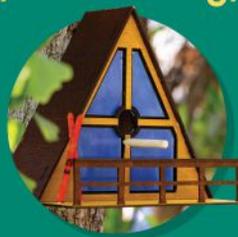


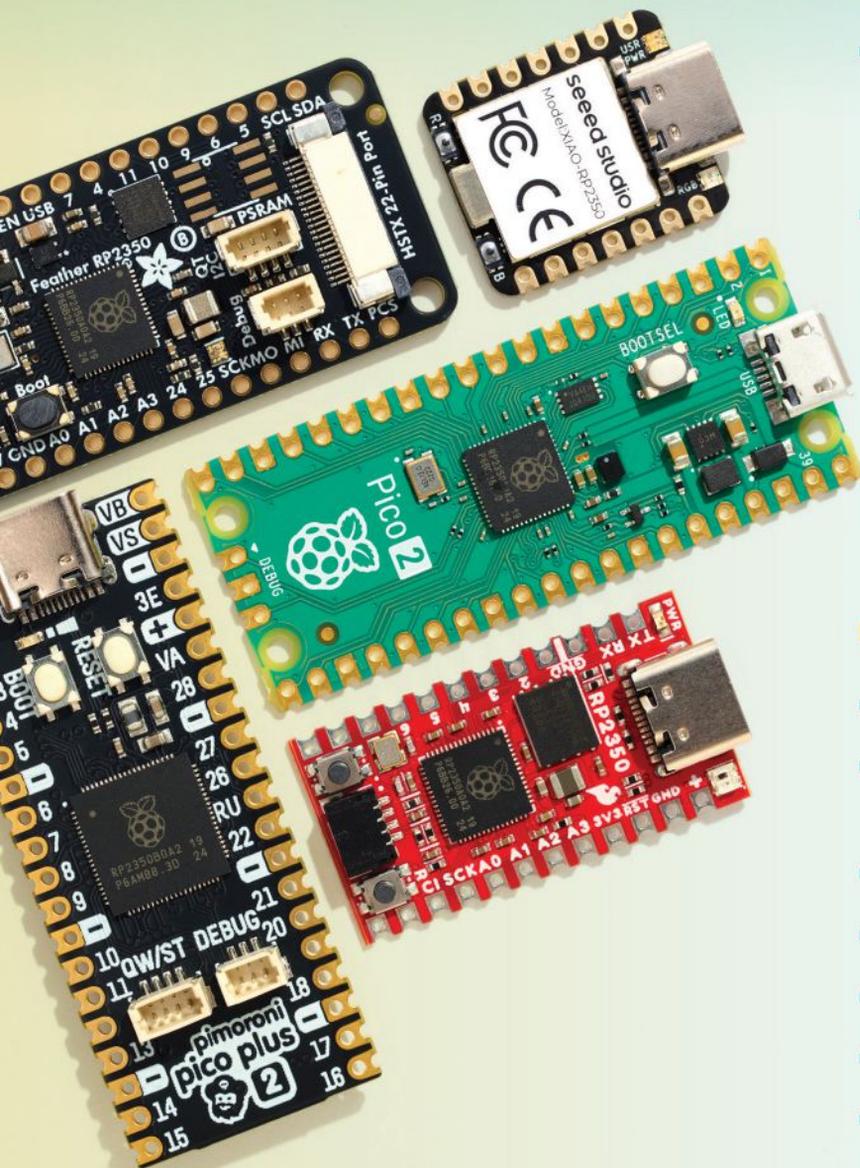
Make:



SNOW BIRDS:
Laser-Cut a
Ski Chalet
Birdhouse

BOARDS GUIDE 2025

CORE STRENGTH!



