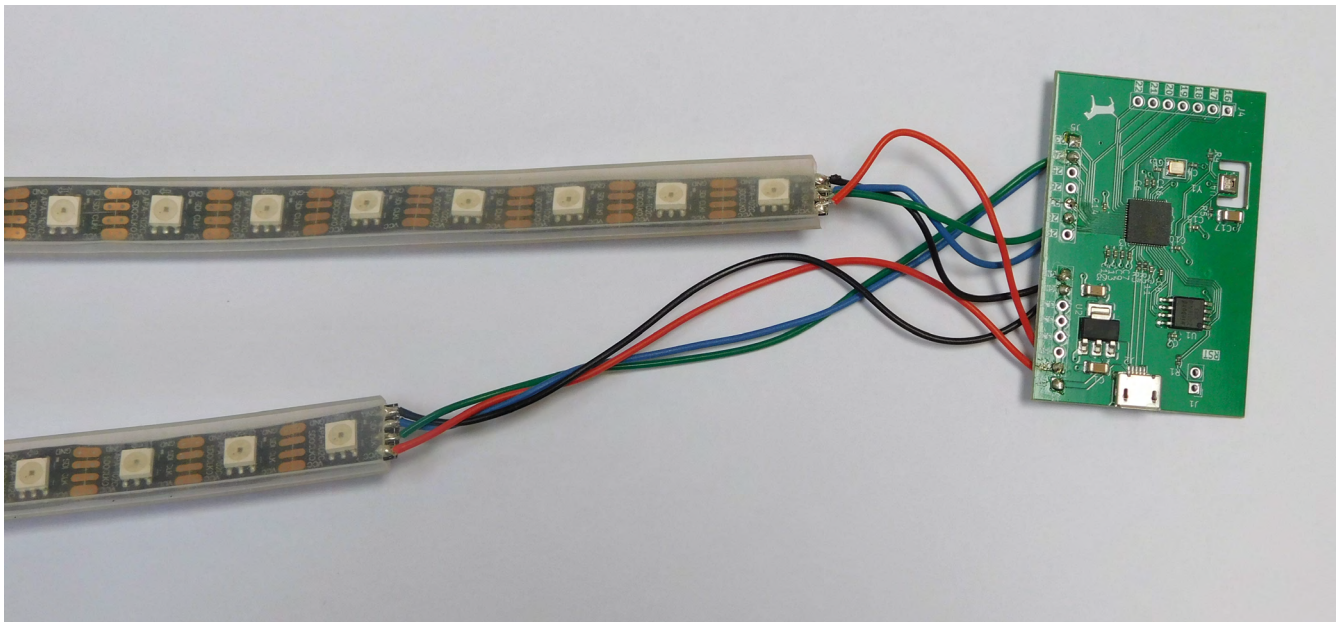
Buy online: magpi.cc/store

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

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



Making an RP2040 temperature sensor

Let's explore adapting our RP2040 layout to make a simple weather station

Jo Hinchliffe

Jo Hinchliffe (AKA Concretedog) is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

Above
The completed AHT20 board

In this final project of the KiCad series, we're going to adapt our RP2040 layout to make a rudimentary weather station. The PCB will host an AHT20 sensor, which is a good-quality sensor that can detect both temperature and humidity via I2C. It's made by the same company that created the venerable DHT22, which has long been a popular sensor for this type of project.

Scouring the JLCPCB parts library revealed that they have two versions of the AHT20 in surface-mount packages. It's important to double-check components carefully, as one of the options (the F variant) was built so that the vent hole for the sensor was placed underneath the package, so required a very accurate footprint that included a matching hole through the PCB under the footprint. With our correct package identified, we created a KiCad footprint for the 6-pin device (although two pins on the device are not connected).

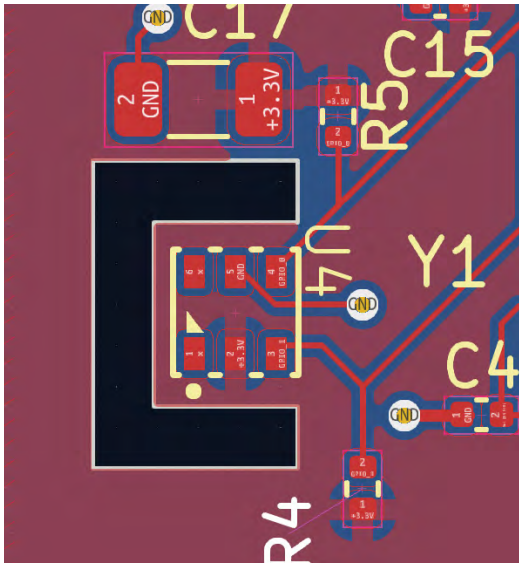
The next step is a quick read of the datasheet. It's pretty straightforward to connect the AHT20, with SDA and SCL pins having a pull-up resistor and a bypass capacitor placed near the package. So far, straightforward. The datasheet also recommends, where possible, that the package has a routed channel around it to isolate it thermally from the rest of the system. You can get away without this, but your temperature readings will be influenced by the rest of the components on the board. You can never fully thermally isolate a temperature sensor, but the more you can disconnect it from the rest of the board, the more accurate your readings will be.

DOTSTAR GPIO PINS

You can use any pins for DotStar LEDs. However, underneath the library, they communicate using the SPI protocol. If you use the pins that are enabled for the appropriate SPI functions, you'll get better performance. For a strip of 14 LEDs, the difference is negligible, if there even is any. However, if you're handling larger displays, it pays to fully utilise the specific SPI hardware on the RP2040.

Figure 1 (Below)
Adding a slot around the AHT20 sensor to thermally isolate it somewhat from the PCB

Right
Our completed PCB layout

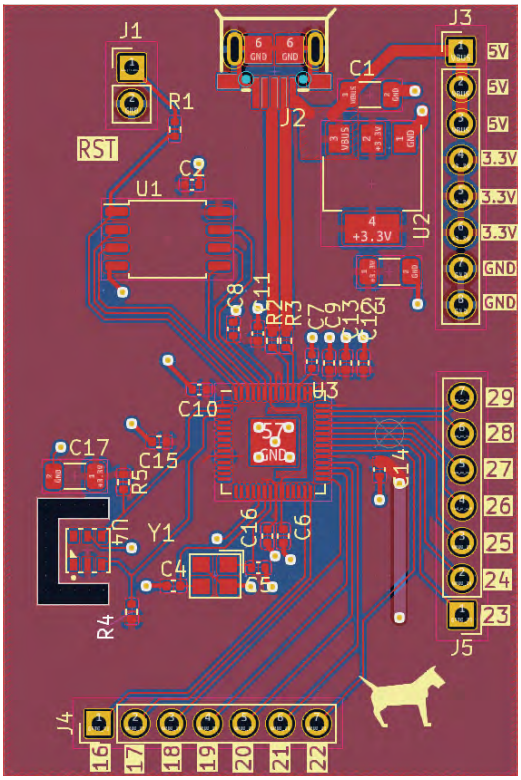


There are two ways we can create this cutout in KiCad: add the cutout in the footprint, or place the component in the PCB design and then add the cutout manually. Obviously, if you add the cutout to the footprint, you then have to edit the footprint if you want to change the cutout design. Then, if you have placed a footprint with the channel in the PCB Editor, but not created an overall edge-cut geometry, you can't correctly preview your board as the 3D viewer will consider the small channel the outside edge of the entire design.

We opted to add the channel in the PCB editor, having placed and routed the AHT20 sensor. We've often, in this series, used Inkscape externally to create and then import SVGs for graphic elements and edge-cut geometries but, for simple items like this, using the 'Draw a basic polygon' tool in KiCad is simple and convenient.

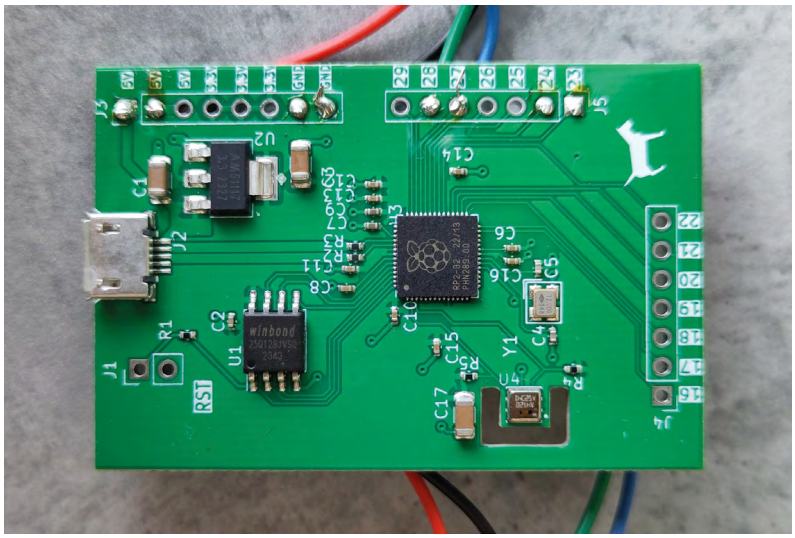
It's important that you check what are the minimum slot dimensions that your fabricator can handle, as this will be limited by the size of the bit they use for routing. At time of writing, this is 1 mm on JLCPCB, so we made our channel around 1.5mm wide (**Figure 1**).

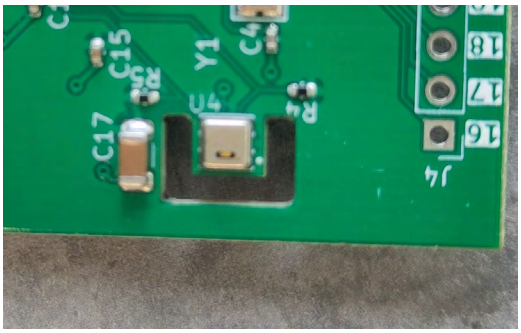
> **We wanted to be able to hook this board up to a wide range of outputs,** so we broke out a healthy number of GPIO pins



The rest of the board lay up was pretty straightforward. We wanted to be able to hook this board up to a wide range of outputs, so we broke out a healthy number of GPIO pins, and also added a small pin header with numerous power connections.

After the usual session of silkscreen labelling and tidying, we then went through the familiar process of creating Gerbers, BOM, and Centroid positional files, and uploaded the board for fabrication and assembly. →





Above (Left)
This is smaller and more robust than a dev board and separate temperature sensor

Above (Right)
A break in the PCB thermally isolates the temperature sensor as much as possible

GETTING HELP

We've covered the basic use of KiCad in this series but, as you progress, you may have questions or hit upon problems that you aren't sure how to solve. There are a few places that you can go for additional support.

The first port of call in most cases should be the documentation at **docs.kicad.org**. These go into more depth than we have in this series, and should also be updated as things change.

There are also online communities, both of the forum variety (**kicad.org/community/forums**) and chat (**kicad.org/community/chat**). In both cases, these are community resources for users to help each other out.

If you are using KiCad for a professional project and you want an appropriate level of support, the KiCad Services Corporation may be able to help. It covers everything from helping you use KiCad, to bug fixes, and implementing features you need. This company employs some of the lead developers of KiCad, so directly supports the project.

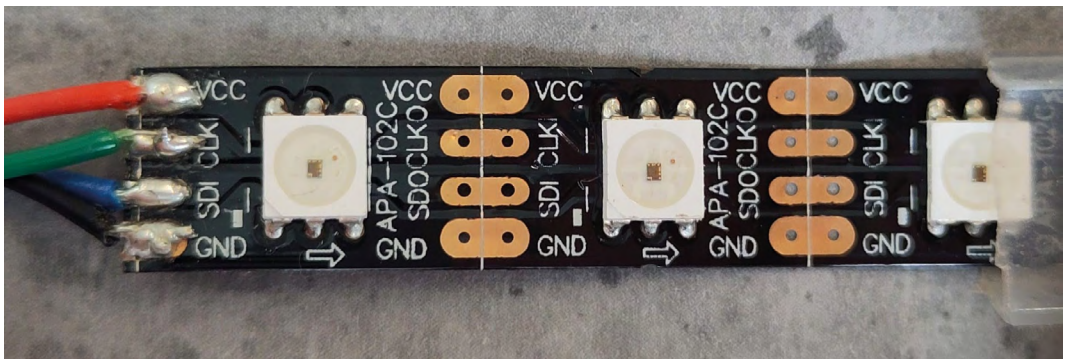
BRING IT TO LIFE

In the previous issue, we looked at creating a custom version of CircuitPython but, this issue, we'll use the easier (though perhaps a little more boring) method of using the CircuitPython build for Raspberry Pi Pico, and incorporating any changes we need in the code.

When you receive your boards, they should come with blank flash chips, which means that when you plug them into a computer, an RP2 flash drive appears. You can copy the Raspberry Pi Pico build of CircuitPython onto this (download it from **circuitpython.org**). Once this has fully copied over, the RP2 drive should disappear and a CircuitPython drive should appear. The only thing we need to add to this is the DotStar module that we'll need to control our LEDs. You can get this from the Library bundle which is also at **circuitpython.org**. This comes as a zip file, so extract it, and copy the **adafruit_dotstar.mpy** file from the **lib** folder to the **lib** folder on the CircuitPython drive. Your device is now all set up and ready to start coding.

The AHT20 sensor on this board is connected via I2C, and we can connect to it using the `adafruit_ahtx0` module. We just need to set up an I2C object first. This is done with:

```
i2c = busio.I2C(board.GP1, board.GP0)
sensor = adafruit_ahtx0.AHTx0(i2c)
```



Above
Make sure you attach wires to the right end of the LED strip (the arrow shows the direction the information travels through the strip)

Once this is set up, you can access the temperature and humidity with `sensor.temperature` and `sensor.relative_humidity`.

Now we've got temperature and humidity readings on the board, the next question is what to do with them. We've broken out the spare GPIO pins, so we can hook up almost any other hardware we want. We've opted for two strips of DotStar LEDs. These are addressable RGB LEDs, so we can set any of them to be any colour. We're going to use them as thermometer-style outputs, with one LED lit up at a time. As the temperature or humidity goes up, the lit LED moves up the relevant strip.

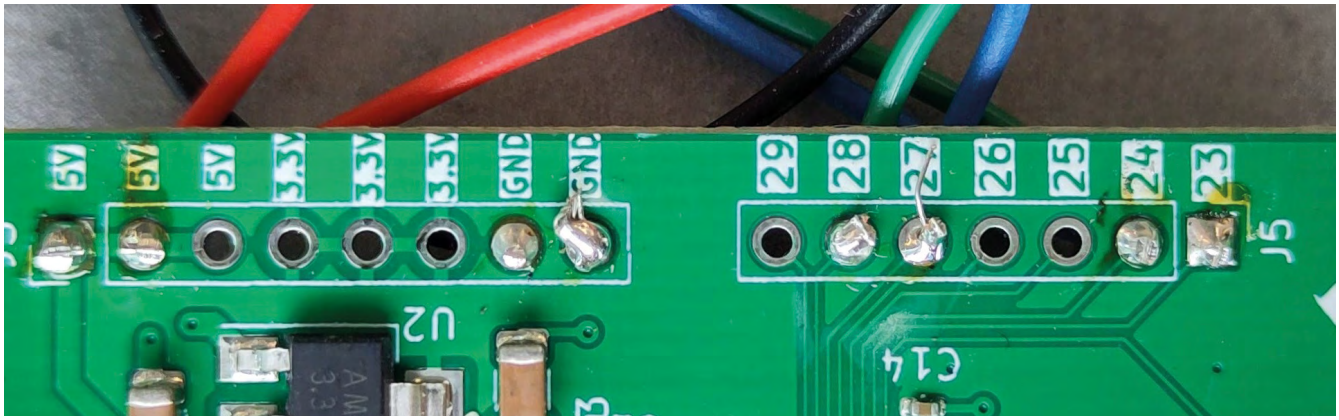
The DotStar LED strips have four connections, VCC goes to 5V, GND goes to GND (and there are two of each on the board, so they can connect directly to the pin). Each strip also has a Clock and Data input connection, labelled SCK and DIN. We've used pins 23 and 24, and 27 and 28 for these, but any of the GPIO pins will work.