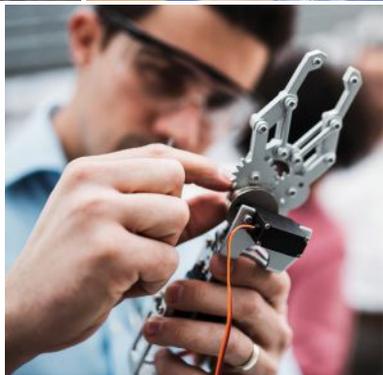
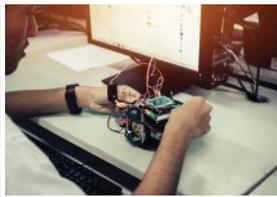
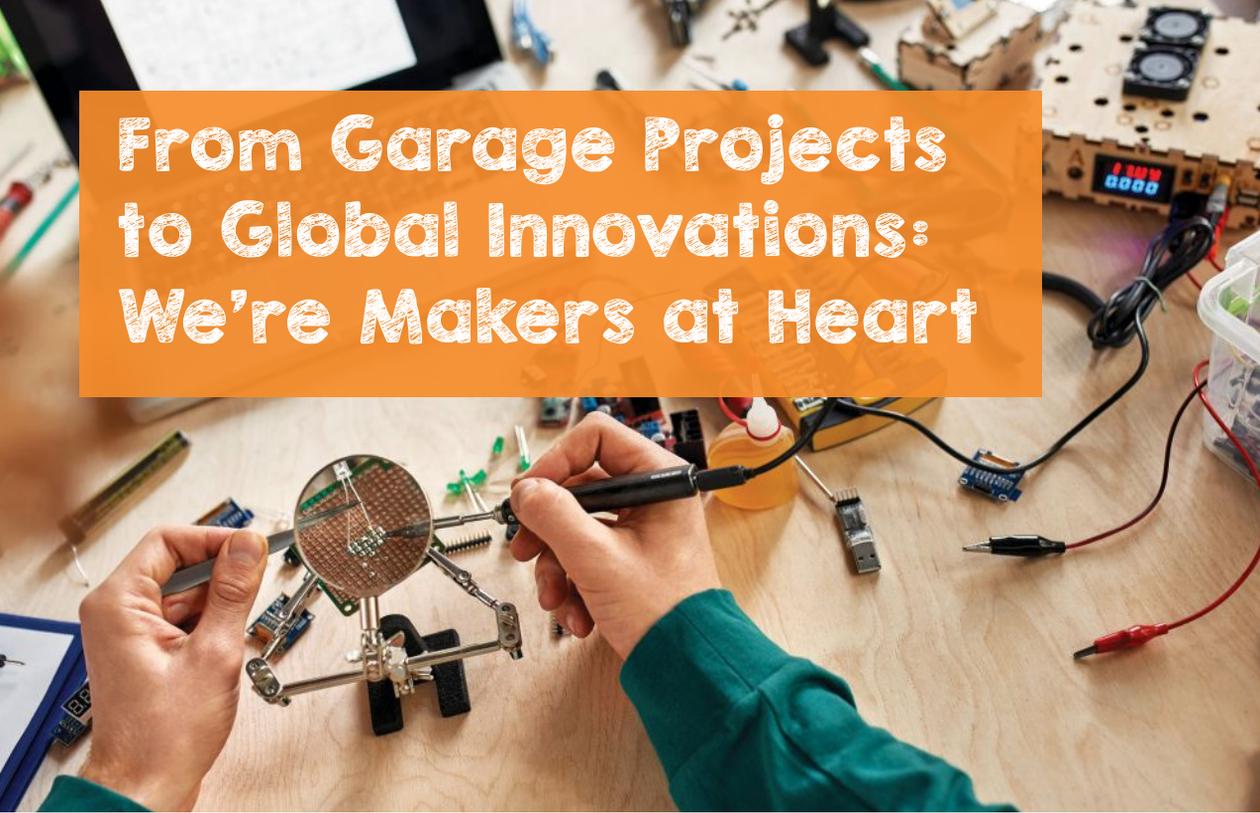
- New **RP2350** Chip Flexes Double Dual Cores
- Hot Boards for AI, LEDs, Wearables, Education, and IoT

SPECIAL INSERT:
77 BURLY NEW BOARDS



- Robot Kits for Kids
- 3D-Printed Sparkly Mirror Fabrics
- AI Companion Bot
- Router Restarter
- Autotuned Kazoo
- Gatorade Battery
- Camera Obscura

From Garage Projects to Global Innovations: We're Makers at Heart



MICROCHIP

microchip.com



The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners. © 2023 Microchip Technology Inc. All rights reserved.

PCBWay

**SALE FOR PCB
PROTOTYPE ORDER**

ONLY \$5

FOR 10 PCBs

WHY CHOOSE PCBWAY?

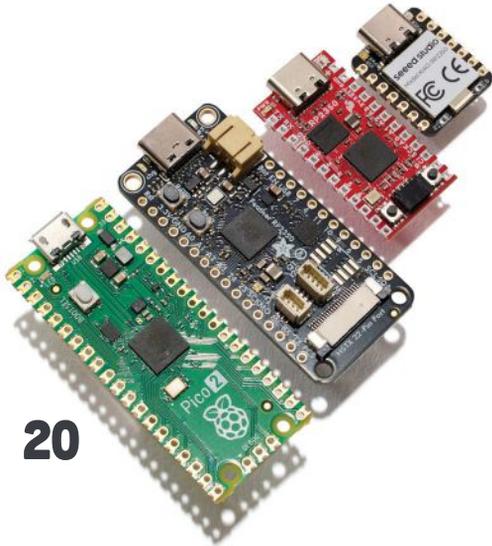
- Top-quality products ■
- Fast turnaround time ■
- Hassle-free ordering ■
- 24/7 customer support ■

**JOIN PCBWAY'S 7th PROJECT DESIGN
CONTEST AND WIN BIG!**

Address:
www.pcbway.com

Email:
service@pcbway.com





20



24



48

COLUMNS

From the Editor's Desk **06**

Tips and tidbits from readers like you.

Welcome: A Community Round **07**

An opportunity to invest in Make: Community.

Made on Earth **08**

Amazing builds from around the globe.

FEATURE

Why Make Electronic Music **12**

Explore the subversive joys of building your own unique and alien instruments.

Maths in the Wild **16**

Science-artist Nick Sayers expresses the beauty in mathematics through hands-on installations.

BOARD TRICKS

My Sweet Board **20**

There's so much to love in this year's bumper crop of new dev boards.

Basic Boards **22**

Go beyond Raspberry Pi and Arduino with these similarly robust, but lesser known, boards.

"Hey, Robot!" **24**

Chat with and command your own embedded-AI companion bot using local LLMs, no internet needed.

How to "Make" Friends **32**

Some of our favorite companion bot creators talk about how their electronic buddies keep evolving.

Micro:bit Wizardry **34**

Use a kid-friendly, low-cost processor to create a magic wand and a "familiar" friend.

Oxocard Connect Innovators Kit

Make: Edition **42**

Our sweet new kit makes learning programming and electronics quick, easy, and fun — with ready-made code and plug-in cartridges.

Badge of Love **48**

Creating a video game wedding invite with Adafruit's nifty badge boards.

SPECIAL INSERT

Make: Guide to Boards 2025

We explore the latest and greatest in the world of MCUs, SBCs, and more, so you can pick the best board for your next project.

DIY ENERGY

Batteries With Bulbul **52**

A bird's-eye view of battery types and applications.

ON THE COVER:

Clockwise from top: Seed Xiao RP2350, Raspberry Pi Pico 2, SparkFun RP2350 Pro Micro, Pimoroni Pico Plus 2, and Adafruit Feather RP2350.

Photos: Mark Madeo and Andy McElfresh

1+2+3: Beverage Battery 58

Use household items and your favorite electrolyte-filled sports drink to make this simple battery.

PROJECTS

3D-Printed Mirror Fabric 62

Embed tiny mirrors and mesh into your 3D prints to create sparkling fabrics.

My Autotune Kazoo 68

Deploy real-time pitch shifting like the pros, in the most amateur instrument.

Ski Chalet Birdhouse 76

A laser cutter makes creative birdhouse designs extremely fun.

Camera Obscura 80

Make a walk-in camera room to project the outside world inside (and upside down).

Squishy Tech: DMX Lighting for Makers 86

Set up professional stage lighting with just an Arduino.

Breathe Breezy 92

This DIY powered respirator keeps you safe and cool.

Router Restarter 96

Cure the scourge of internet and Wi-Fi outages with this daily auto rebooter.

Hack Your Toothbrush 100

Part 2: Salvage the magnets from discarded electric toothbrushes and use them for fun projects!

Toy Inventor's Notebook:

Barrel of Makes! 104

Update a classic toy with a laser-cut version of the *Make*: mascot, Makey the robot.

SKILL BUILDER

How to Detect Spy Tech 108

Teardown: How the budget K18 Bug Detector works — and how to use it.

Cheap Yellow Displays 116

Get started with new Arduino libraries and example projects for these inexpensive ESP32+LCD boards.

TOOLBOX

How to Choose Robot Kits for Kids 120

Making sense of billions of bots.

Toolbox 124

Gear up with the latest tools and kits for makers.

OVER THE TOP

A Kinder Robot 128

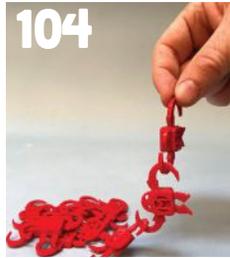
Kind Humanoid's vision for a future of robotic partners is ambitious and inspiring.



62



86



104



108

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION (Required by Act of August 12, 1970; Section 3685, Title 39, United States Code). 1. MAKE Magazine 2. ISSN: 1556-2336 3. Filing date: 10/1/2024. 4. Issue frequency: Quarterly. 5. Number of issues published annually: 4. 6. The annual subscription price is \$4.99. 7. Complete mailing address of known office of publication: Make Community, LLC 150 Todd Road Ste. 100, Santa Rosa, CA 95407. 8. Complete mailing address of headquarters or general business office of publisher: Make Community, LLC 150 Todd Road Ste. 100, Santa Rosa, CA 95407. 9. Full names and complete mailing addresses of publisher, editor, and managing editor: Publisher, Dale Dougherty, Make Community, LLC, 150 Todd Road Ste. 100, Editor, Keith Hammond, Make Community, LLC, 150 Todd Road Ste. 100, Santa Rosa, CA 95407, Managing Editor, N/A, Make Community, LLC, 150 Todd Road Ste. 100, Santa Rosa, CA 95407. 10. Owner: Make Community, LLC; 150 Todd Road Ste. 100, Santa Rosa, CA 95407. 11. Known bondholders, mortgages, and other security holders owning or holding 1 percent of more of total amount of bonds, mortgages or other securities: None. 12. Tax status: Has Not Changed During Preceding 12 Months. 13. Publisher title: MAKE Magazine. 14. Issue date for circulation data below: Fall 2024. 15. The extent and nature of circulation: A. Total number of copies printed (Net press run). Average number of copies each issue during preceding 12 months: 44,990. Actual number of copies of single issue published nearest to filing date: 45,095. B. Paid circulation. 1. Mailed outside-county paid subscriptions. Average number of copies each issue during the preceding 12 months: 30,955. Actual number of copies of single issue published nearest to filing date: 30,629. 2. Mailed in-county paid subscriptions. Average number of copies each issue during the preceding 12 months: 0. Actual number of copies of single issue published nearest to filing date: 0. 3. Sales through dealers and carriers, street vendors and counter sales. Average number of copies each issue during the preceding 12 months: 3,723. Actual number of copies of single issue published nearest to filing date: 3,015. 4. Paid distribution through other classes mailed through the USPS. Average number of copies each issue during the preceding 12 months: 0. Actual number of copies of single issue published nearest to filing date: 0. C. Total paid distribution. Average number of copies each issue during preceding 12 months: 34,678. Actual number of copies of single issue published nearest to filing date: 33,644. D. Free or nominal rate distribution (by mail and outside mail). 1. Free or nominal Outside-County. Average number of copies each issue during the preceding 12 months: 729. Number of copies of single issue published nearest to filing date: 744. 2. Free or nominal rate in-county copies. Average number of copies each issue during the preceding 12 months: 0. Number of copies of single issue published nearest to filing date: 0. 3. Free or nominal rate copies mailed at other Classes through the USPS. Average number of copies each issue during preceding 12 months: 0. Number of copies of single issue published nearest to filing date: 0. 4. Free or nominal rate distribution outside the mail. Average number of copies each issue during preceding 12 months: 851. Number of copies of single issue published nearest to filing date: 577. E. Total free or nominal rate distribution. Average number of copies each issue during preceding 12 months: 1,580. Actual number of copies of single issue published nearest to filing date: 1,351. F. Total free distribution (sum of 15c and 15e). Average number of copies each issue during preceding 12 months: 36,258. Actual number of copies of single issue published nearest to filing date: 34,995. G. Copies not Distributed. Average number of copies each issue during preceding 12 months: 8,732. Actual number of copies of single issue published nearest to filing date: 10,100. H. Total (sum of 15f and 15g). Average number of copies of single issue published nearest to filing date: 44,990. Actual number of copies of single issue published nearest to filing date: 45,095. I. Percent paid. Average percent of copies paid for the preceding 12 months: 95.64%. Actual percent of copies paid for the preceding 12 months: 96.14%. 16. Electronic Copy Circulation: A. Paid Electronic Copies. Average number of copies each issue during preceding 12 months: 4,045. Actual number of copies of single issue published nearest to filing date: 4,245. B. Total Paid Print Copies (Line 15c) + Paid Electronic Copies (Line 16a). Average number of copies each issue during preceding 12 months: 38,723. Actual number of copies of single issue published nearest to filing date: 37,889. C. Total Print Distribution (Line 15f) + Paid Electronic Copies (Line 16a). Average number of copies each issue during preceding 12 months: 40,303. Actual number of copies of single issue published nearest to filing date: 39,240. D. Percent Paid (Both Print & Electronic Copies) [(16b divided by 16c x 100)]. Average percent of copies paid during preceding 12 months: 96.08%. Actual percentage of copies paid for single issue published nearest to filing date: 96.56%. I certify that 50% of all distributed copies (electronic and print) are paid above nominal price: Yes. Report circulation on PS Form 3526-X worksheet 17. Publication of statement of ownership will be printed in the Winter 2024 issue of the publication. 18. Signature and title of editor, publisher, business manager, or owner: Todd Sotkiewicz - Business Manager. I certify that all information furnished on this form is true and complete. I understand that anyone who furnishes false or misleading information on this form or who omits material or information requested on the form may be subject to criminal sanction and civil actions.