We can create the two DotStar objects with:

```
dots_temp = dotstar.DotStar(board.GP28, board.
GP27, 14, brightness=0.2, auto_write=False)
dots_humid = dotstar.DotStar(board.GP24, board.
GP23, 14, brightness=0.2, auto_write=False)
```

You can then access them like lists and assigning an RGB tuple to an element in the list with. For example, `dots_temp[0] = (100,0,0)`. There are also some methods that we can use including `fill()`, which sets all LEDs in the string to a particular colour. The full code for our temperature and humidity thermometers is: →

```
@import time
import board
import busio
import adafruit_ahtx0
import board
import adafruit_dotstar as dotstar

dots_temp = dotstar.DotStar(board.GP28, board.
GP27, 14, brightness=0.2, auto_write=False)
dots_humid = dotstar.DotStar(board.GP24, board.
GP23, 14, brightness=0.2, auto_write=False)

i2c = busio.I2C(board.GP1, board.GP0)
sensor = adafruit_ahtx0.AHTx0(i2c)

dots_temp.fill((0,0,0))
dots_humid.fill((0,0,0))

temp_range = (10,30)
temp_colour = (100,0,0)

humid_range = (0,100)
humid_colour = (100,100,0)

while True:
    print("\nTemperature: %0.1f C" % sensor.
temperature)
    print("Humidity: %0.1f %%" % sensor.relative_
humidity)
    dots_temp.fill((0,0,0))
    dots_humid.fill((0,0,0))

    temp_index = int(((sensor.temperature-
temp_range[0])/(temp_range[1]-temp_range[0])) *
len(dots_temp))
    dots_temp[temp_index] = temp_colour
    dots_temp.show()

    humid_index = int(((sensor.relative_humidity-
humid_range[0])/(humid_range[1]-humid_range[0])) *
len(dots_humid))
    dots_humid[humid_index] = humid_colour
    dots_humid.show()

    time.sleep(2)
```



Above
We've left as many pins as possible for additional hardware to be connected

You can save this to your board and you should see the appropriate LEDs light up. Hold your thumb over the temperature sensor and you should see the LEDs change.

GOING FURTHER

We've now come to the end of our series on KiCad, and we've worked our way up from simple boards that connect modules and breakout pins, to creating our own microcontroller boards with the hardware we want, and we've programmed them along the way. From here, you can take this hobby wherever you like.

> **If you're interested in the skill of PCB design, then you can work on** technically more challenging boards

If you're more interested in practical solutions, then you can keep designing boards as you need them. If you're interested in the skill of PCB design, then you can work on technically more challenging boards. For example, you could try starting with some of our examples and seeing how small you can make them. You could work on interesting or artistic boards, such as those used for badges at hacker-focused tech events.

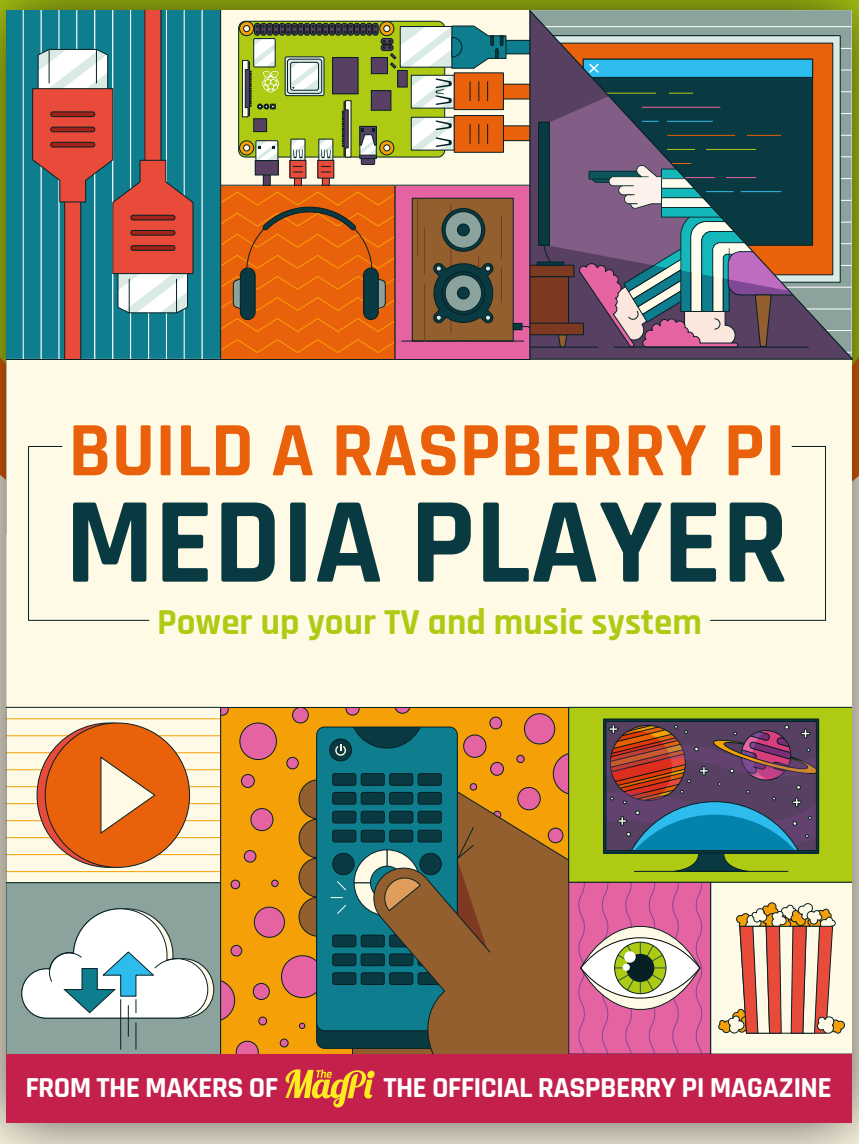
OTHER PROGRAMMING LANGUAGES

We've focussed on CircuitPython because that's the language we're most familiar with, but you can program the RP2040 board in a wide variety of languages.

The two officially supported options are C (via the Pico SDK), and MicroPython. Additionally, there are community projects that have brought a wide variety of languages to the RP2040 processor. Some of these might work out of the box, and others might need a little tweaking in order to run.

- **Arduino** – technically, this is C++, but the IDE and ecosystem make this feel like a distinct language. It's incredibly popular with hobbyists, mostly because there are languages for essentially every bit of hardware ever created.
- **TinyGo** – **hsmag.cc/TinyGoPico**. Go was designed by Google to help them solve very large problems. Despite the fact that microcontrollers, by their nature, very rarely solve large problems, this language has leaked into the embedded world. We've never used it, so can't really comment, but lots of people smarter than us think it's good, so it probably is.
- **Rust** – **github.com/rp-rs/rp-hal**. As far as we can tell, this is supposed to do the same things as Go, but it was created by Mozilla rather than Google.
- **Forth** – **hsmag.cc/PiPicoForth**. According to people with grey hair who inhabit universities, Forth is the greatest language ever written. We're not entirely sure why, but they're very convinced of this fact.
- **BASIC** – **geoffg.net/pi comite.html**. This writer learned to program in BASIC and has a certain fondness for it. However, the world has moved on over the last 30 years, and BASIC hasn't. It can give some people a good hit of nostalgia, but that's about its only purpose.

Your FREE guide to making a smart TV



magpi.cc/mediaplayer

Raspberry Cash

Build a working cash register with a Raspberry Pi

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Right
Yes, you could always just wear a trader's money belt and carry a calculator if you need to, but a battery-powered cash register with a working drawer looks much nicer. Being able to take cash and generate receipts without a power connection can save hundreds of pounds in cash if you're at a trade show where the organisers charge for power connections

QUICK TIP

If you want to delete the last item scanned, use the **BACKSPACE** key. If you have text in the register's buffer, it will delete the last character typed. If the buffer is empty, it will delete the last line item added to the current sale.

The world of the future is a largely cashless society, but the world of today still uses metal tokens and paper promises to facilitate transactions. In an urban store you'll probably find an electronic point of sale (EPOS) system that handles the shopping experience, but when you're on a market stall away from regular power, the conventional EPOS experience isn't as easy to implement. In this article, you'll see how to create a working cash register with a secure money drawer, barcode scanner, and receipt printer. You'll get it all running from a Raspberry Pi and DeWalt battery, and you'll also have a built-in daily transaction log, and the option to use any wireless device as an extra till display.

CALCULATOR WITH CONTROL ISSUES

Cash register and EPOS system are really just catch-all terms for a fancy calculator, and it's worth defining exactly what we're going to be building here, and why. A typical modern cash register sits on the counter next to the cashier and it allows the cashier to calculate the total value of the items being purchased. In the old days, this was a purely mechanical beast that required the cashier to know the price of each item and enter it correctly. As technology advanced, barcodes were used to give each item a unique code that could be read by the cash register and checked against a database of prices, so that the cashier no longer needed to enter the items manually into the register unless the barcode was damaged. In modern times, the cashier

is sometimes entirely missing from the equation and the shop makes us do all of the work ourselves, paying for items by putting money into a slot or waving a debit card in front of a scanner. What you will be creating is something like the last generation of cashier-driven register. Your cash register will have a touchscreen interface, a barcode scanner, a remote keypad, a receipt printer, and the facility to keep daily transaction lists that can be read at a later date. This register won't accept debit or credit cards by default, but it should be possible to add that functionality if you want it. With a little extra work and a third-party API like Square, you could generate transactions that accept card payments from a contactless reader.

The cash register application is written in Python, and uses a Flask web server to generate a web page that displays the current transaction in a browser. The browser launches in kiosk mode when the computer starts. This might seem a strange way to implement the display, given that there are several GUI tools (like Pygame, PyQt, or Tkinter) available for Python, but it's actually a very flexible way to implement the interface. The web browser can be easily customised with HTML and CSS to alter its appearance without needing to change the main Python code, and additional interface functionality can be added using JavaScript on the client side. You can also connect multiple web browsers to the same cash till and use them as remote displays or remote terminals for the register.

For this project, you will be connecting the Raspberry Pi to a basic receipt printer, which requires between 5 and 9 volts to power it. You may have already spotted the immediate problem that the Raspberry Pi GPIO header uses 3.3V and isn't 5V tolerant, although this isn't actually a huge issue. The printer only needs bidirectional communication to report the status of the paper tray (which isn't really necessary for everyday use) and the 3.3V GPIO voltage of the Raspberry Pi is high enough to trigger a high input on the printer, so it's possible to connect the printer directly to the Raspberry Pi without a level shifter, →

REGISTER INPUT

The cash register application gets input from the barcode scanner and keypad by monitoring system-wide key presses using pynput. Key presses from the touchscreen web interface are generated by passing an API request from the web page back into the main app, which generates the appropriate key codes using pynput. It's worth noting that the generated key code may vary between OS and the type of device that generates them. For example, the decimal point may generate different codes depending on whether it's being sent from the key on the main keyboard, the numeric keypad, or an external device. You may need to adjust the Python code slightly to suit your particular input devices.

The barcode scanner used in this project appears as a standard input device, and to all intents and purposes acts like a keyboard. When the user scans a barcode, the code is translated into a string of characters and 'typed' into the computer. To avoid the need for a database of scannable items, the barcode is encoded as a simple string with a ':' symbol as the delimiter. The first part of the string is an item description, and the second is the price of the item. So a barcode with the string 'snacks:5.4' would generate a sale item with the title 'snacks' for £5.40 (or whatever currency you're working in). The type of barcode isn't actually very important: as long as it can encode the text you want to enter correctly, the scanner will read and decode it as plain text.

YOU'LL NEED

- **Raspberry Pi 3** (or similar – less powerful versions draw less power)
- **7" HDMI touchscreen** or a tablet to view the till display
- **Barcode scanner** amazon.co.uk/dp/B08CHFN3T2
- **DC solid-state relay** amazon.co.uk/dp/B07PYZZ3G4
- **Cash drawer, 12V RJ11 connection** amazon.co.uk/dp/B09ZL3J1M9
- **M110 thermal receipt printer** amazon.co.uk/dp/B0C33JSKTL
- **Set of laser-cut case pieces** (or suitable project container)
- **2 × XL4015 5A step-down adjustable PSU** amazon.co.uk/gp/product/B081JP5YZP
- **DeWalt 5A battery**
- **3D-printed DeWalt battery adapter** (or similar, hsmag.cc/DWBatteryAdapter)
- **Inline fuse 1A**

provided that you don't connect to the RX pin (GPIO 15) on the Raspberry Pi.

You can normally force a receipt printer to produce a self-test by powering it on with the button pressed. Although it varies from brand to brand, you'll normally see information about the firmware, character sets, and serial connection settings for the printer. Most serial ports will be defaulted to 9600 or 19,200, with the usual 8 bits, no parity, and 1 stop bit.

Thermal printers use a control language called ESC/POS, using escape codes to generate character effects. It's an old technology – and unlike a modern desktop printer, a thermal receipt printer has very limited capabilities. Printing images is quite complicated to achieve, and printed text relies on a built-in font with escape codes used to apply simple modifiers like reverse-printing and double-size characters. There's also a built-in facility to generate barcodes in most receipt printers. Some units also have a special connection that can be used to trigger the cash drawer when a receipt is printed.

While it's possible to set up a serial connection to the printer and send the escape codes directly, it's much easier to use Adafruit's Thermal Printer library and Blinka. Blinka allows you to use CircuitPython APIs in regular Python, and can be installed from **circuitpython.org/blinka** or using the pip installer.

QUICK TIP

Thermal printers are fussy and easy to confuse. If you're getting garbage printed unexpectedly, try power-cycling the printer to reset it.

Below
Thermal receipt printers are very simple and communicate using an RS-232 or TTL port, although some of the more modern units have a USB or Bluetooth connection

DRAWER TRIGGERS

Cash drawers are normally available in 12 V or 24 V varieties. The 24 V version is probably more common on modern EPOS systems, while the 12 V versions are usually found on older-style cash registers. Some new cash drawers are smart and have a processor built-in with a USB trigger, but in general, most 'normal' cash drawers just have an RJ11 socket that connects to the solenoid and to a microswitch that triggers when the drawer is open. On these cash drawers, you can easily map the pinout of the wires by measuring the resistance between the pins on the RJ11 socket. The resistance of the solenoid will be easy to detect with a multimeter, and the continuity between the pins connected to the microswitch should also be easy to detect. Opening the drawer is as simple as applying the correct voltage to the solenoid for a short amount of time (typically less than 250 ms). On some of the more complicated drawer systems, there are two independent cash drawers with separate solenoids that share a centre tap. This is sometimes referred to as 'Epson wiring' with pins 2 and 5 connected to the outside (negative) legs of the solenoids, while pin 4 is the shared (positive) centre tap. There are several other systems in use, but these are probably the most common in the modern day.

Finally, you'll need to install pynput, which is the module that reads and generates key press events. You can install pynput straight from the pip package manager using `python -m pip install pynput`.

One thing to watch out for with thermal printers is that they are relatively slow and don't really communicate bidirectionally with the device they're connected to. This means that there's no signal to say when the printer is ready to print, or when the printer is already printing. It's quite possible to send data to the printer so quickly that the internal buffer becomes overwhelmed and it starts missing characters. You need to make sure that the program you're using won't send the data to the printer faster than it can output onto the till roll.

Begin making your own cash register by installing the latest version of Raspberry Pi OS onto a Raspberry Pi 3, or later model, and making sure that all of the packages are up to date. Once you're ready, go into the Preferences menu on the desktop and bring up the Raspberry Pi Configuration application. Check that the serial port is enabled,

and serial console is disabled, accept any changes, and reboot the machine.

Next, you'll need to install some Python libraries that the cash register application depends on to get your code working. You'll need the Blinka library and the Adafruit Thermal Printer library, which you can install by following the instructions at **hsmag.cc/AdafruitReceiptPrinter**. Next, you'll need to install the Flask framework, which you will be using to create a Python-based web server. Follow the instructions found at **hsmag.cc/Flask** to do this. Then, install the cash register files into a folder called **till** on the desktop (or in the location of your choice). The main application in the folder is **keys.py**.

Next, you need to wire up all of your electronic components to the Raspberry Pi's GPIO pins. You'll be drawing power from an 18V battery, and dropping to 5V and 12V using buck converters. The 12V buck converter is used to open the cash drawer, and can →

KIOSK MODE

You'll probably want to automatically run the **keys.py** file and start a kiosk mode web browser when the computer starts up. To do this, you need to install some packages that will help you control the window manager, and edit the autostart file in **/home/user/.config/lxsession/LXDE-pi**.