"When adults see me with a companion bot they might think it's weird, but kids get so excited: 'OMG, it's a robot!' They know where it's at."
—Jorvon "Odd Jay" Moss

PRESIDENT

Dale Dougherty
dale@make.co

VP, PARTNERSHIPS

Todd Sotkiewicz
todd@make.co

EDITORIAL

EDITOR-IN-CHIEF
Keith Hammond
keith@make.co

COMMUNITY EDITOR
David J. Groom
david@make.co

PRODUCTION MANAGER
Craig Couden

ONLINE EDITOR
Sam Freeman

CONTRIBUTING EDITORS
Tim Deagan
William Gurstelle

CONTRIBUTING WRITERS
Debra Ansell, Cabe Atwell,
Daniel Bachfeld, Larry Bank,
Danielle Boyer, Rich Cameron,
Chipperdoodles, Christina Cole,
Katherine Connell, Ted Donnell,
Guy Dupont, Alex Glow,
Kate Hartman, Joan Horvath,
Shawn Hymel, Victoria Jaqua,
Bob Knetzger, Andy McElfresh,
Krizia Medero, Kirk Pearson,
Marshall Piroos, Mo Rahman,
Nick Sayers, Wayne Seltzer,
Angela Sheehan, Matt Stultz,
Cy Tymony, Theron Wierenga,
Lee Wilkins

CONTRIBUTING ARTISTS
Mark Madeo

MAKE.CO

ENGINEERING MANAGER
Alicia Williams

WEB APPLICATION
DEVELOPER
Rio Roth-Barreiro

DESIGN

CREATIVE DIRECTOR
Juliann Brown

BOOKS

BOOKS EDITOR
Kevin Toyama
books@make.co

GLOBAL MAKER FAIRE

DIRECTOR, GLOBAL
MAKER FAIRE
Katie D. Kunde
GLOBAL LICENSING
Jennifer Blakelee

MARKETING

DIRECTOR OF
MARKETING
Gillian Mutti
PROGRAM COORDINATOR
Jamie Agius

OPERATIONS

ADMINISTRATIVE
MANAGER
Cathy Shanahan

ACCOUNTING MANAGER
Kelly Marshall

OPERATIONS MANAGER
& MAKER SHED
Rob Bullington

LOGISTICS
COORDINATOR
Phil Muelrath

PUBLISHED BY

MAKE COMMUNITY, LLC
Dale Dougherty

Copyright © 2024
Make Community, LLC. All rights
reserved. Reproduction without
permission is prohibited.
Printed in the U.S. by Schumann
Printers, Inc.

Comments may be sent to:
editor@makezine.com

Visit us online:
make.co

Follow us:
X @make
f makemagazine
makemagazine
makemagazine
makemagazine

Manage your account online,
including change of address:
makezine.com/account
For telephone service call
847-559-7395 between the
hours of 8am and 4:30pm CST.
Fax: 847-564-9453.
Email: make@omeda.com

Make:
Community

Support for the publication
of *Make:* magazine is made
possible in part by the
members of Make: Community.
Join us at make.co.

CONTRIBUTORS

**What fictional robot
would you most like to
be friends with?**



Krizia Medero
Minneapolis, Minnesota
(3D-Printed Mirror Fabric)
Baymax from *Big Hero 6*:
kind-hearted, helpful, and
as soft as a marshmallow.



Kate Hartman
Toronto, Ontario, Canada
(Guide to Boards: Wearables)
Marvin the Paranoid
Android from *The
Hitchhiker's Guide to
the Galaxy*. His deadpan
doom-and-gloom persona
makes him so relatable!



Shawn Hymel
Lafayette, Colorado
(“Hey Robot!” Local LLMs)
Johnny 5 for his curiosity,
humor, and courageous
pursuit of empathy and
independence.

Issue No. 91, Winter 2025. *Make:* (ISSN 1556-2336) is published quarterly by Make Community, LLC, in the months of February, May, Aug, and Nov. Make: Community is located at 150 Todd Road, Suite 100, Santa Rosa, CA 95407. SUBSCRIPTIONS: Send all subscription requests to Make:, P.O. Box 566, Lincolnshire, IL 60069 or subscribe online at makezine.com/subscribe. Subscriptions are available for \$34.99 for 1 year (4 issues) in the United States; in Canada: \$43.99 USD; all other countries: \$49.99 USD. Periodicals Postage Paid at San Francisco, CA, and at additional mailing offices. POSTMASTER: Send address changes to Make:, P.O. Box 566, Lincolnshire, IL 60069. Canada Post Publications Mail Agreement Number 41129568.

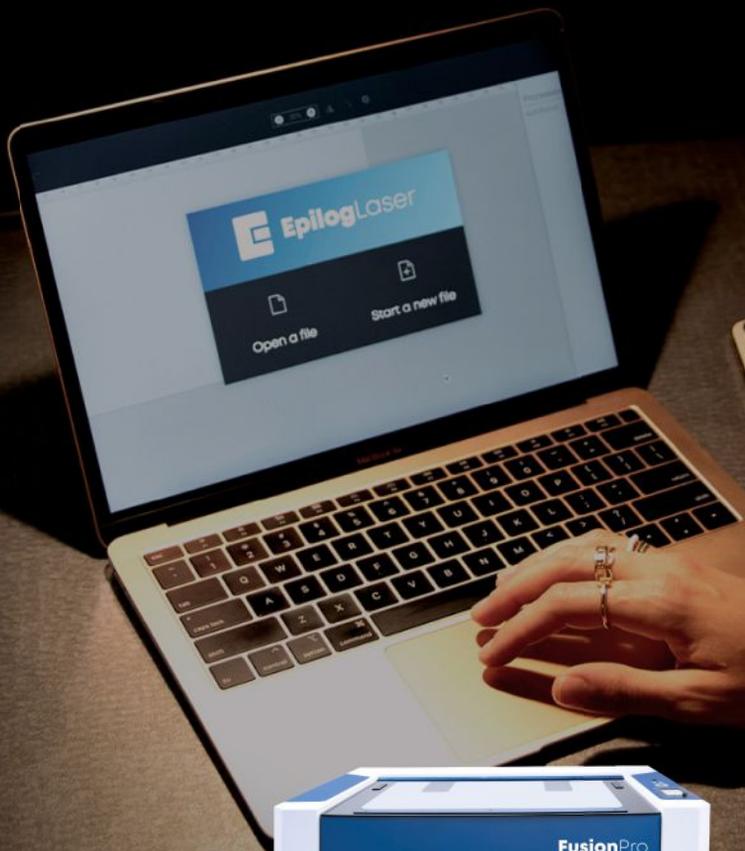
PLEASE NOTE: Technology, the laws, and limitations imposed by manufacturers and content owners are constantly changing. Thus, some of the projects described may not work, may be inconsistent with current laws or user agreements, or may damage or adversely affect some equipment. Your safety is your own responsibility, including proper use of equipment and safety gear, and determining whether you have adequate skill and experience. Power tools, electricity, and other resources used for these projects are dangerous, unless used properly and with adequate precautions, including safety gear. Some illustrative photos do not depict safety precautions or equipment, in order to show the project steps more clearly. These projects are not intended for use by children. Use of the instructions and suggestions in *Make:* is at your own risk. Make Community, LLC, disclaims all responsibility for any resulting damage, injury, or expense. It is your responsibility to make sure that your activities comply with applicable laws, including copyright.



Introducing **Epilog**Pulse



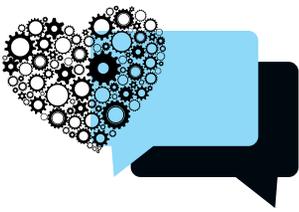
Control your laser
from any
device –
including
Macs
and
Tablets!



 **Epilog**Laser

888.437.4564 | SALES@EPILOGLASER.COM
EPILOGLASER.COM/MAKE





FROM THE EDITOR'S DESK

SURPRISE MAKER BADGE

I purchased books from Maker Shed and they arrived with this nifty “Here There Be Makers” badge, which needed some LEDs and a battery installed. It took me three whole goes, but I got there (youtu.be/sgs-xxL1-8Q). Thanks again, that was a fun surprise to find in the package. —*John Elliot V, Sydney, Australia*

CIRCUIT DIAGRAMS SORELY MISSED

I found your magazine a couple of years ago and currently subscribe. There are several small electronics projects I'd be interested in doing except there is no circuit schematic diagram to show how all the elements are connected. Many people would breadboard the project first, modify and test, and only solder everything together when they are finally happy with it. There is no easier way to do this than the basic circuit diagram.

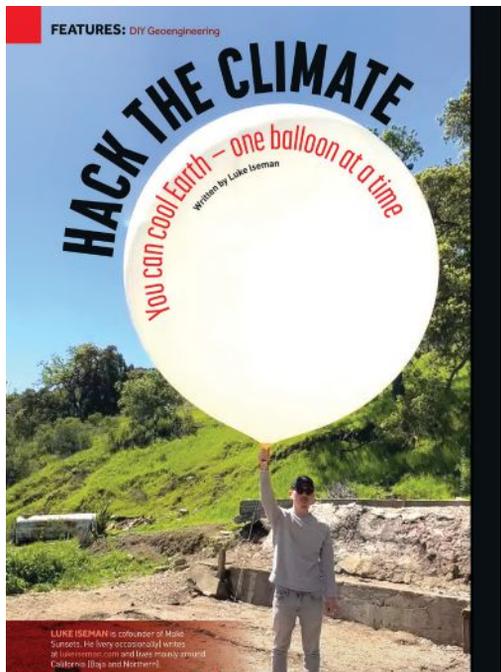
—*Kalen Martens, Vancouver, British Columbia*