Begin by installing the packages:

```
sudo apt-get install x11-xserver-utils unclutter
```

Edit the autostart file using nano:

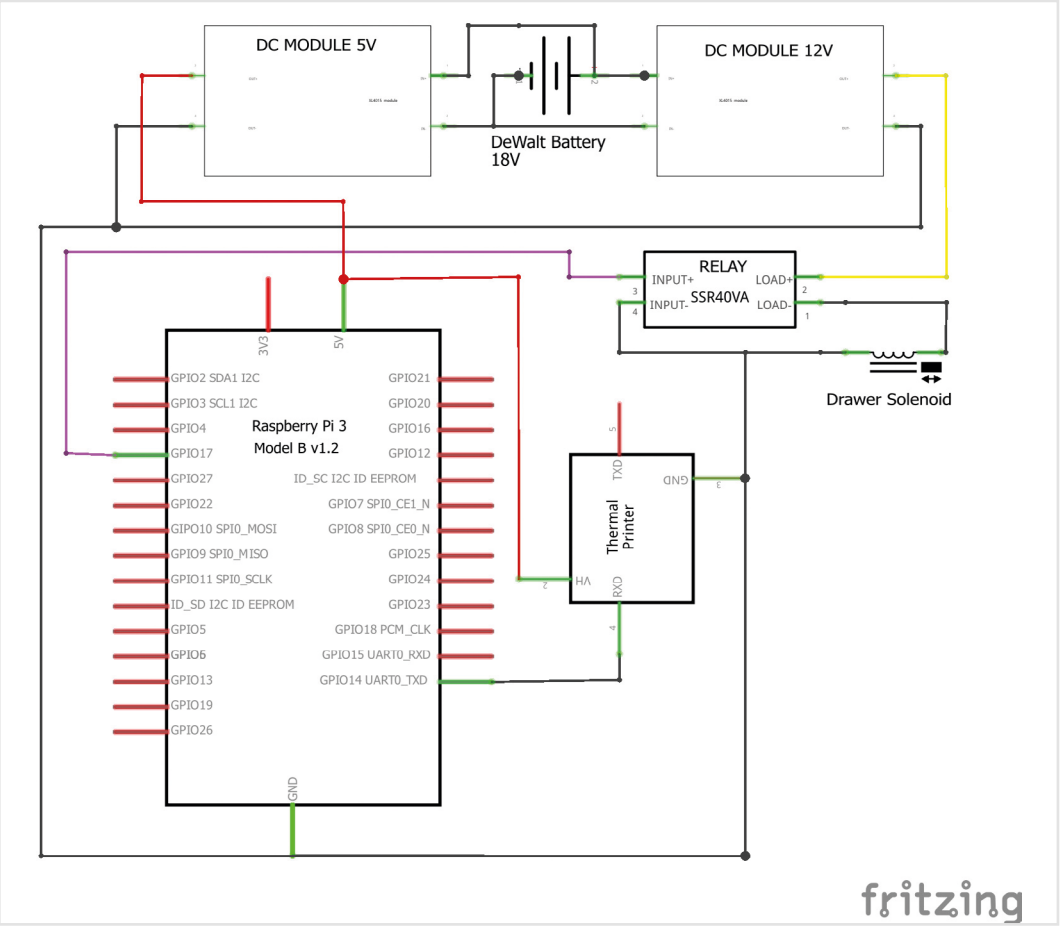
```
sudo nano /home/pi/.config/lxsession/LXDE-pi/autostart
```

Add the following lines to the autostart file:

```
@python /home/user/Desktop/till/keys.py
@chromium-browser --kiosk --incognito --disable-pinch --overscroll-history-navigation=0 http://127.0.0.1:5000
@xset s noblank
@xset s off
@xset -dpms
@unclutter -idle 0.1 -roo
```

Save the file and exit nano. This autostart configuration will cause the Raspberry Pi to boot into a kiosk mode, disable the screensaver, and hide the mouse. It'll also mean your desktop will be completely blank. If you want to be able to access the desktop, you'll need to edit the autostart again to include the following lines at the top of the file:

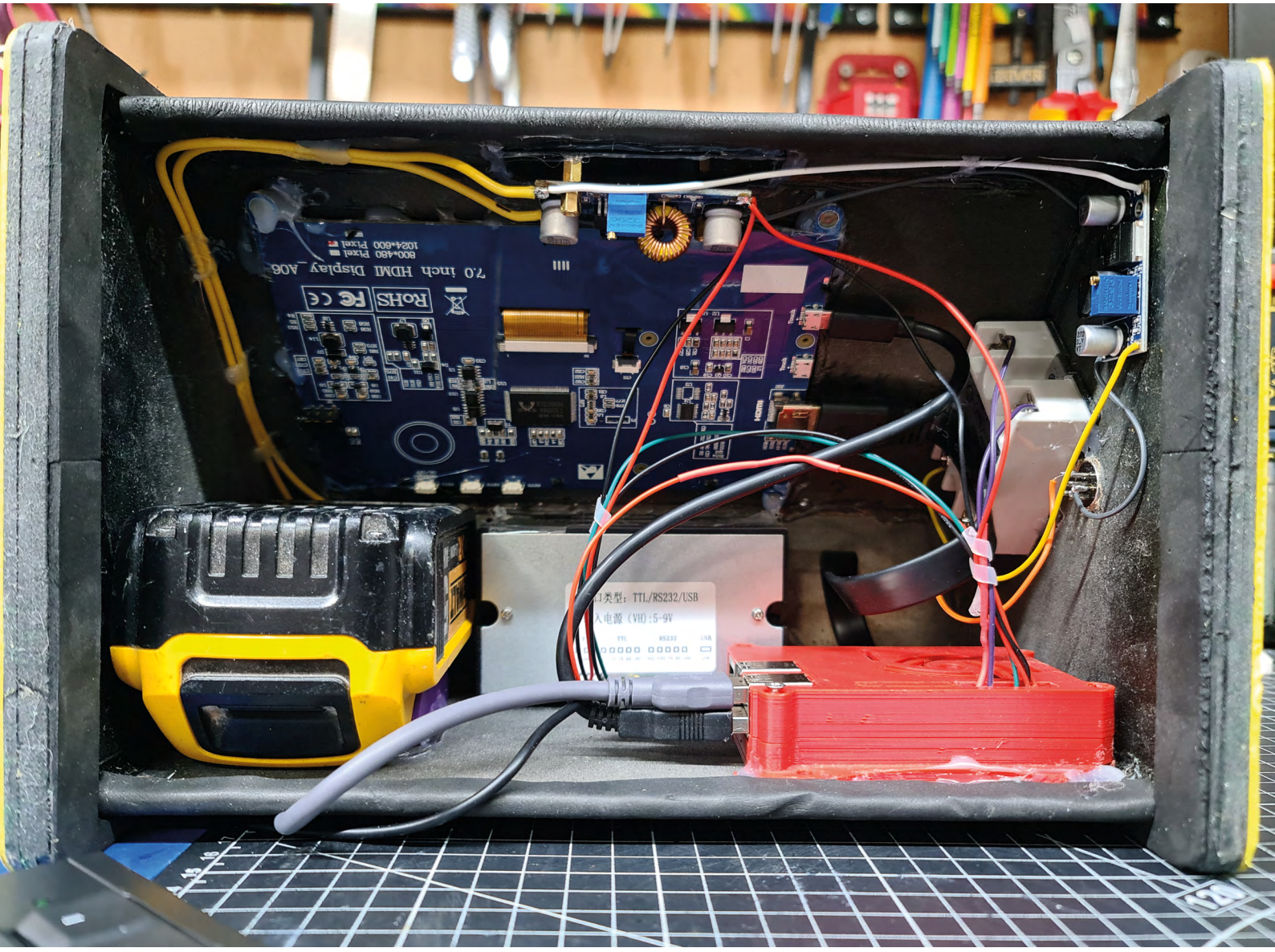
```
@lxpanel --profile LXDE-pi
```



QUICK TIP

The more powerful the Raspberry Pi, the more battery power it will need to run. A Raspberry Pi 3 will be fine for this project.

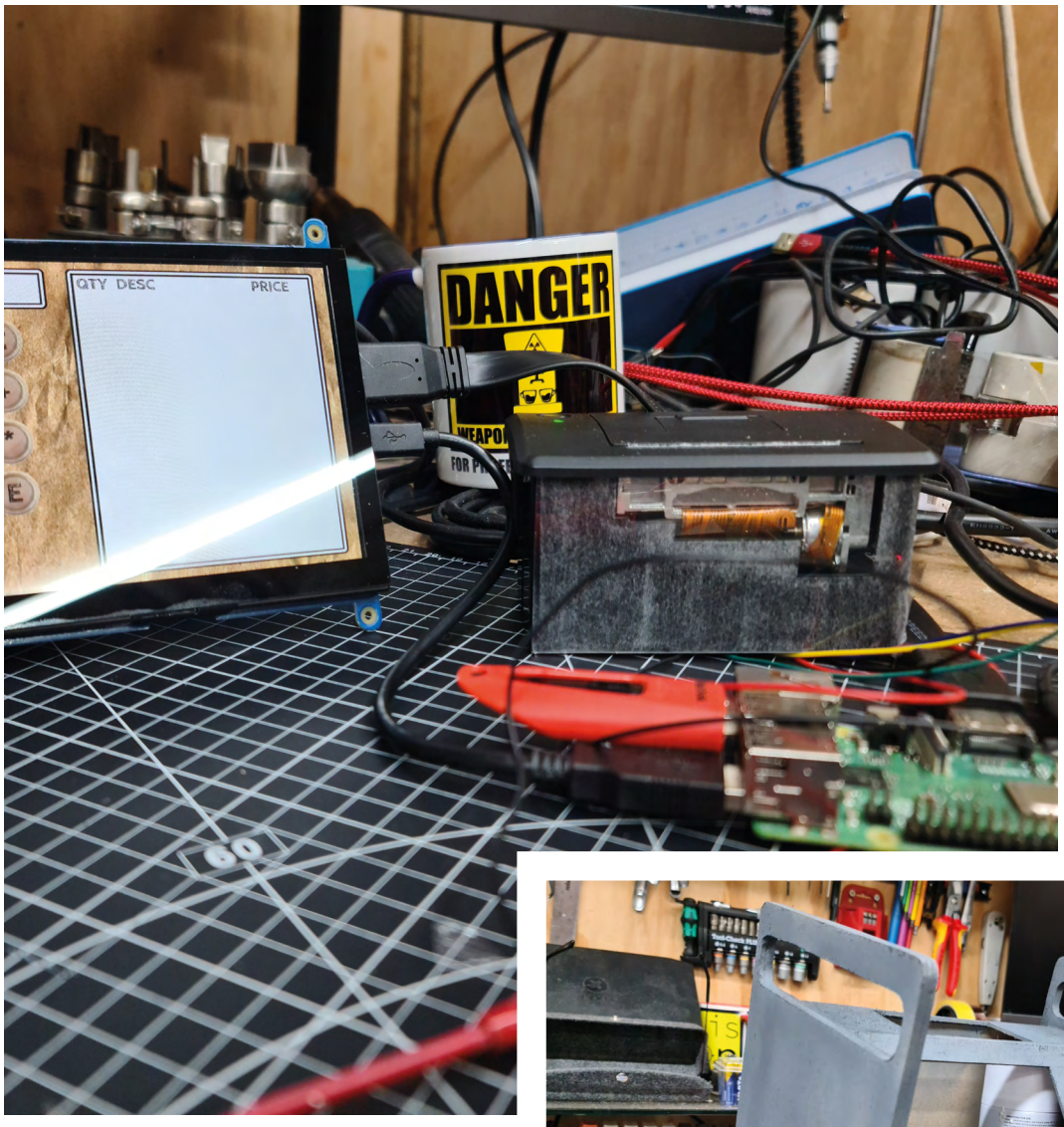
Left
You can see from this diagram that the wiring for the cash register is not complicated, but there are a couple of things that you should watch out for. Firstly, this diagram doesn't show the USB or HDMI connections for a touchscreen or other peripherals. When you are mounting your Raspberry Pi inside your case, be sure to allow enough space for cables to plug in. You should also apply the same thinking to make sure that you can get batteries in and out easily. Remember that any digital cable is susceptible to interference, so keep the cables to your external components as short as possible



Above
The inside of the case should be neat and tidy, properly fused, and inaccessible to the general public. Lots of venues will ask you to create a risk assessment for your stall, and prove that your equipment is safe. Exposed wires or pins (even low voltage ones) don't look good to an event organiser or health and safety officer

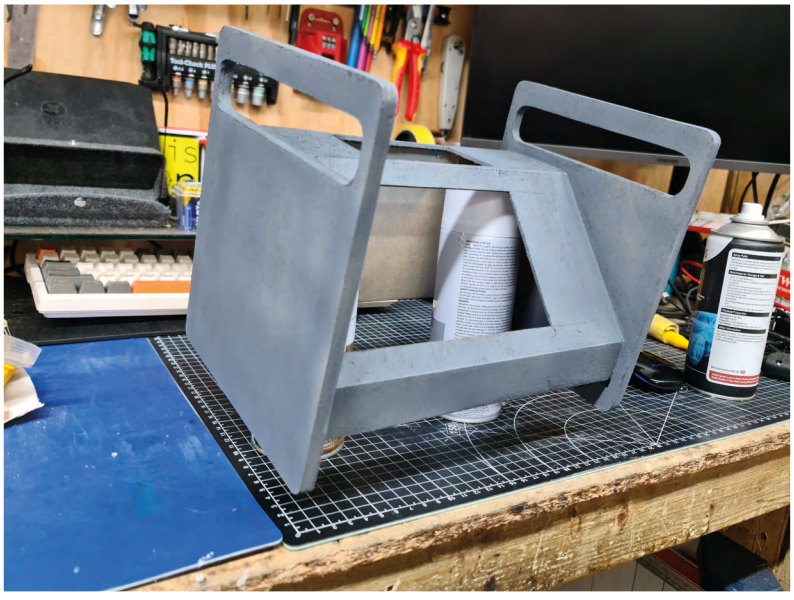
be omitted if you have a 24V cash drawer. If you have a 12V drawer and want to risk omitting the buck converter anyway, the risk and responsibility are your own to consider. It may work fine, or it may burn out the drawer solenoid after a while.

The 12V power is controlled by a solid-state relay connected to GPIO 17, and should only trigger for a few milliseconds at a time, which would probably prevent the solenoid from damage, but that isn't guaranteed safe to use. The other 5V buck converter is used to power the Raspberry Pi and the thermal printer. It's not unusual for thermal printers to print out their input voltage during a self-test. This isn't very accurate, so don't be too worried if the reported voltage is slightly under 5 volts. The thermal printer is also a power hog, and can draw up to 10W while printing. The Raspberry Pi itself (and USB peripherals) will only be drawing a couple of watts, but if you are using a touchscreen display, it can add another 4 or 5 watts to overall consumption depending on how it's configured. If you're planning on a long workday, consider taking along extra batteries just in case you start running short.



Below
This project is likely to get bounced around a bit when you're setting up and disassembling your store. Make sure that your case is sturdy enough to deal with that by gluing your joints well

Left
There's plenty of spare pins for expansion of the cash register, so adding NFC readers, servos, or custom button boards is absolutely possible



There are a few extra things that you could consider adding to this project if they suit your needs, including adding an NFC reader to read staff cards so that you can record who took a sale, or adding extra 'hidden' web pages to make downloading daily transaction records easier. You could also add a second battery connector to double the battery life and make it possible to swap batteries without restarting the till. You could also implement a local database of items with prices, so you can set the prices at the till rather than on the barcode itself.

TUTORIAL

3D-printed planters

Make the most of your growing space

Ben Everard

Ben enjoys gardening because, unlike most of his projects, plants will complete the building process themselves (as long as he remembers to water them).

Right
You can mix and match pots of different styles, sizes, and colours

Despite the best efforts of the gods of rain, the winter is finally coming to an end. Plants are sprouting all around – ones we planted and ones we haven't. This author has decided to turn his attention to generally making his home a little greener and much more flavourful.

Plants, like all life, are amazing replicators. Inside each seed is a tiny spool of cellulose filament and a solar-powered print head capable of squirting out this building block into exactly the right shape for that species (OK, we're computer nerds rather than biology nerds, and might have got that bit wrong). This makes us think that they're perfect partners for a bit of 3D printing.

The two big problems we've had with herbs are that there are quite a lot of them and they take up a lot of space (whether you want them to or not). One little mint plant very quickly becomes a whole carpet that, while delicious, covers half the garden.

There is a solution to both these problems – vertical gardening. This is the process of building your plants upwards rather than outwards. With this, we can fit a lot of different plants into a small space and keep each contained in its own pot.

ONWARDS AND UPWARDS

A lot of vertical gardening is built around hydroponics, and if that's a route you want to go down, you'll find

Above The PrusaVert system uses old filament spools to add strength to the planter

plenty of inspiration online. However, we wanted a more traditional soil-based approach for now. After perusing some repositories of 3D-printable models, we found a couple to try out.

PrusaVert (**hsmag.cc/prusavert**) incorporates Prusament spools (which we just so happen to have a few spares of at the moment) to make a robust barrel-like vertical planter suitable for life outside, while High Gardens (**hsmag.cc/highgardens**) makes it possible to tower-up plant pots on the windowsill.

> **With this, we can fit a lot of different plants into a small** space and keep each contained in its own pot

Let's take a closer look at PrusaVert first. Prusament spools are two sides of plastic joined by cardboard in the middle. You can pull the two sides apart to remove the middle. You can buy replacement filament with just the cardboard, then slot the sides from the previous spools on. Alternatively, you can use these as the basis for additional builds, such as this vertical garden. Each tier consists of three 3D-printed plant pots that attach to one of the spool halves. They are designed to screw together, but we decided to try a drop of glue instead and, so far, they seem to be holding. Each tier can be joined together, but we have left them loose so they can be rearranged as needed. We think this will be useful when it comes to bringing some of them indoors to overwinter.

The pots come in two sizes – we went with the larger 12 cm-high version as these will hold more water and allow for bigger plants. Our plants seem happy and are quickly adapting to their new home.