Editor Keith Hammond replies: *Thanks Kalen, we love schematics, too! Often an author provides a physical wiring diagram or breadboard layout instead, but we strive to include schematics whenever available. You'll find some nice ones in this issue.*

DANGERS OF GEOENGINEERING

I believe what Luke Iseman and his company Make Sunsets are doing to be against the end goal of humanity reversing (and outlasting) our effect on the climate. I can't stop them from lifting balloons but I can write to you about choosing to amplify and encourage this dangerous practice (“Hack the Climate,” *Make*: Volume 90). Piecemeal sulfur balloons would only help hydrocarbon oligarchs to drag humanity further in the hole we've dug ourselves into. —*Conrad Kirby, via email*

Editor Keith Hammond replies: *Thanks for your thoughtful letter, Conrad. Iseman admits his small-scale launches of sulfur dioxide to the stratosphere won't yet move the needle, and I suspect the controversy is the point — to prod us all to hurry and figure out whether this could work to deflect solar radiation, and what the rules should be. We see the potential downsides — ozone impacts, prolonged fossil fuel dependence — but we encourage his boldness in instigating an urgent conversation. 🗳️*



Help Us Celebrate 20 YEARS!

Make: is turning 20 years old (!) next issue and we want to hear from you! How has *Make:* been a part of your life? What are your favorite projects from *Make:*? Email us at editor@makezine.com, subject line “Happy Birthday *Make:*” and we may print your response in an upcoming issue.

WELCOME

A Community Round

Announcing an opportunity to invest in *Make: Community*

by Dale Dougherty, President of *Make: Community*

I once explained to a friend that I was most proud that *Make:* magazine and Maker Faire had created a community of makers — how makers found each other through the magazine and events and how an even larger public could now find them. Giving makers opportunities to share their work in public became the mission of the magazine and Maker Faire but it also proved the best way to grow the whole community.

Next year in February, we will celebrate the 20th anniversary of the first issue of *Make:* magazine. I could not have imagined the impact that *Make:* combined with Maker Faire would have on so many people — some of you have been subscribers since the first issue! — but I am grateful for it.

I am especially happy when I hear from young engineers and creative artists that *Make:* magazine and Maker Faire were an important influence on their lives. I'd like to think that makers demonstrated that it was possible to do this kind of work, and then showed them how to do it themselves. That's how we all created more makers. The real value of the term *maker* is not in how it is defined — I've said that you should define it any way that includes yourself — but in how it connects you to others, in particular, helping you find other makers.

This community doesn't belong to anyone, nor is any one person or company at the center of it. Perhaps that's one of the reasons that the loose-knit maker community has a chance to endure. Twenty years of work reflects the efforts of one generation, and I am now thinking about how to pass this work on to the next generation.

That's all a way of explaining a new initiative, a community funding round for *Make:* and Maker Faire. Today I am the sole owner of *Make:* and Maker Faire; I have supported the business when it needed help. Currently, I have no interest in selling the business to another entity, who may or may not understand the value of what we do. The people who understand it the best and value it



the most are members of the maker community. What I would like to see happen is that over time, the community itself, the community that has grown out of *Make:* and Maker Faire, will come to own more and more of this organization.

As part of this initiative, we have just launched a community investment round giving you the opportunity to own a stake in *Make:* and Maker Faire. For as little as \$100, you can become a part-owner of *Make:* and participate in our future growth. It is an investment in the next generation of the maker movement.

Your investment will help us:

1. Continue producing *Make:* magazine in print each quarter
2. Bring Maker Faire to more locations nationally and worldwide
3. Develop new online resources and tools for our community
4. Foster grassroots innovation and ingenuity in our local communities
5. Promote the power of play, hands-on learning, and the practice of problem-solving in education.

Thank you for being an integral part of what we do and part of the larger maker community. Working together, the organization behind *Make:* and Maker Faire can grow in partnership with its community and help ensure that we inspire generations of makers to come. 🍷



If you'd like to learn more about investing in *Make:*, please visit: wefunder.com/make



MADE ON EARTH

Amazing builds from around the globe

Know a project that would be perfect for Made on Earth?

Let us know: editor@makezine.com

THE NATURE OF FOUND OBJECTS

AIRTIGHTARTWORK.COM

All his life, **Matt Wilson** has known he wanted to be an artist, a maker whose work reflects the natural world we live in but is unlike anything seen before. His stunning metalwork pieces, with their intricate details and polished sheen, fit these lofty criteria. On the surface, Wilson's sculptures appear molded out of a single piece of metal. Taking a closer look, though, you'll glimpse familiar materials — silverware, wires, springs, sewing machine wheels, and other reclaimed and recycled pieces of scrap metal. This effect is achieved by design, as Wilson creates his sculptures to inspire others to appreciate the simple and overlooked items that serve crucial roles in our modern society.

Born and raised in Greenville, South Carolina, Wilson works as an independent full-time artist in Charleston, creating and selling metal sculptures as Airtight Artwork. His initial design process is mainly internalized, as his artistic experience allows him to shape his vision inside his head without a sketch or 3D model. The recycled materials he repurposes come from many secondhand sources, which lend themselves well to different sculpting styles. "I keep a pretty good inventory so I can pick and choose styles/designs of utensils based on the look I'm going for," he says. "Sometimes it just pops in my head really easily. Other times it's a fun brain exercise to see what potential the material could have in a future sculpture."

Before putting everything together with MIG and TIG welders, Wilson uses basic tools to manipulate his metals — pliers, vises, band saws, angle grinders, and rawhide hammers. Because he finds visible welds distracting and ugly, he has developed a technique over the years to hide his welds to create a cohesive and seamless design.

For those interested in metalworking, Wilson recommends starting small. "Diving into metal sculpture art is a big commitment for the amount of tools you will need. I often tell people that want to do something similar, to start by learning to draw. I went to school for drawing and painting. I've drawn every day of my life. It's a huge part of being able to know when things look right." —*Marshall Piros*





REAL RETRACTABLE LIGHTSABER HEROTECH.ENGINEERING

In *Star Wars* lore, the lightsaber is more than just a weapon; building one is a symbol of a Jedi's connection to the Force. To that end, **Jacob Uy, Madison Tong, and Aaron Codrington** of Needham, Massachusetts — better known as HeroTech on YouTube — set out to create the most iconic weapon in the *Star Wars* universe and managed to successfully achieve that goal using some out-of-the-box thinking and some off-the-shelf hardware. “Our main challenges were clear,” Jacob explained in the project video. “We needed a fully contained blade extension and retraction; a compact, to-scale hilt design; [and] a realistic blade and sound.”

The DIY lightsaber can extend and retract its blade at the push of a button, just like in the movies, and features a brilliant LED blade complete with authentic sounds. The saber was designed using high-powered chip-on-board (COB) LED strips, throwing off enough light to see the blade in sunlight. The LEDs are affixed to a springy aluminum magician's cane that can extend to just over 3 feet long and retract small enough to fit inside the lightsaber's hilt. The LED strips loop around the tip of the cane, which

provides a solid look from all angles.

The hilt is designed to scale, mimicking the dimensions of Anakin Skywalker's lightsaber, which is around 2 inches in diameter. A circular spool design was designed to store the LED strips inside the hilt when retracted, along with rotating bearings at key contact points to keep them from fouling. A metal geared motor with a 1:90 gear ratio controls the extension and retraction of the blade, while a Xiao RP2040 MCU, paired with an Adafruit motor driver, handles the motor control.

The engineers also equipped the lightsaber with a Proffieboard V3.9, a special sound board found in other custom lightsaber designs, to handle blade ignition, retraction, and additional lightsaber sound effects, which can also be customized to the user's preferences. The entire lightsaber is powered by a four-cell 14.8V LiPo battery that's also housed within the hilt, which is 3D printed using ABS material.

The result of HeroTech's lightsaber design is an impressive accomplishment carried out by a team of dedicated engineers and *Star Wars* fans, which not only looks authentic but operates just like those in the movies. —*Cabe Atwell*



FLY BY WIRE

LAURAMATHEWSART.COM

Laura Mathews has always been fascinated with the biological mechanics of flight. Ever since her family rescued and raised a sparrow chick, the memories of watching the tiny pink chick slowly grow into a being of wings and winds have fueled her creative enterprises. Unfortunately, humans don't have wings and can't experience the magic of flight on their own, so Mathews now replicates this effect through the next-best avenue — laser-cut wooden puppets and marionettes.

Mathews' wooden works usually begin life as rough pencil sketches, which are then converted into vectors in CAD software and superimposed over reference images to ensure anatomical and articulation accuracy. In her workshop in Nottingham, UK, she continues a cycle of digital refinement and physical assessment until she's happy with how the puppet looks, moves, and feels, which can take anywhere from months to years. "Sometimes just one problem will take weeks to solve. I'll make multiple mistakes, and it can get very frustrating," Mathews says. "But, I am incredibly persistent. Finding the answer is always so rewarding and it gives me the confidence to persevere the next time I'm stuck."

Many of her marionettes' internal mechanisms are based on folding umbrella linkages, which operate similarly to bones and tendons. Because of this, Mathews is able to make big movements from small string contractions and accurately plot a puppet's motion, which greatly enhances its overall mystique and realism. "I love the contrast between the clean, perfect, digital environment, and the messy chaos of my physical workshop," Mathews admits. "It's a symbiotic process of something dreamlike becoming real."

Mathews says she's often asked whether she plans to try automating her creations with motors and servos, a perceived "next step" in her craft's evolution — but in her eyes, that defeats the point. "Servos are noisy, whereas a puppet is nearly silent. Your muscles control the strings and with practice, flying becomes natural — almost as though you're inhabiting another body." You can see more of Mathews' marionettes on her Instagram [@LauraMathewsArt](https://www.instagram.com/LauraMathewsArt). —*Marshall Piro*s

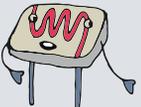




Written by Kirk Pearson • Illustrations by Maisy Byerly

WHY MAKE ELECTRONIC MUSIC?

THE SUBVERSIVE JOYS OF BUILDING
YOUR OWN INSTRUMENTS



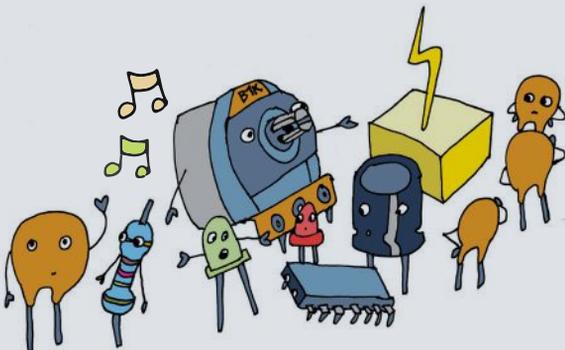
As a kid, I became fascinated with electronic music when I noticed how much it pissed off my parents.

My childhood (in the early aughts) was a weird time for popular music. The radio was dominated by aggressive bro-metal and boy bands that sounded like the sonic equivalent of processed cheese. Pop culture was as bland as it gets, but once in a blue moon the radio would play an instrument so colorful and alien that my child-brain would struggle to imagine the thing that produced it. Again and again, the answer would be accompanied by an arm wave and the word *synthesizer*.