CHOOSE YOUR OWN

High Gardens is composed of multiple pieces. You start with a base and then add as many pots as you like (or dare). Each pot slots into the one below it. →

Above
The chicken run is in the sunniest spot in this garden, and now we can put that sun to good use

Right
A hook on the planter goes over the plank to secure the pot in position

These slots let the water flow through the central pillar so, in theory at least, you can add water to a top reservoir and it will water each pot, with any excess draining into the tray at the bottom. We found that it didn't work particularly well as it was hard to gauge how to add water to the top so that all the pots would be watered without overflowing the tray at the bottom. We instead watered each pot individually, but we did find the water flow system useful as it meant that each plant could drain and the excess water would be captured.

We found that a tower roughly three or four pots high was stable on its own, but any higher seemed a bit precarious. This is partly due to the presence of two energetic kittens in the test area. We once ended up with plants scattered about the floor, but this was due to a kitten jumping up and clinging onto the blind, pulling herself along until she was above the High Gardens, and then dropping down onto it. If your environment is more sedate, you may get away with a higher garden. There are wall clips available (**hsmag.cc/hgclip**) which make the prospect of compost-covered floors a bit less likely.

There is a selection of different designs and sizes for both pots and bases, though some of the bases look alarmingly narrow. Perhaps some people like

TYPES OF **FILAMENT**

PLA isn't UV (ultraviolet) stable. This means that, over time, the UV rays in sunshine will gradually break it down. How fast this happens depends mainly on how much UV they are exposed to. Most plants like to be positioned outside or on windowsills which are, unfortunately, places where there's quite a bit of UV light.

Depending on what plants you grow, you might find that the foliage protects the plastic from UV. This might be sufficient to ensure that your planter lasts long enough, or it might not.

You can protect PLA with a layer of paint. Alternatively, you can use a UV-stable filament such as ASA.

Above
Each planter protrudes enough for the plants to get plenty of light, but if you place it against a wall, you might struggle to get plants to grow in some sections

to live dangerously, or perhaps some places aren't patrolled by feline vandals. The choice, as they say, is yours.

HANGING UP

Both the previous options build up from the floor, but sometimes you want to attach your pots to a wall. That's exactly what this vertical gardening planter does: **hsmag.cc/vertplant**. The attachment is simplicity itself – there's a hook on the planter, and you just need to secure a plank of wood to the wall with enough of a gap to hook the planter over. You can have as many or as few as you like hanging from your planks.

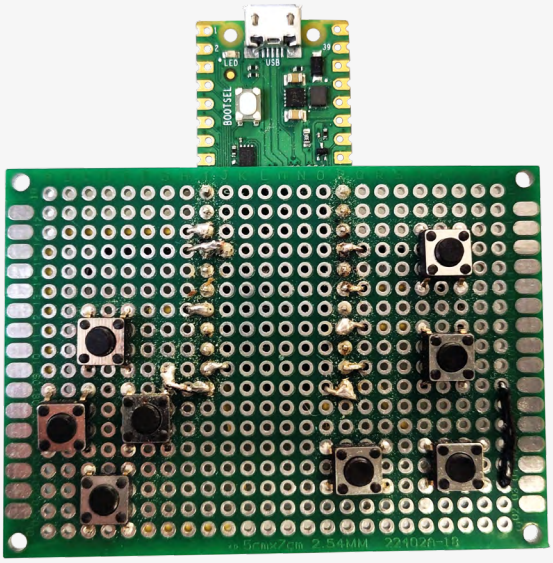
Misan, the designer of this planter, has a few versions with different levels of drainage. The v4 has intensive draining, with just a mesh at the bottom. This may be OK for succulents and other plants that like dry conditions, but moisture-loving plants may struggle. The v3, with its less-intensive draining, will be better suited for most plants.

This model just about prints successfully without supports, but we got better results when we incorporated a few. It doesn't need a full set, but using PrusaSlicer's automatic support painting feature, we added just enough to ensure the long bridges printed well without wasting lots of filament.

We've been growing herbs indoors and outdoors in these for a few weeks, and so far, we're delighted with the results. We've got plenty of flavourful greenery to add to meals and drinks, and they look and smell wonderful as well. As the weather gets warmer, we'll print more and more of these to keep up with the growing demand.

LABELS

As well as printing the right plant pots, you can print other accessories for your garden. The most popular are plant labels. These can be as decorative or as plain as you like. Printing in two colours (either with an automatic colour switcher, or by switching filament at a particular Z height) is a great way to make striking labels that add a bit of colour to your veg beds.



Pico keyboard and gamepad

Two ways of using a Raspberry Pi Pico to control a computer

Above
The easiest way of building a controller like this is to solder some buttons onto protoboard

Ben Everard

Ben spends more time making games controllers than he does playing games. It's a strange hobby, but he enjoys it.

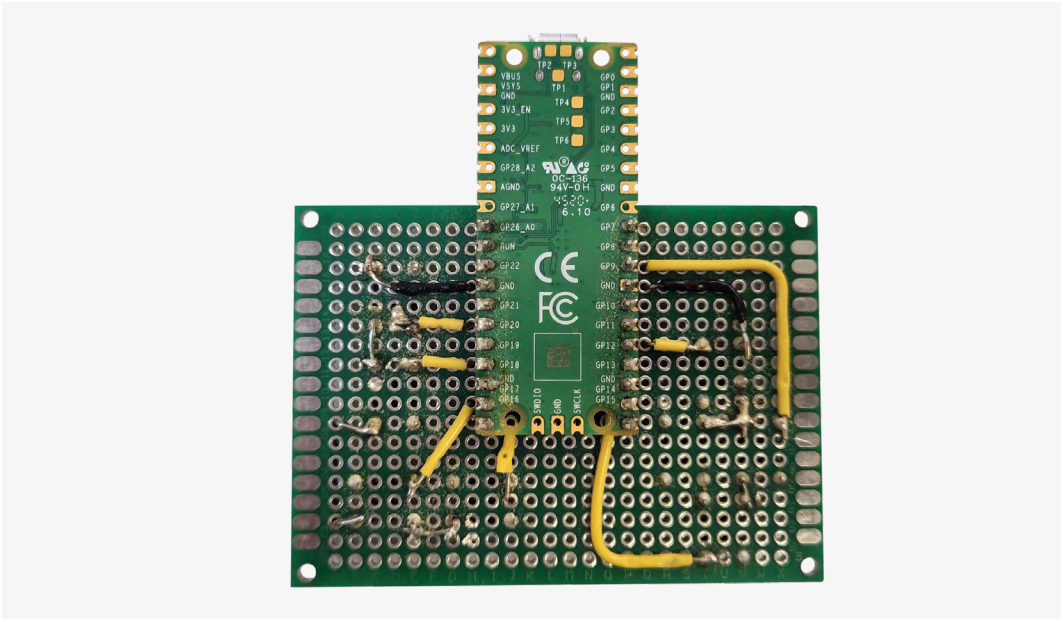
The first thing that you usually learn when doing electronics with a microcontroller is blink an LED. The second is connecting a button. In this article, we're going to take this second skill and use it to build a peripheral for a computer. There are surprisingly large stacks of software between the switch on a keyboard and the program you use it to control. Fortunately, though, we don't have to concern ourselves with them. We can grab some pre-written firmware, adjust a bit of configuration, and then have everything 'just work'.

Buttons come in many different shapes and sizes, but they mostly do the same job – bring two contacts together when you press them. For our microcontroller to read the state of the button, we just have to connect it between a GPIO pin and ground.

You can hook up GPIO pins like this on many microcontrollers, but there are a few features that make some work better than others. The first is internal pull-ups. To read the position of a button, you need to connect one part of it to ground, but you also need to add a resistor between the GPIO pin and a positive voltage. Since this is a very common thing to

GETTING TO GP2040-CE WEBCONFIG

You configure GP2040-CE using the web-based config tool. To get there in the first place, you have to hold down button S2 while you plug in your controller. However, if you don't have a button S2, you can flash the force-webconfig version as we have done. If you want to get back to the web config, you can just flash it again.



Left
Once all the components are in place, you can solder wires between them to connect them up as needed

do with microcontrollers, many have built-in resistors, and these are known as internal pull-ups.

The second feature that will prove useful to us is USB device support. Most microcontrollers can plug in via USB, and be programmed that way, but depending on the implementation of the USB software and hardware on the microcontroller, some can be programmed to be keyboards, mice, and gamepads.

We won't go into these in too much detail other than to say that the Raspberry Pi Pico is an excellent choice for both of these reasons, and it also has another little trick up its sleeve – an abundance of ground pins. These mean that you can wire up buttons individually without having to worry about the logistics of connecting all the buttons to ground.

ON / OFF

You can connect the switches up using whatever method you feel comfortable with. You can use a breadboard, you can solder wires between them, you can protoboard them – it really doesn't matter. Just make sure that each switch is connected between a GPIO pin and ground.

We're going to make a games controller first, so you're going to need at least six buttons (four for direction and two action buttons). Once you've wired them up, it's time to add the software.

We're going to use the GP2040-CE firmware. This takes an RP2040 board and turns it into a high-performance gamepad. It takes care of all the tricky things like communicating with the computer and checking for button presses. All you have to do is tell it which GPIO pin relates to which button. You can get the firmware from **hsmag.cc/GP2040**.

ENTER THE MATRIX

You might have noticed that Raspberry Pi Pico has 26 GPIOs you can use to attach buttons to, but that would only satisfy the most militant of minimalist keyboard users. There are some microcontrollers with more GPIOs, but they aren't particularly common (at least in the hobbyist community), and dev boards breaking out all these connections can be expensive. How, then, are we able to make DIY keyboards?

The answer is a matrix. Rather than connecting up each button to its own GPIO pin, we create a grid of wires horizontally and vertically. We can then place switches joining a horizontal and vertical wire at the crossing points.

We can then use the GPIO pins on one axis as output, and the other as inputs. By providing power to just one row at a time, we can use the inputs on the columns to detect which buttons in that row are currently pressed.

We can scan through the rows quickly to detect presses at any place on the matrix with only a few GPIO pins. The only potential problem is that multiple button presses can create alternative paths through the matrix which can cause problems. The solution is to use diodes – a component that allows electricity to pass in only one direction. Using this combination of a matrix layout and diodes, you can connect up to 169 buttons to the 26 GPIO pins on Pico.

There, you'll see versions for lots of different controllers all built around the RP2040 microcontroller. The one we want is the **force_webconfig.uf2** file. Download this and flash it to your Raspberry Pi Pico.

Once it has been flashed, you should be able to access the web editor by pointing your browser to **http://192.168.7.1**. →

Left
Regular readers may remember this controller from a previous article. We were able to use the new firmware without any hardware modifications

On this website, you can go to the 'Pin Mappings' section and select which GPIO pins you want to be responsible for what actions. Once you've entered everything and clicked Save, you can reboot your Pico as a gamepad. It's a regular USB gamepad, so you can use it for your favourite games in just the same way you'd use any other gamepad. Want more features? No problem, you can add them. As well as adding buttons, you can add analogue inputs such as joysticks and feedback from LEDs and buzzers.

TINY KEYBOARD

GP2040-CE is great for making games controllers, but what if you want to control a game that takes input from regular old keyboard presses? We can do that too using the exact same circuit as with our gamepad. There's a sizeable community of DIY keyboard makers who have created some excellent and hackable keyboard firmware. The one that we'll be using is called KMK, and it is based on CircuitPython.

To use it, you first need to install CircuitPython on your Pico. To do this, head to **circuitpython.org** and download the latest version for Raspberry Pi Pico, then flash it to your device. You should find that when you plug in your Pico now, a USB drive called 'CircuitPython' appears.

Now, you'll need the KMK firmware from **hsmag.cc/getkmk**. Unzip the file and copy both the **kmk** directory and the **boot.py** file to the CircuitPython drive. To turn this into a working keyboard, we need to add a **code.py** file that describes the particular keyboard.

JOYSTICK

Joysticks come in two forms: analogue and digital. You can get either type in modules that are easy to wire up to a microcontroller. Analogue joysticks are basically two potentiometers, set up so each one reads a separate axis: one for X and one for Y. Each potentiometer has three connections: one for 3.3 V, one for ground, and one output. You can wire them up to an ADC input on Pico and read them using the GP2040-CE firmware.

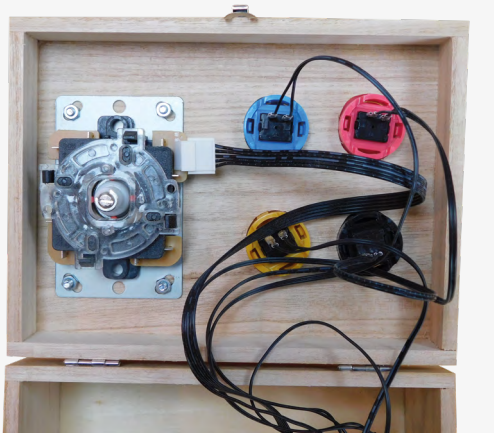
Digital joysticks are basically a stick with four buttons, with a different button being pressed depending on how you move the stick – it's basically a D-pad arrangement rotated in on itself with a stick to do the pressing. In these joysticks, there is usually a single common ground connection, and then a separate GPIO connection for each button. You can wire them and connect them to either gamepad or keyboard firmware.

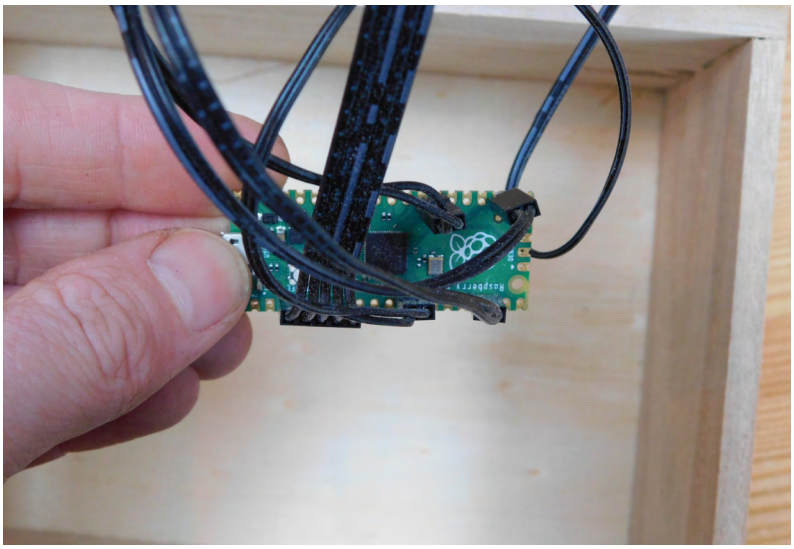
The code for our keyboard is:

```
import board
from kmk.kmk_keyboard import KMKKeyboard
from kmk.scanners.keypad import KeysScanner
from kmk.keys import KC


# GPIO to key mapping - each line is a new row.
_KEY_CFG = [
    board.GP2,  board.GP3,  board.GP4, board.GP5,
    board.GP9,  board.GP13, board.GP17, board.GP21
]

keyboard.keymap = [
    [
        KC.UP, KC.DOWN, KC.RIGHT, KC.LEFT,
        KC.A, KC.B, KC.C, KC.D,

    ],
]

# Keyboard implementation class
```

Right
Thanks to the number of ground connections on Pico, we can solder on all the buttons without complicated wiring

Below
The joystick is really just four buttons that get triggered when the stick is moved





```
class MyKeyboard(KMKKeyboard):
    def __init__(self):
        # create and register the scanner
        self.matrix = KeysScanner(
            # require argument:
            pins=_KEY_CFG,
            # optional arguments with defaults:
            value_when_pressed=False,
            pull=True,
            interval=0.02,  # Debounce time in
floating point seconds
            max_events=64
        )

keyboard = MyKeyboard()



if __name__ == '__main__':
    keyboard.go()
```

OTHER FIRMWARE

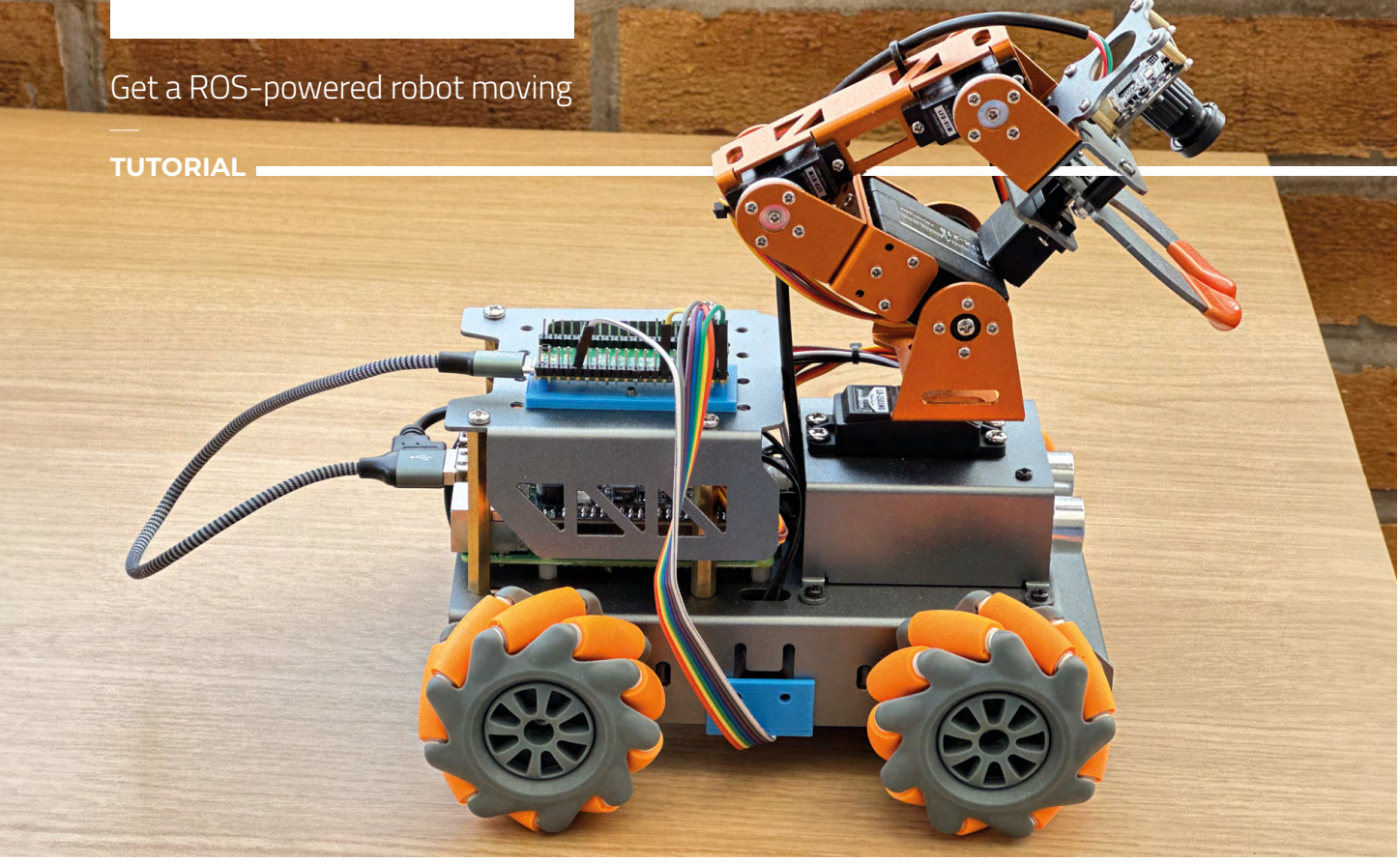
We've used KMK because it's based on CircuitPython, which we're familiar with, but there are other options. QMK is probably the most famous of the keyboard firmware, and there are a lot of things built on top of it, including complex and powerful systems to allow multiple key presses to be interpreted in different ways.

ZMK is another option that's built specifically for wireless keyboards, but unlike KMK and QMK, we've not tested it out fully.

The main part here is the two lists: `_KEY_CFG` and `keyboard.keymap`. These two match up to link a GPIO pin with a key press. You might have to adjust the GPIO pins depending on where your buttons are attached.

Save that **code.py** file to the CircuitPython drive and you should be able to start using your Pico as a very minimal keyboard.

You can use this controller as is, or expand it to have as many buttons as you can cram on. It can be as simple or complex as you like, but the important point is that it can be customised to exactly the way you want to control your computer. Happy gaming!



PART 02

Get a ROS-powered robot moving

Create a drive system for a ROS robot with motor control and movement feedback

Above
The motion sensor clips underneath the robot and is connected via a Raspberry Pi Pico

Below
You can see the controller HAT at the top right-hand corner, second one down

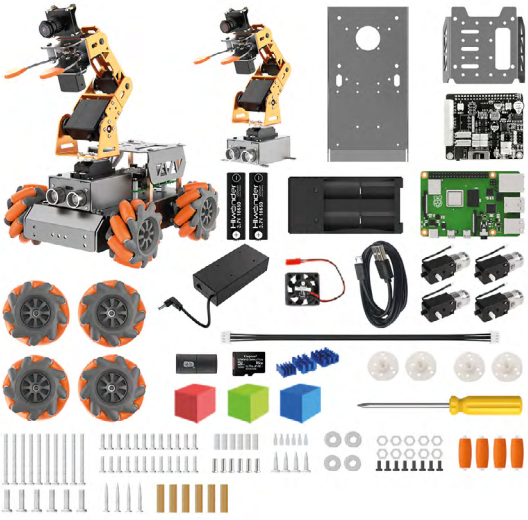
Rob Miles

Rob 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**.

In the last article, we discovered how ROS 2 applications are structured and how components can communicate using publish and subscribe. In this article, we are going to build on this to make the robot move, and then we are going to discover how we can add a sensor to determine the distance and direction of movement.

ROBOT CONTROL

The robot kit is from Hiwonder (**hiwonder.com**). It uses a HAT which fits on the Raspberry Pi driving the robot. The HAT drives the motors and servos and is controlled by Python code supplied by Hiwonder. It works well, but the author was keen to use ROS 2 with the robot, so the first step was to convert the motor driving code into a ROS 2 node with the original name of motors.



The motors node will subscribe to a topic which will be used by any process wanting to tell the robot to move. However, before we can create the motors node, we must look at how the motors are presently controlled by the Hiwonder code.

CONTROL POWER WITH PYTHON

Each of the wheels on the robot is turned by a DC motor connected to the controller HAT. The controller HAT creates pulse-width modulation (PWM) signals to control the power going into the motor. PWM works by rapidly turning the driving voltage on and off to vary the amount of power being delivered. You can get the effect of a dimmed light at home by rapidly turning a light switch on and off (although this wouldn't make you very popular with the house owner). PWM works in a similar way, except that the on and off times are much smaller, and the switching is performed by a MOSFET. The HAT contains circuitry that generates PWM signals for four motors. All we need to do is call the Python **`setMotor`** function to tell this hardware to set the speed of a particular motor. The function accepts two parameters: the number of the motor to be controlled and a power level in the range of -100 (full-power reverse) to 100 (full-power forwards).

```
setMotor(1,50)
```

The above statement would ask **`setMotor`** to turn motor 1 forwards at half power. The motor driver code supplied by Hiwonder is shown in **Figure 1**. This function is in the **HiwonderSDK/Board.py** source file. The first few lines of the function (45–54) validate the incoming motor number and speed value. It is interesting to see how different errors are managed. An attempt to access a non-existent motor (for example, motor number 5) would cause an exception to be thrown and the function will interrupt the program that called it. However, an attempt to use speeds outside the speed range of -100 to 100 would result in the speed being capped at the limit values and the function continuing. In other words, trying to make the motor go at speed 200 would not generate an exception, the function would use speed 100. This can lead to a philosophical programming discussion of the type best held over a beer (Why not fail for all errors?), but we haven't got time for this because we want to make our ROS drive system.

The code from lines 56 to 65 creates an inter-integrated circuit (I2C or IIC) message and sends it to the motor controller to set the power level. If the first attempt to send the message fails, the exception handler is triggered and tries to send the message a second time. If the second attempt fails, the function will fail with an exception.

FLEXIBLE WORKING

The robot we are using is controlled by a Raspberry Pi 4 with 4GB of memory. This is because the robot has a vision system which needs a powerful machine. However, you can run ROS on very small platforms (including Raspberry Pi Zero). You can also run a ROS application over multiple connected devices. A node running on one device can subscribe to messages published by a node running on another. As an example, you could have a simple rover controlled by a Raspberry Pi Zero which is connected over Wi-Fi to a vision system running on a Raspberry Pi 5. ROS gives you great flexibility when deploying your solution, and you don't have to decide where the nodes run at the start.

QUICK TIP

If you have a robot of your own that you want to control with ROS, you can replace the **`setMotor`** function with one that controls your robot hardware and everything else in this article will work.

Figure 1
If you've ever wondered what 'professional' code looks like, this is a good example

```
44  def setMotor(index, speed):
45      if index < 1 or index > 4:
46          raise AttributeError("Invalid motor num: %d"%index)
47      if index == 2 or index == 4:
48          speed = speed
49      else:
50          speed = -speed
51      index = index - 1
52      speed = 100 if speed > 100 else speed
53      speed = -100 if speed < -100 else speed
54      reg = __MOTOR_ADDR + index
55
56      with SMBus(__i2c) as bus:
57          try:
58              msg = i2c_msg.write(__i2c_addr, [reg, speed.to_bytes(1, 'little', signed=True)[0]]
59              bus.i2c_rdwr(msg)
60              __motor_speed[index] = speed
61
62          except:
63              msg = i2c_msg.write(__i2c_addr, [reg, speed.to_bytes(1, 'little', signed=True)[0]]
64              bus.i2c_rdwr(msg)
65              __motor_speed[index] = speed
66
67      return __motor_speed[index]
```

LET'S TWIST AGAIN

Now that we know how the robot motors are controlled, we need to connect these motors to ROS applications. In the previous article, we discovered that ROS nodes communicate using a 'publish and subscribe' model. We built a **`button`** node which published messages when buttons were pressed, and a **`display`** node which subscribed to messages and displayed them when they arrived. Our motor controller will subscribe to a topic and then drive the motors in response to messages published on that topic.

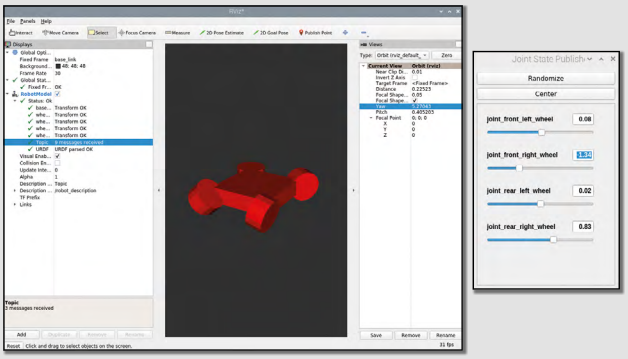
The Twist message in ROS can be used to express the direction we want to move the robot and the amount the robot should rotate when it moves. If we make the **`motor`** node understand Twist messages, it can be used with other robot applications. A Twist message contains two components: the linear velocity and the angular velocity. Both values are expressed as vectors with values for the X, Y, and Z components. This sounds much more complicated than the kind of 'ahead half speed' messages we'd like to use, but it does turn out to be a lot more flexible. →

YOU'LL NEED

- **A reasonably powerful desktop computer, laptop, or Raspberry Pi (preferably a 4 or 5 with 4GB of RAM)** to run the development environment and ROS
- **A Raspberry Pi Pico**
- **A Near Optical Flow SPI Breakout** (shop.pimoroni.com/products/paa5100je-optical-tracking-spi-breakout) if you want to track your robot
- **Your own little robot or the parts to make one**

NICE PLACE TO WORK

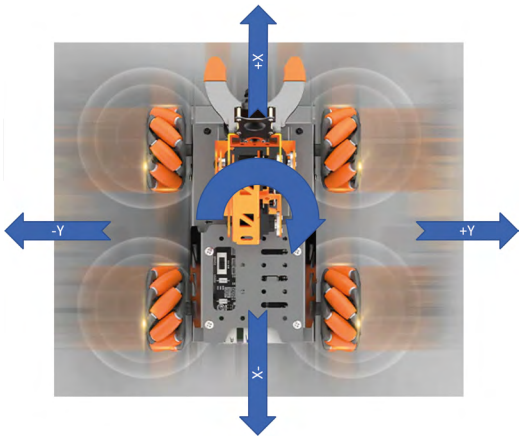
ROS comes with graphical tools that you can use to build and simulate your robot application. Below, you can see the author's first attempt at designing a robot using the Unified Robotic Description Format (URDF). He used the RViz2 graphical tool and the joint state publisher to validate his robot description (which, in this case, needed some work). These tools require quite a powerful machine. The author used a Raspberry Pi 5 with a solid-state drive (SSD) for this part of the project. He then placed the code in a GitHub (**github.com**) repository and the smaller Raspberry Pi in the robot and loaded it from there.



Above
It turned out that the wheels were rotating on the wrong axes…

Figure 2
The Z value in this framework points straight up, i.e. out of the page

Figure 2 shows how the direction velocity values apply to the robot, which is pointing up the page. The X direction is forwards, the Y direction is left and right, and the Z direction is up and down. So, if I want the robot to move forward at one metre per second (which would be very fast), I'd set a linear velocity of (1, 0, 0). Angular velocity is the rate at which the robot should turn while it's moving. The rate is specified in radians per second. Angles can be expressed in degrees (with 360 degrees in a circle) or radians (with 2 * Pi – around 6.28 – radians in a circle). Working with radians is confusing, but it pays off when we start doing maths with the values. If both angular and linear velocities are both non-zero, the robot will move in a particular direction and turn at the same time. We humans call this 'walking round a corner'. Whether or not a robot can move in a requested direction depends on the drive configuration. A car-like robot with steering wheels at the front will have difficulty rotating on the spot (i.e. it could not implement a Twist request with an angular component but no linear component). One of the great things about the mecanum wheel drive system is that it is possible for a robot to move in any direction and turn at the same time. We can use a bit of trigonometry to convert a Twist move request into the speeds each wheel should move:




```
def calculate_motor_speeds(twist):
    # Robot dimensions in meters
    L = 0.059  # Half the distance between front
and back wheels
    W = 0.067  # Half the distance between left
and right wheels

    # Convert twist velocities to motor speeds
    Vx = twist.linear.x
    Vy = twist.linear.y
    Vz = twist.angular.z

    # Calculations based on the mecanum wheel
formulae
    V_FL = Vx - Vy - (L + W) * Vz
    V_FR = Vx + Vy + (L + W) * Vz
    V_RL = Vx + Vy - (L + W) * Vz
    V_RR = Vx - Vy + (L + W) * Vz
    scale = 100.0 / max_wheel_power
    max_wheel_speed = max(abs(V_FL), abs(V_FR),
abs(V_RL), abs(V_RR), 1)
    setMotor(1,int(V_FL * scale))
    setMotor(2,int(V_FR * scale))
    setMotor(3,int(V_RL * scale))
    setMotor(4,int(V_RR * scale))
```

The `calculate_motor_speeds` function above takes in a Twist value and extracts from it the three values that are needed to control the mecanum motors. These are the linear X and Y values and the angular rotation. It uses a bit of maths, which the author doesn't quite understand (but seems to work), to calculate power values for each wheel (Front Left, Front Right, and so on). Then it scales the values so that they are in the range 0 to 100. Finally, it calls `setMotor` for each motor. Now that we have our motor power control sorted out, the next thing we need to do is make a ROS node that can receive Twist messages from other parts of the robot system.

THE TOPIC OF MOVEMENT

The code below describes a **MecanumDriveSubscriber** object which will respond to messages to control the motor power. **The __init__** method runs when the object is created and creates a topic called **cmd_vel** which accepts Twist messages.

```
class MecanumDriveSubscriber(Node):
    def __init__(self):
        super().__init__('mecanum_drive_
subscriber')
        self.subscription = self.create_
subscription(
            Twist,
            'cmd_vel',  # Topic name
            self.listener_callback,
            10)
        setMotor(1,60)
        time.sleep(0.5)
        setMotor(1,0)
        self.get_logger().info("Motor subscriber
running")
```

The __init__ method also makes a motor move briefly. This proved very useful when testing. It was obvious when the drive had been started. The class also contains a method called **listener_callback**.

```
def listener_callback(self, msg):
    calculate_motor_speeds(msg)
    return
```

The **listener_callback** method is called whenever a message is posted to the **cmd_vel** topic. The method just passes the incoming message to the **calculate_motor_speeds** function to move the robot.

TAKING CONTROL

The aim is to make a robot which will decide for itself which way it wants to move. But for testing, it is very useful to be able to control the robot directly from the keyboard. We can make a simple node which checks for key presses and publishes to a topic when it sees them. The code below creates a class called **KeyboardTeleop** which publishes movement instructions to the **cmd_vel** topic.

```
class KeyboardTeleop(Node):

    def __init__(self):
        super().__init__('keyboard_teleop')
        self.publisher_ = self.create_
publisher(Twist, 'cmd_vel', 10)
        self.timer = self.create_timer(0.1, self.
timer_callback)  # Poll at 10 Hz
        self.get_logger().info("Use WASD keys to
control the robot")
        # Save terminal settings
        self.settings = termios.tcgetattr(sys.
stdin)
```

The **__init__** method runs when the object is created. It does two things: it creates a publisher which can send Twist messages to **cmd_vel**; it also creates a timer which is triggered ten times a second. The timer will call a method called **timer_callback**. Let's have a look at the code for the timer callback. →

```
def timer_callback(self):
    key = self.get_key()
    if key == 'w':
        self.get_logger().info("w - forward")
        self.publish_twist(1.0, 0.0)  # Forward
    elif key == 's':
        self.get_logger().info("s - backward")
        self.publish_twist(-1.0, 0.0)  # Backward
    elif key == 'a':
        self.get_logger().info("a - left turn")
        self.publish_twist(0.0, 1.0)  # Left Turn
    elif key == 'd':
        self.get_logger().info("a - right turn")
        self.publish_twist(0.0, -1.0)  # Right
Turn
    elif key == ' ':
        self.get_logger().info("space - stop")
        self.publish_twist(0.0, 0.0)  # Stop
    elif key == '\x03':  # CTRL-C
        self.destroy_node()
        rclpy.shutdown()
```

DOCKER IS WONDERFUL

Docker (**docker.com**) is not a robotics tool. Docker is used to deploy large and complex applications. You create a 'Docker image' file which contains all the components you need for an application, and this image runs on your machine inside a software object called a 'Docker container'. The container sits on top of a layer which hides details of the underlying system. This makes it possible to deploy a ROS solution (which would otherwise need a particular version of Linux) on a standard Raspberry Pi. Learning a bit about Docker is a great idea; you can use it to deploy and manage all kinds of software.