You see, until incredibly recently, synthesizers were seen as incredibly uncool. Electronic music was simultaneously associated with MDMA-fueled ragers and pretentious computer geeks. Movies of that era portrayed electronic musicians as dorky, emotionless, wannabe performers that wouldn't know good music if it slapped them in the face. Meanwhile, every portrayal of a guitar player was suave, technically dexterous, and dripping with sex appeal I can assure you they didn't have.

But my child brain hadn't yet accepted any of those connotations. To me a synthesizer was simply a synthesizer, and it seemed to offer a virtually unlimited color palette of sounds to work with — far more sounds than my three-piece middle school punk band could make. I never saw electronic instruments as a means to replace anything acoustic — rather, they allowed you to play with physics in a way traditional instruments did not.

All instruments make sound by vibrating matter. An acoustic instrument, like a violin, jiggles around air molecules that push other air



The Magnet Carta: Dogbotic's six-channel tape looping ambient machine.



The band Vansire, inspecting the Magnet Carta.



Vansire, in concert with musical instruments by Dogbotic.

YOU SHOULDN'T HAVE TO LIMIT YOUR MUSICAL PALETTE TO THE STOCK PATCHES THAT SOME CORPORATION PRE-PACKAGES YOUR KEYBOARD WITH.

molecules and eventually transmit the sound to your ear. Electronic instruments, however, jiggle around electrons. Only at the last step do those electrons push and pull a speaker, which creates the pressure wave in the air. It's far easier to control how an electron moves than an air molecule. This means in order to change the sound itself, all we have to do is slightly change the pattern the electrons wiggle in. By abstracting a wave of air pressure into a wave of changing voltage, we can sculpt with audio. The process is nothing short of fascinating, and somewhat profound. When you're playing with electric charges, it's not hard to make a sound that nobody has ever heard before (oftentimes, by accident).

The klutzy, awkward character of electronic instruments never bothered me, and I became fascinated by them — they were instruments that seemed to have some complex ideology to them that truly angered the right person. Much like how early rock and roll in the 1950s seemed to scare an older conservative generation, there was something about a synth that seemed imbued with its own political ideology. Why did they seem to make people so upset?

Synthesizers, drum machines, FX pedals, and all electronic instruments are deeply symbolic. When you play one on stage, your performance — on some unwritten level — is in dialogue

with technology, demonstrating a way that humans and machines can co-exist. Electronic instruments challenge people's assumptions about not only what types of music are considered acceptable, but also what sounds can be considered "musical" in the first place. It's a field that continually challenges what is considered appropriate, and as a result is shockingly accepting of new ideas and people. Music made with a machine poetically blurs the line between the performer and instrument, allowing one to be an extension of the other. All electronic music inherently makes these statements, and the context around them is always changing. Our social interpretations of technology change quickly, which allows electronic music to take on new meanings as history creeps onward.

I've been building synthesizers for the better part of 10 years, and as much as I love waxing poetic on the political nature of making an instrument, it's also just plain fun. Making a musical circuit isn't hard. In fact, you'd be astonished how quickly a total beginner can do it! Like strumming your first guitar chord, building a simple oscillator is an oddly gratifying moment of discovery and connection — a sound that did not exist before now exists. Synthesizers are quirky, temperamental little monsters that live on current the way you live on food. They breathe charge imbalances as you breathe air.

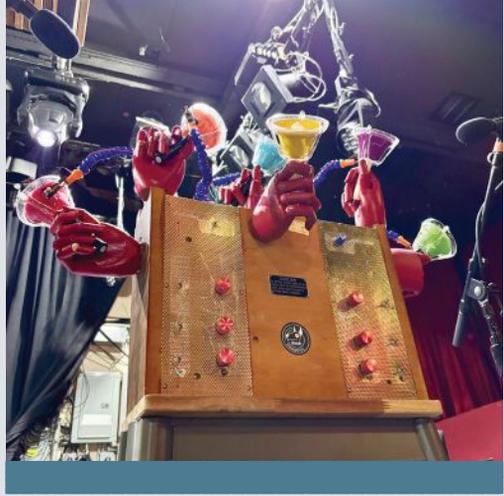
Congratulations, you mad scientist, you.

Building your own circuitry is surprisingly empowering for a lot of people, even if they aren't a "musician" (whatever that means). Learning basic electronics is a fast way to teach you that nothing in the built world is magic — your computer, microwave, and smoke detector all work with the same basic principles, and they all can be understood. Comprehending the built world around us helps us think more clearly about the ways technology can help and harm us. At the very least, building a synth is a friendly reminder that businesses aren't the arbiters of new sounds. You shouldn't have to limit your musical palette to the stock patches that some corporation pre-packages your keyboard with.

Companies have spent a long time trying to convince people that synths are "products" produced by wealthy teams of people and sold to musicians. The truth is that synthesizers are folk instruments. For decades synths were created by folks like you and me, with little technical training, simply because people found the sounds so interesting. Breakthroughs in electronic music are not always technical — most of them are simply creative new angles to look at things that have already existed. While the synthesizer is hardly a new invention, it remains every bit as political a tool as ever.

Don't let your parents tell you otherwise — electronic music is important to pay attention to! And as your world gets bogged down with technology you didn't ask for — say, AI-generated propaganda, personalized advertisements, or commercial surveillance — the electronic arts can help keep society's opinions about these technologies in check. Technology, industry, and communication are all intermingled in a complex dance that neither asks for your consent nor cares about it at all. But there is a silver lining — you have the ability to use that very same technology to reflexively comment on it. Electronic music is a living art with a track record of pushing against the status quo, and critiquing the world that birthed it. It's not a discipline, it's a party! And you're invited!

Whenever a new technology redefines what the role or job of