The author has created a Docker image which contains ROS and a selection of useful tools that you can use with the examples in these articles. He has written a guide to setting up a Raspberry Pi, installing Docker, building a Docker image and running it, along with instructions for the exercises in this article. You can find it all here: **hsmag.cc/RosIntro**.

The function uses a method called **get_key** to get the currently pressed key. It then selects a movement option and calls a function called **publish_twist** for each movement.

```
def publish_twist(self, linear, angular):
    msg = Twist()
    msg.linear.x = linear
    msg.angular.z = angular
    self.publisher_.publish(msg)
```

The **publish_twist** method takes the linear and angular values and uses them to construct a Twist value (called **msg**) which is then published. This node can be used to drive the robot from the keyboard. If we start these two nodes running, we can use the keyboard to steer the robot around.

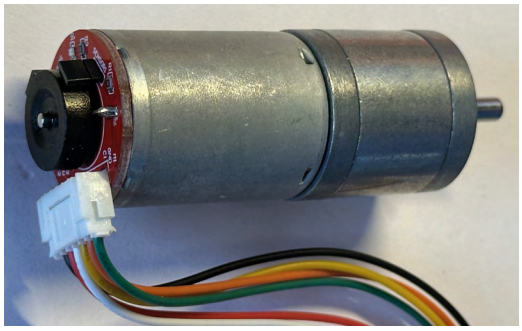


Figure 3
The black disc on the left of the motor contains a magnet which moves past the two sensors when the motor turns

Figure 4
The sensor is fitted with two LEDs to illuminate the scene viewed by the camera

KEEPING IT REAL

We've seen that we can put different power values into motors to make them turn at different speeds. We've also found a way to take a movement request and convert it into speeds for each wheel. However, we have no way of mapping a requested speed onto a value to be sent to the motor controller. We know that putting a bigger number into the **setMotor** function will make the motor turn faster (and a value of 100 makes it run full speed), but we don't know how fast the motor moves. This works fine for remote control as the system above shows. It is quite fun to steer the robot around. However, we'd like to be able to make the robot move a particular distance, and at the moment this isn't possible.

We could start to solve this problem by obtaining the speed of the motors when our system turns them. For example, if we observe that a motor turns 60 times in one minute at full speed, that would mean the motor turns once a second. The wheel has a diameter of 65 mm. The circumference of the wheel would be Pi * 65 mm (around 190 mm). This means that at full speed, the wheel would move the robot 190 mm every second. Unfortunately, this would not work well in practice because the motor slows down when it is under load.

Figure 3 shows a motor which has a shaft encoder. The encoder sends pulses when it detects motor movement. We could use this signal to determine exactly how much a motor is turning and work out how far a wheel is moving the robot. Unfortunately, the motors in the Hiwonder robot don't have shaft encoders, so there is no way that the robot control software can know how far the motors have turned. The author decided to add a sensor to the robot to detect movement. This will provide feedback to a ROS application which can then adjust the movement commands. The sensor he has chosen is called an 'optical flow sensor'. It works in the same way as an optical computer mouse. The camera on the sensor detects changes in position of the surface beneath the robot and transmits X and Y values to a connected computer.

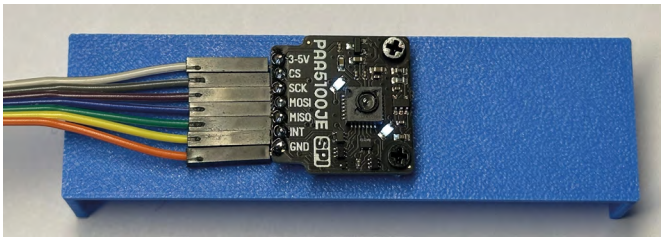
GETTING INTO THE FLOW

Figure 4 shows the selected sensor. The sensor has been mounted on a bracket which will be clipped underneath the robot. When the robot moves, the sensor will send movement information, giving the direction and speed of the movement it has detected. This information will be broadcast so that control software can monitor the robot's position.

ENTER THE PICO

The sensor uses an SPI connection to transmit movement values. There is no easy way to connect the sensor to the motor control HAT on the Raspberry Pi on the robot, so instead, a Pico was used to collect movement information and transmit it over a USB serial connection into the robot. The Pico was then plugged into a USB port on the Raspberry Pi. This worked well and means that the sensor can be easily added to any robot with USB connections.

Figure 4 shows how the sensor and the Pico are connected together. The following Python program was adapted from the example code for the sensor and runs inside the Pico to send movement information to the robot.



```
import time
import json
from machine import Pin
from breakout_paa5100 import BreakoutPAA5100 as
FlowSensor

flo = FlowSensor()
flo.set_rotation(FlowSensor.DEGREES_0)
led = Pin(25,Pin.OUT)

while True:
    delta = flo.get_motion(timeout=0.1)
    if delta is not None:
        x = delta[0]
        y = delta[1]
        result = { "x":x, "y":y }
        led.value(1)
        print(json.dumps(result))
    time.sleep(0.1)
    led.value(0)
```

This code runs inside the Pico (not on the robot). It connects to the sensor and then repeatedly reads it. If the sensor has new data, the values of x and y are assembled into a JSON message and printed to the console (which is the serial output for the Pico). A device connected to the sensor via the serial port receives the movement messages. Note that for this code to work, your Pico must be running the MicroPython image provided by Pimoroni. You can find it here: **hsmag.cc/PimoroniPicoUF2**.

ROBOT TRACKING

We now need a node which can receive the movement messages from the Pico and publish the values for a `controller` node to use. This will be a bit like the `keyboard` node we have just seen, except that rather than checking for key presses, the `timer_callback` method checks for messages from the movement sensor.

```
def timer_callback(self):
    if self.serial_port.in_waiting:
        data_str = self.serial_port.readline().
decode('utf-8').strip()
        try:
            data_json = json.loads(data_str)
            x = float(data_json.get('x', 0))  #
Convert to float
            y = float(data_json.get('y', 0))  #
Convert to float

            twist = Twist()
            twist.linear.x = x
            twist.linear.y = y
            # Assuming z, angular x, y, and z are 0
or set them as needed
            self.publisher_.publish(twist)
            self.get_logger().info(f'Publishing:
{twist}')
        except json.JSONDecodeError:
            self.get_logger().error('Could not
decode JSON from serial data')
        except ValueError as e:
```

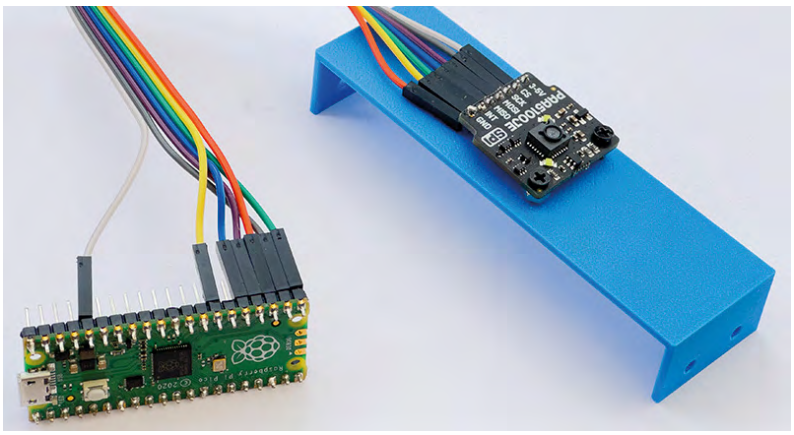
The code above shows the `timer_callback` function for the motion sensor node. It checks to see if there are any characters available from the serial port and reads a line if there is. Then it decodes the JSON and extracts the x and y values from it. These values are then used to create a Twist value which is published. A process wanting to track the robot movement can just subscribe to the events and be informed when the robot moves.

TAKING CONTROL

The author has created a drive package which contains the `keyboard`, `motor`, and `motion` nodes which can move the robot. He is presently working on a `controller` node which will use these to move the robot particular distances. He hopes that these articles have served as a useful introduction to the lower-level components of ROS. If you want to refine your robotics skills, you can work through the exercises on the project GitHub page, which you can find here: **hsmag.cc/RosIntro**.

POWER FOR TESTING

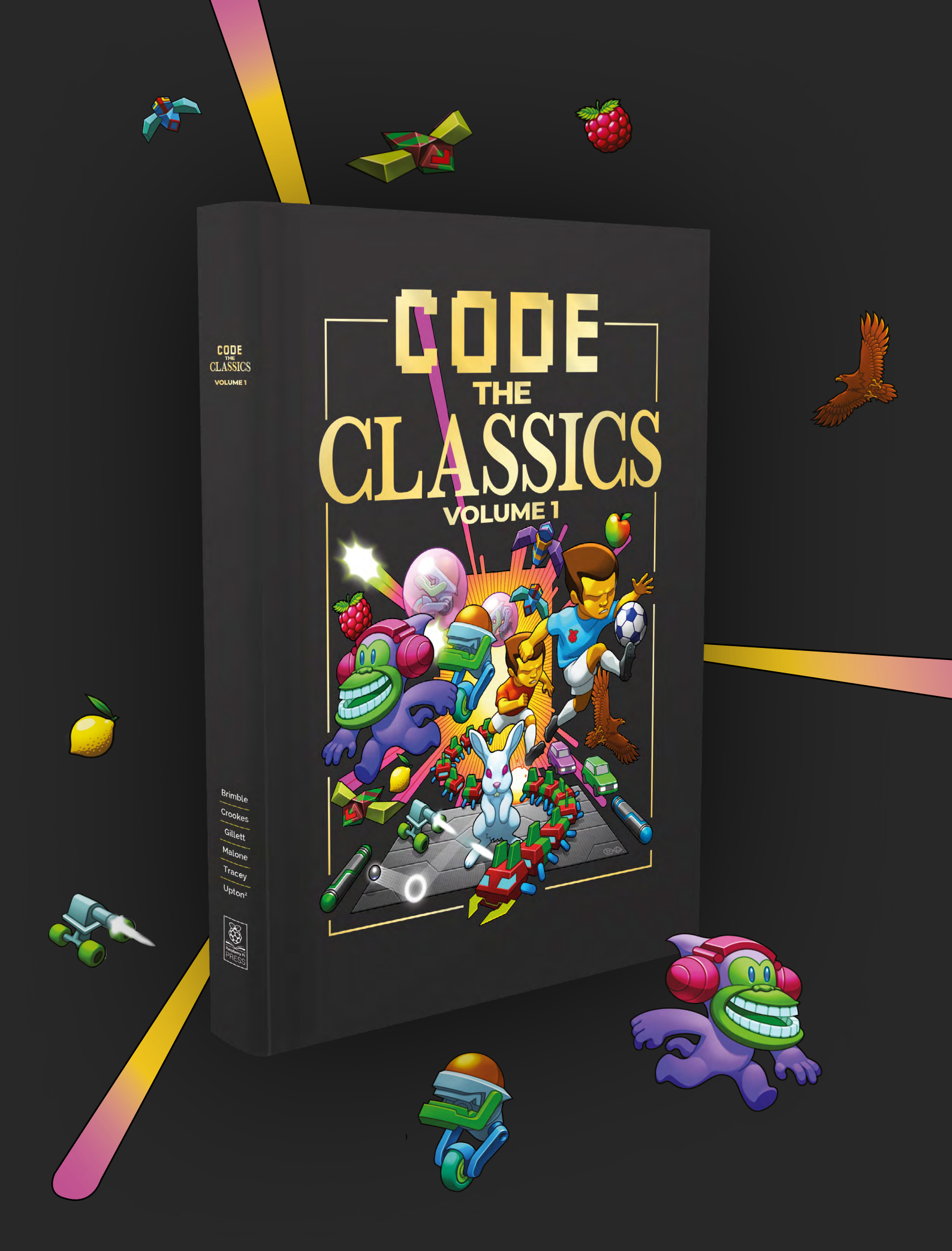
The robot is battery-powered and runs for around half an hour on the enclosed batteries. The author added a speed limiter to the `setMotor` function so that he could restrict the motor power and allow the robot to run on a mains adapter with less power capacity than the batteries. This proved very useful when developing the code.



Above
All of the connections are on one side of the Pico

QUICK TIP

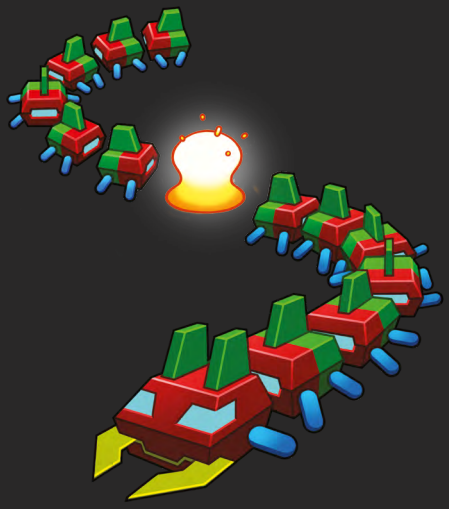
When creating code that talks to hardware, it is useful to make something happen when the code is working. The sensor reader flashes the LED on the Pico each time it sends a movement message. This was very useful when debugging.


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²
Raspberry Pi PRESS

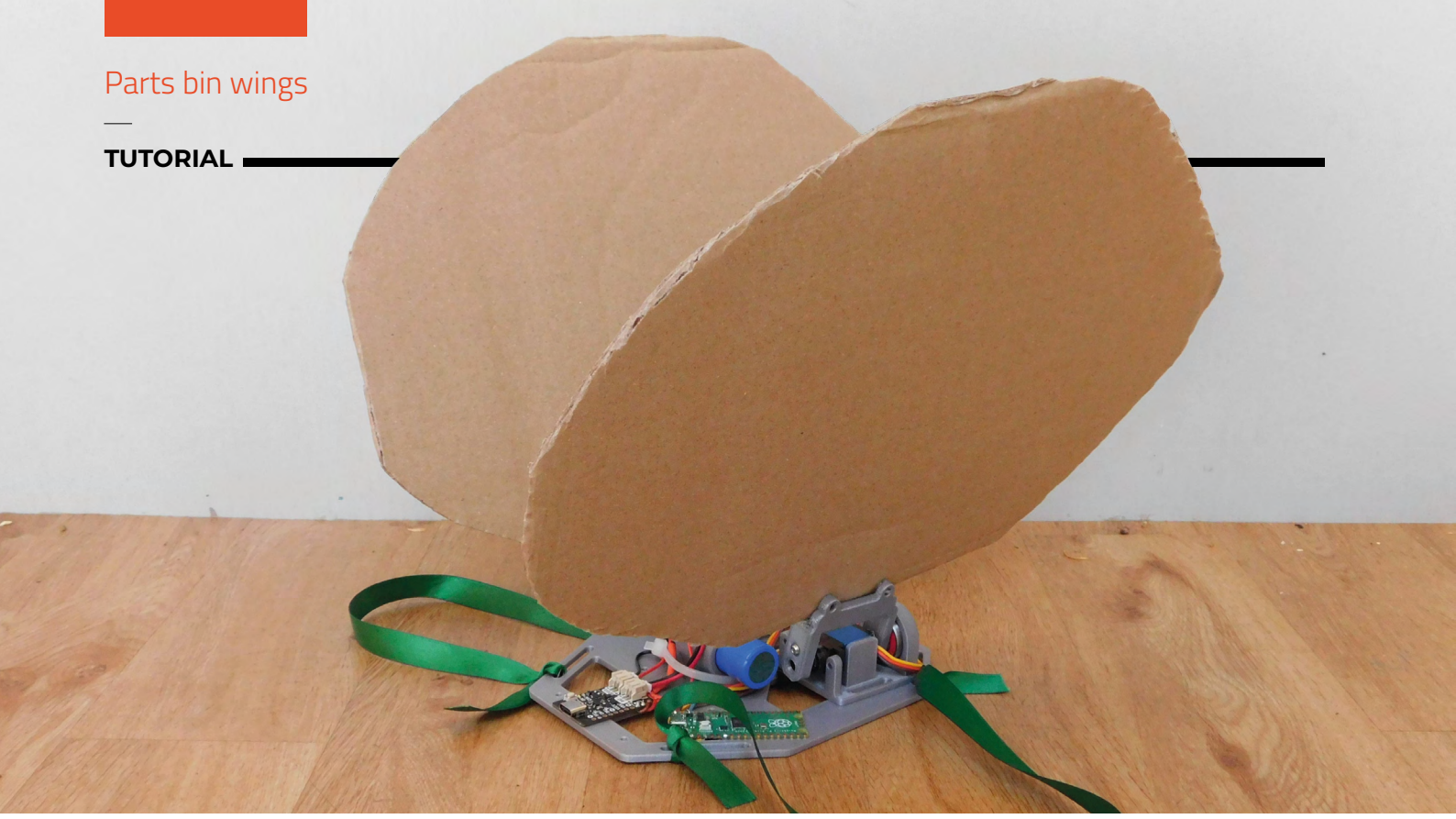
CODE THE CLASSICS VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.

- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**



Parts bin wings

We shamelessly adapt an existing design with the parts we have on hand

Ben Everard

Ben spends more time making games controllers than he does playing games. It's a strange hobby, but he enjoys it.

Above
Swap out the wings, and this is a prop that can be recycled for many costumes

We saw an excellent tutorial from Adafruit on making some animatronic wings: hsmag.cc/AdafruitWings. They looked awesome! The only problem was, we didn't have the parts needed to make them. Obviously, we could have ordered all the parts we needed from those good folks in New York, but we did have a lot of similar parts. We wanted to know, could we reproduce the wings using just what we had hanging around the workshop left over from previous projects?

Re-using stuff you have is a great practice to get into. It's cheaper, better for the environment, and means you can make stuff now rather than waiting for a delivery (and this can be key to finishing a project rather than ending up with a pile of half-finished projects).

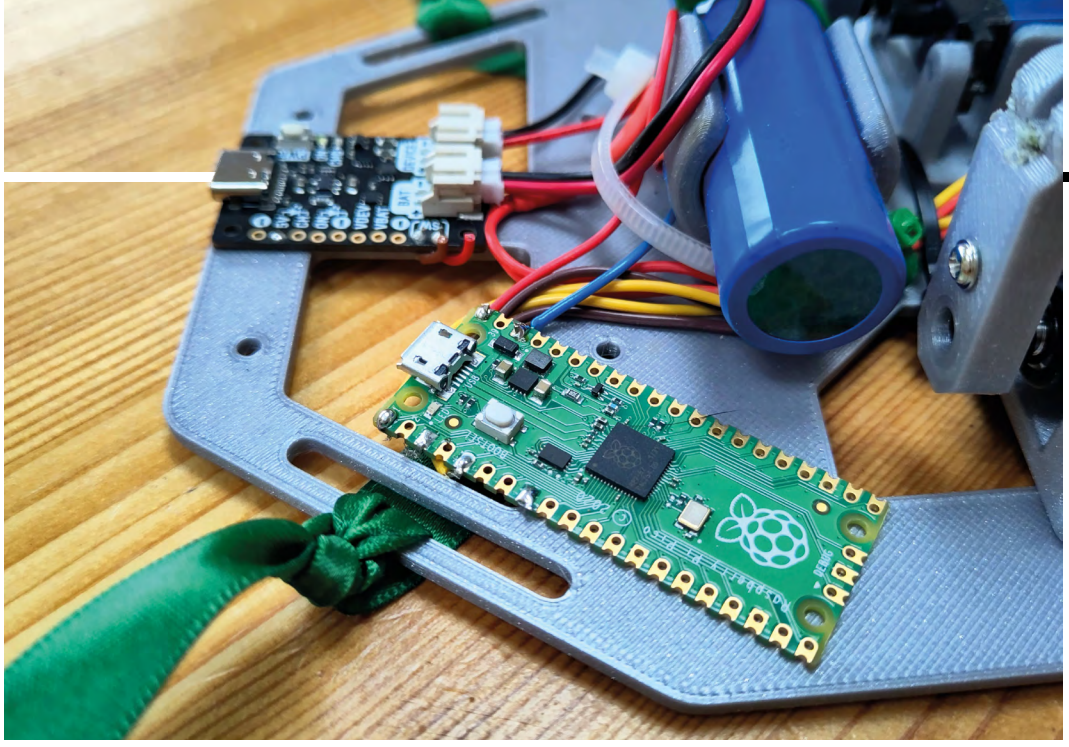
It does sometimes mean suboptimal results – we'll look at some of the areas where our wings aren't as good as the originals. However, as the saying goes, done is better than perfect. We set ourselves the challenge of building this without buying anything new. Obviously, you won't have the same bits and pieces lying around, but we'd encourage you to use what you have and improvise where necessary.

A SOLID BASE

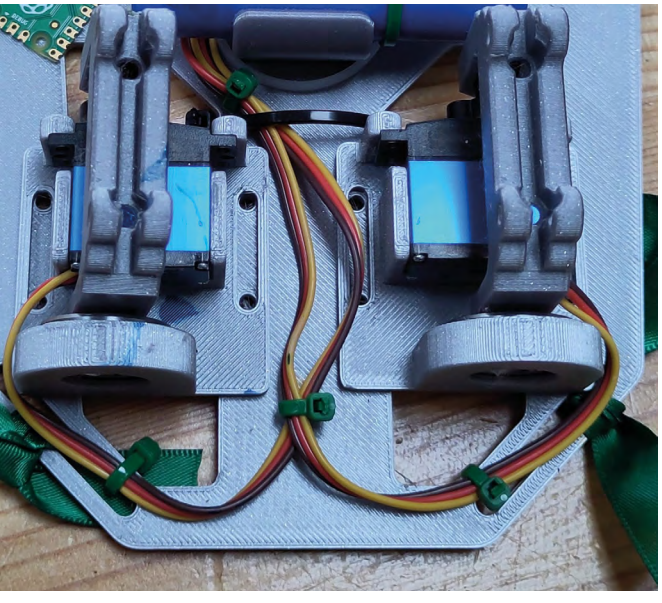
We won't go through the whole build as the base is documented well on the Adafruit website. We'll focus on the parts where we've differed from Adafruit.

At the core of the project is a pair of servos, and we had some the exact same size, so we were off to a good start. The other parts we had that matched exactly what was used are a pair of 608-ZZ bearings (we used them to make a filament spool holder and had a few left over) and a lithium battery (that has been through a few projects already, and regular readers will recognise from an extremely janky pair of headphones).

That's not a bad start. However, we didn't have a Feather M4 Express, a Servo FeatherWing, a FeatherWing Doubler, a PowerBoost Charger, or any of the nuts and bolts used.



The first thing we had to work out was if we could mount the servos, as the servo arms designed by the Adafruit team screwed into the bearings. After a bit of fiddling, we worked out that rather than attach these arms to the bearings with a machine screw, we could simply add a protrusion to the arm. This had the downside of forcing us to print the arm in a less favourable orientation, which both meant additional support material and layer lines in a weaker position. However, in practice, they printed OK and seemed strong enough.



With the servos in place and working, it was time for our next hardware flip – changing the Feather M4 for a Raspberry Pi Pico. The short version of this story is that it just worked. The longer version is that it's actually really impressive that you can switch controllers like this. While both the Feather M4 and the Raspberry Pi Pico are built on Arm CPU cores, they're built on different Arm cores with completely different things around these Arm cores. The fact that it works so easily on different microcontrollers is thanks to the work of the CircuitPython developers.

We swapped the PowerBoost board for a Pimoroni LiPo Amigo board. This was swapping a similar component for a similar component. The biggest impact of the microcontroller and power board swap is that they no longer fitted on the mounting plate. At this point, we had two options: design a new mounting place, or hack it together. We opted for the latter of these two options. Our workshop is currently devoid of any machine screws below M4, so we couldn't have built it without ordering more hardware anyway, and that's against our self-imposed rules.

We now have a bunch of parts and no nuts and bolts to join them. We do, however, have a tube of superglue (also known as Krazy Glue or cyanoacrylate). Sometimes the easiest solution is perfectly acceptable. Yes, superglue probably isn't as secure as bolting everything together, but then the wings are by far the weakest part, so it's really not the end of the world if the base is slightly weaker than it could be. Gluing them on does mean that the cables are a bit more exposed than they could be, so we used plenty of cable ties to tie the cables down and keep things as tidy as possible.

That's just about all there is to it. We didn't add the potentiometer, because we didn't really need the features it offered.

It would have been slightly stronger and noticeably more tidy if we had built it as it was designed, but then it's strong enough and tidy enough as it is, and with many builds, speed is of the essence. The quicker you can complete something, the more likely you are to see it through to completion.

Waiting for parts to arrive is a recipe for distraction and waning enthusiasm. At least, that's our experience (and our excuse).

Above
We're not completely sure if this Raspberry Pi Pico will come off the glue without breaking, but given that, at £4, it's about the same price as a delivery of bolts to screw it on, we're willing to take the risk

Below Left
We were late arrivals to the world of cable ties, but now keep some in different sizes on hand because they're incredibly useful

DON'T MISS THE **BRAND NEW** ISSUE!

SUBSCRIBE FOR JUST £10!

- **THREE!** issues of The MagPi
- **FREE!** Raspberry Pi Pico W
- **FREE!** delivery to your door

+ FREE RASPBERRY PI PICO W*

Three issues and free Pico W for £10 is a UK-only offer. Free Pico W is included with a 12-month subscription in USA, Europe and Rest of World. Not included with renewals. Offer subject to change or withdrawal at any time.

magpi.cc/subscribe

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

ONLY THE BEST

DIY kits

A collection of fun kits to build

By **Marc de Vinck**

Kits! Anyone who knows me, knows I love kits. It's how almost everyone gets into soldering and electronics. I manufactured electronic kits for many years and, at times, I really miss it. I loved the process of prototyping and figuring out what people would like to build. And even though I don't create and sell kits any more, I do still love to pick some up and spend some time building other people's creations. In this Best of Breed, I will be looking at some kits that are not only fun, but useful too. That second part, usefulness, is often overlooked. I really enjoy building something and then being able to use it for a long period of time afterwards. Not just blinky little LED things, although those are fun too! But I'm far more inclined to pick up a kit that, after I've enjoyed building it, I can plug it in, turn it on, and get some longevity out of the labour of love.

So, let's jump right in and take a look at a few kits that I've built before, and a few kits that I have on my list to build in the future.

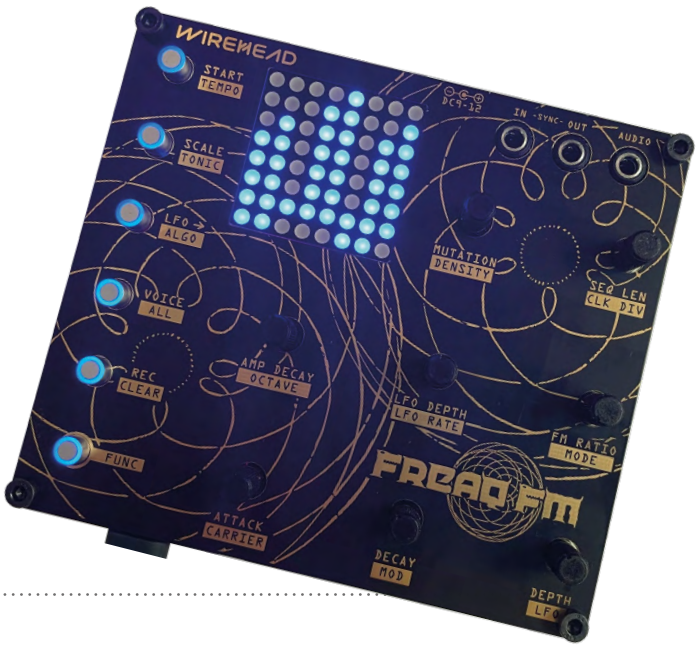
Freaq FM Digital Synthesizer (DIY Kit) vs PDP-11 replica kit: the PiDP-11

TINDIE $135 | tindie.com

TINDIE $297 | tindie.com

The Freaq FM is a desktop 8-bit digital synthesizer designed by Wirehead Instruments in Australia. It features dual two-op FM voice architecture and multiple waveforms, and an LFO and modulation envelope. So, what does that mean? It means that you can make some really cool-sounding distortions and electronic music. And you can even sync it with external instruments.

You can buy this as an assembled synthesizer or do what I would do, which is buy the kit. Building a synth is a very popular DIY project, and one that I highly recommend everyone do. Be sure to head over to the product page, which includes more information and a great demo of what this unit can do. After watching the video and listening to the sounds, I'm sold! It's awesome!



The PDP-11 replica kit from Obsolescence Guaranteed, located in Switzerland, is an amazing piece of hardware history that would be incredibly fun to solder together.

The original PDP-11 computers were sold by Digital Equipment Corporation from the start of the 1970s all the way through to the late 1990s. Yes, the 1990s!

In fact, from my limited research, it seems like some of these are still in use, which is kind of amazing considering how powerful a Raspberry Pi is, never mind how inexpensive.

Behind the scenes of this replica version is a Raspberry Pi that runs the system, but you wouldn't know, considering all the amazing retro-looking controls in the front.

The PiDP-11 runs all the original operating systems, including UNIX systems 5/6/7, 2.11BSD, RSX-11, RSTS-E, and RT-11. And it can simulate many different types of peripherals.

If you like retro computing, go ahead and check out this kit!

VERDICT

Freaq FM Digital Synthesizer

Ideal for makers who make music!

10/10

VERDICT

PDP-11 Replica kit: the PiDP-11

A truly unique kit.

9/10

BEST OF BREED

Cordwood Puzzle Too

PIMORONI $11.75 | pimoroni.com

Now this is a strange kit! Certainly not for the faint of heart. Unlike most kits that I love, the Cordwood Puzzle Too by Boldport doesn't include any instructions. That's part of the fun, or frustrating part, depending on your viewpoint.

In the 1950s and 1960s, engineers saved space by using what is called the Cordwood assembly method. Components were sandwiched, quite often vertically, between two circuit boards. As surface-mount technology advanced, and components got smaller, this type of circuitry faded away. Boldport created an inexpensive and fun little demonstration of this period technique. If you're up for a challenge, this might be a good kit for you. But if you're new to soldering, I'd hold off on picking this one up until you learn the basics. Head over to the website to learn more about this unique puzzle kit, and some hints on how to assemble it.

MintyBoost Kit – v3.0

ADAFRUIT $19.50 | adafruit.com

The MintyBoost from Adafruit is a must-have kit for anyone getting started with soldering. I have bought over 100 of these kits to teach soldering over the years. People are so amazed at how easy it is to build, and how interesting the electronics are that make it work. Yes, you can charge your phone from two AA batteries! The latest version of the kit provides 500 mA at 5 V. It works with almost anything that requires 5 V power. It's a perfect example of kits that are fun to build, and have longevity. Head over to Adafruit to learn more about this fun little kit.

VERDICT

Cordwood Puzzle Too

Interesting and beautiful.

9 /10

VERDICT

MintyBoost Kit - v3.0

Why buy when you can make it?

10 /10

unPhone – all-in-one LoRa, Wi-Fi and BT dev device with touchscreen and LiPo battery

PIMORONI $147.64 | pimoroni.com

Build your own phone? Sort of! The unPhone is an IoT development platform that is based on the very popular ESP32-S3 microcontroller. It's an open-source project developed by the University of Sheffield, in partnership with Pimoroni and Gareth Coleman. So, what does it do exactly?

It's basically an all-in-one radio unit that features LoRa, Wi-Fi, and Bluetooth technologies coupled with a LiPo battery and LCD touchscreen.

It also includes an SD card reader, accelerometer, compass, and vibration motor for haptic feedback. It won't exactly replace your phone, but you can send messages over long distances, or create a wireless IoT device.

VERDICT

unPhone – all-in-one

Powerful and versatile.

9 /10

BEST OF BREED

Noisette Optical Theremin kit

TINDIE $44 | tindie.com

I love theremins! I have built several over the years. The Noisette by Zeppelin Design Labs is an optical theremin, my favourite type, that looks to be very easy to assemble and fun to use. If you've never heard of the theremin before, just imagine those old black-and-white horror movies with their creepy and spacey soundtracks.

The Noisette features two optical sensors, allowing you to control pitch and volume. There is also a master volume knob and a wave shape selector. You can create square waves, sine waves, or a mix of either. It also features an on-board speaker for ease of use. This looks like a great kit for teaching anyone basic electronics and learning how to solder.

VERDICT

Noisette Optical Theremin kit

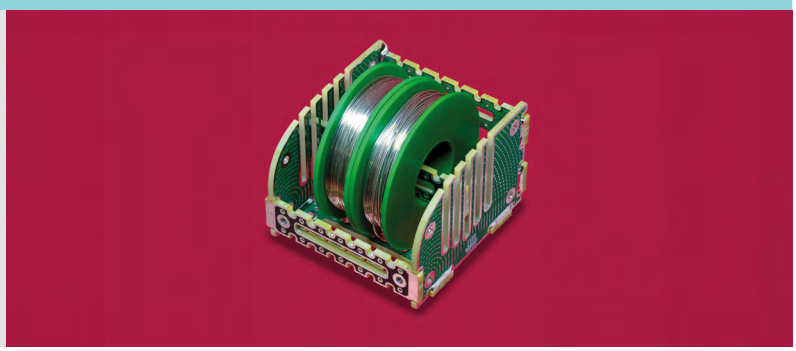
Fun to build and play.

10 /10

SPOOLT

PIMORONI $8.32 | pimoroni.com

Who says you can't solder together some tools too? Typically, you solder electrical components, but in this kit from Boldport, you solder together a solder spool holder. It's a clever use of PCBs. It's also something everyone needs! Pick one up and get ready to build a fun little conversation piece for your electronics workbench.



M5Stack Card Computer

Bridging the line between microcontroller and computer

M5STACK ◈ **$29.90** | **hsmag.cc/cardputer**

By **Ben Everard**

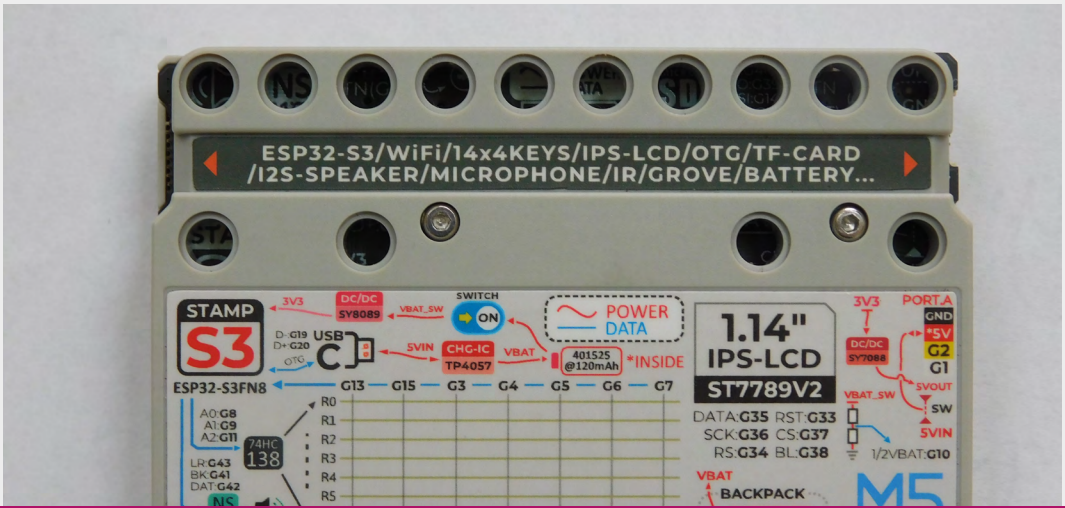
The Card Computer, also known as the Cardputer, is an expansion of the M5StampS3. This base board has an ESP32-S3, which has a dual-core Tensilica processor running at up to 240MHz with 512kB of RAM, 8MB of flash, and Wi-Fi and Bluetooth connectivity.

Onto this removable board is a 56-key keyboard, 1 W I2S amp with speaker, 1.14 inch IPS screen, microphone, and 1400 mAh battery. You can also add more hardware via a Grove connector.

All of this comes in a sturdy plastic case which features Lego-compatible mounts and magnets for attaching to, well, anything ferrous. This is a huge amount of hardware for the price, so it should come as no surprise that it does have limitations: the 240 × 135 pixel screen is pretty small for most purposes; the buttons are hard to press, and the speaker is quite tinny. The hardware isn't bad, but each piece is very basic. The advantage comes, not in any one piece of it, but in having it all together in a single, affordable device.

The ESP32 packs a big processing punch – it's the most powerful of the common microcontrollers. It can, for example, emulate a Game Boy (Espeon: **hsmag.cc/cardputergb**). Given the processing power of this board, and the fairly minimal interfacing options (just two GPIO pins on the Grove connector), it really feels more like a computer than a microprocessor board.

While it doesn't have an operating system in the conventional sense, some of the firmware can launch different apps, and fulfil the role of a rudimentary OS. One slight note of caution to consider here: while 512kB of RAM is a huge amount by most microcontroller standards, many other ESP32 boards have additional PSRAM, which the Cardputer does not. In some memory-hungry



Right ◈
Any spare space is filled up with details of how the hardware is connected up

Left
56 buttons is a huge number for a computer this size

applications, you might find it more limited than other options.

You can program the Cardputer via the ESP-IDF, or Arduino. The example code is a bit lacking, and it's not the easiest microcontroller to get started with – so we wouldn't recommend this for beginners – but if you've got a bit of experience with embedded programming, it should not present too many problems.

Fortunately, though, if you set out programming the Cardputer, you're not on your own. There is an emerging community around this device that is based around a subreddit (**hsmag.cc/cardsr**), and a Discord server (**hsmag.cc/carddisc**). They've created a few useful additions to the official firmware, including a MicroPython build (and app launcher called MicroHydra (**hsmag.cc/cardhydra**), a binary app launcher (**hsmag.cc/cardlauncher**), and some modifications to the official firmware (**hsmag.cc/cardextended**).

FORM OR FUNCTION?

The big question, in our mind at least, is what is the Cardputer for?

We're a bit perplexed that the makers added Lego-compatible mounts and magnets, but not mounting holes. This means there's no way of bolting it down to anything. The 56-key keyboard obviously gives you a lot of buttons to press, but they're not particularly nice buttons and, with such a small screen, it's hard to do much useful text entry.

There are a few niches that this could fit into well. The Grove-compatible connector means that you can plug in some sensors and it could be a bit of a Swiss Army knife for data collecting. It could also be useful for running Python scripts on the go if you need to interact with some networked hardware. Given its slightly cyberpunk look, we could definitely see it fitting into some cosplay outfit or prop, but the lack of a secure mounting option makes that a bit more tricky than it needs to be.

It could certainly be a fun project for becoming more familiar with the ESP32 microcontroller in general, or the M5Stack specifically. The board at the heart of the Cardputer is removable, so if you prototype something that needs only a portion of the hardware, you could pop it out and just connect up what is required.

In all these cases, it feels a bit like we're making up a potential project because, in truth, we like the Cardputer. What it lacks in utility, it makes up for in fun. Perhaps this is simply because this reviewer has done too many orbits around the sun, but the Cardputer seems like exactly what he thought the future would be like when he was younger.

In many ways, it's like a 1980s home computer but shrunk down to miniature size, and that was what we thought technological progress would be like. Yes, almost all the examples we've seen could be done better by a phone app, but somehow, that just wouldn't be the same.

We don't want to second-guess your projects, so you'll have to work out for yourself if it can be useful to you. Some products, though, are worth it simply because they make you smile, and for us, the Cardputer is one of these. It might not be the most practical microcontroller development board around, but we love it anyway. ◻

VERDICT

It's a fun gadget, but we're not sure what to use it for.

9 /10

Bangle.js 2

ESPRUINO ◈ **£76.80** | **shop.espruino.com**

By **Ben Everard**

There are two sides to any bit of electronics: the hardware and the software. On the Bangle.js 2, both are worth a close look. There is no such thing as perfect hardware – especially on a device as constrained as a smartwatch. There is simply hardware that makes trade-offs – for some users, those trade-offs will be good, and for others, they won't.

Any smartwatch has to find the right balance between features and power consumption. By far the hungriest bit of kit on a smartwatch is the screen. Producing light takes power. A 60 mA backlight might be insignificant to a mains-powered device, but it'll kill a watch in just a few hours. However, a bright, crisp display needs these photons. The Bangle.js 2 keeps this part of the power budget low by using a 176 × 176 3-bit colour transflective display. Transflective means that it's lit by reflected light rather than a backlight, and, in normal light conditions, you can read it without the backlight enabled. Having this display means that the battery life is measured in weeks rather than days. Exactly how many weeks depends on what you're doing with it, but it's likely to be in the range of one to four.

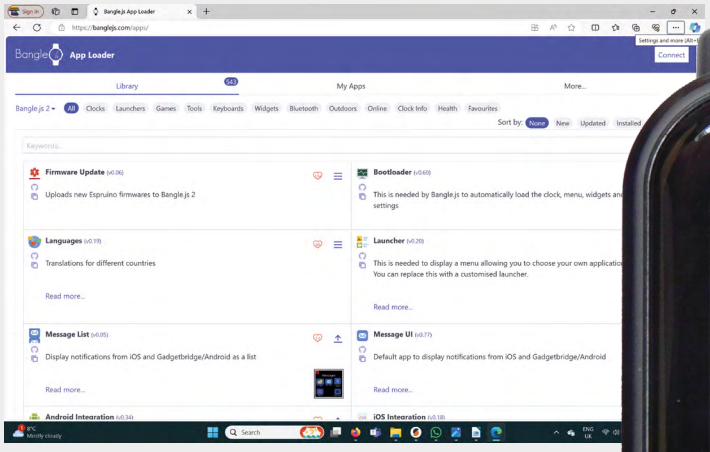
Strapped onto this screen is an impressive range of sensors, including GPS, heart rate, and barometric pressure. A 64MHz Nordic microcontroller with an Arm Cortex-M4 core pulls all the data together and keeps the screen ticking. This is a reasonably powerful microcontroller that should be able to keep up with most of your needs. There's a bit of a delay in starting apps, but not enough to annoy us.

This reviewer is really impressed by the hardware, mainly because of how much software it keeps running and how little battery power it uses. However, there is another take – the screen isn't as bright and crisp as some others. The heart-rate monitor is a little finicky, and the GPS can be slow.

For us, the trade-off of having good battery life is well worth these aspects, but others may prefer a different trade-off.

A SOFTER SIDE

While the hardware is a good proposition, it's the software that brings this watch to the attention of this magazine. The entire software stack is designed to be programmable and is based on the open-source Espruino framework. This means that everything is written in JavaScript – a language that seems to delight as many people as it horrifies. JavaScript was originally a language used to add features to web pages, where it sat between two asynchronous and unpredictable things: people and web servers. The way it developed here – with a strong emphasis on event-based programming – means it's a good fit for some other cases, including where low power use is essential. It can take a bit of time to get your head around event-based programming if you're used to a more traditional paradigm, but it's a good fit for watches (and web pages).



Left
Less excessively 1990s clock faces are available for people whose tastes have progressed since *Saved by the Bell*

Far Left
Apps are delivered to the Bangle directly from a Web Bluetooth-capable browser

There is an app store for Bangle.js 2 at **espruino.github.io/BangleApps** that can load apps directly onto your watch via Web Bluetooth. At the time of writing there were 543 apps available, but many replicated functionality. There are apps for most core functions of heart-rate monitors, GPS trackers, and phone notifications, but beyond this, it's a little sparse. However, the point of an open smartwatch isn't that there's lots of apps available, but that it's easy to create your own.

The software stack is open by design, so you can access all of the watch's hardware without artificial limits, and there's plenty of documentation to get you started. All of the apps in the store are open and linked via GitHub, so there's a lot of code that you can look at for inspiration.

While many smartwatches are intimately linked to smartphones, the Bangle.js 2 is different. It can be linked to a smartphone – and has a companion app for both Android and iOS – but it's fundamentally a standalone device and doesn't need the phone for any of the core functions. The main thing you'd want the phone for is for receiving notifications, or transferring data. The latter can be done with a computer and web browser instead.

Overall, we think this is a great, hackable watch that many people will get a lot of joy out of. The part of the watch we dislike the most is the interface. There's a single button on the side which can be used for short or long button presses. Additionally, there's a touchscreen, but given the small size of the screen, we found it a bit clumsy. We wouldn't say it was hard to use, but it never quite felt natural to us.

However, we absolutely love the battery life and daylight-readable screen. Coupled with the open-source software, this is definitely a smartwatch we can see ourselves using for a long time. We've played around with lots of hackable smartwatches over the years (including the original Bangle.js), and we can confidently say that this is our favourite by a significant margin. ◻

VERDICT

Open, hackable, lasts for ages on a single charge – that's a great combination.

10/10

CROWDFUNDING NOW

The Power Mole

Fire electrons through your windows

From $59 | **hsmag.cc/powermole** | **Delivery:** May 2024

Outdoors projects are great fun, especially once the skies darken in autumn and LEDs shine. Whether it's Halloween props scaring passers-by, winter solstice lights shining, or Christmas decorations dazzling, there's a great run of outside fun over the darker months. However, what's not fun are draughts caused by cables running through open windows.

The Power Mole solves this by letting you transmit electricity directly through glass up to 30 mm thick. This might sound like witchcraft, but it's exactly the same sort of witchcraft that lets you charge your phone without wires. You pop one side of The Power Mole on the inside, and plug it into a normal power outlet. Then you fix the other part to the outside of the window. There's a USB socket that you can plug anything you like into. It provides up to 10 watts at 5 V. This solves a very real problem that we have, and we suspect a lot of other makers do as well. However, we won't be able to try this out as it only ships to the USA. Our readers across the pond can test this out, but we'll have to wait until they open up international shipping.

Left
This is chunky and not particularly aesthetic, but it does solve a very real problem that we face

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.

Invisibility shield

Hide from your enemies

From £54 | **hsmag.cc/invisishield** | **Delivery:** Dec 2024

Invisibility shields, cloaks, and vehicles have been staples of science fiction for almost as long as there has been science fiction, and now, thanks to the Invisibility Shield Company, you can buy one.

This invisibility shield works with some clever optics we don't really understand, but we think it basically comes down to blurring things that are close behind the shield while less blurring things further away. While we haven't tried one out, it's probably no coincidence that the most impressive example photos all have bold horizontal stripes in the background.

There is obviously the concern about nefarious uses of such a device, but fortunately, it's a bit too obvious that something is there for this to be a real concern. This risk is further reduced because it's not see-through in the other direction. In other words, if you can't see the person hiding behind the shield, they can't see you.

Basically, we have no idea what this is for, but also, we want one.

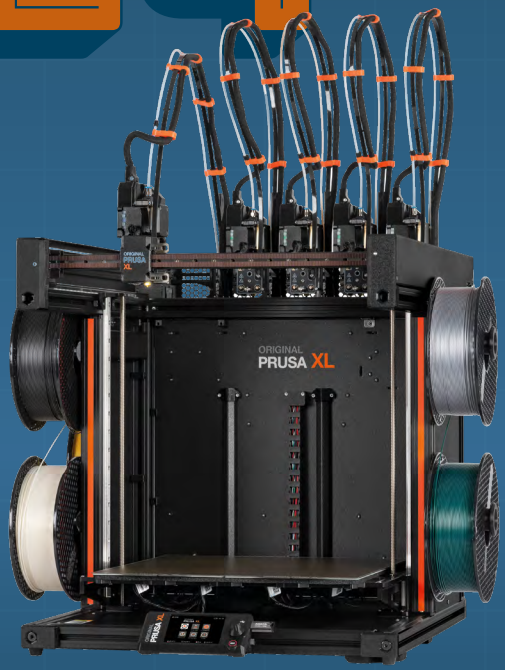
(Far) Left
If you have any idea what you could actually use this for, get in touch – hackspace@raspberrypi.com

Left
The invisibility shield is really impressive in the video. Take a look at the link above to watch it

next month

issue #79 ON SALE 30 MAY

THE BEST 3D PRINTERS 2024



ALSO

- CIRCUITPYTHON
- RASPBERRY PI
- TINY COMPUTERS
- LASER CUTTING

AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

RFID treasure hunter

Some time ago, Robert MacLaren had an idea to create a treasure hunt using RFID (radio-frequency identification) cards and an MP3 player module. Now, thanks to an Arduino Nano, a DFRobot speech synthesis module, and a lovely custom case featuring threaded inserts melted into 3D-printed plastic, it's here.

Using tech to update old games is a great way to get kids accustomed to things that go beep and bleep. Using your own tech, that you've made yourself, is even better – it exposes young minds to the possibilities that a soldering iron can unlock, and most importantly of all, it earns you serious Brownie points.

PiKVM

Remote control **redefined**

Manage your servers or PCs remotely!

POWERED BY Raspberry Pi

PiKVM V4 Mini

Small, cost-effective, and powerful!

- Power consumption in idle mode: just 2.67 Watts!
- Transfer your mouse and keyboard actions
- Access to all configuration settings like UEFI/BIOS
- Capture video signal up to 1920x1200@60 Hz
- Take full control of a remote PC's power

PiKVM V4 Plus

The most feature-rich edition

- More connectivity
- Extra storage via internal USB 3.0
- Upgraded powering options
- More physical security features
- Extra HDMI output
- Advanced cooling solution

A cost-effective solution for data-centers, IT departments or remote machines!

Available at the main Raspberry Pi resellers

core ELECTRONICS · PiShop.ca · RPishop.cz · Welectron. · elektorstore · PIMORONI

Pi-Shop.ch · The PiHut · PiShop.us · CloudFree · KUBii · Cytron Marketplace

ROBU.IN Your ideas, Our Parts · NETTOP COMPUTERS & ELECTRONICS · netsket mall · DIREKTRONIK · RaspberryPi.dk

HiPi.io

No reseller in your country?
Check shop.hipi.io (import fees might apply).

List of official resellers by country: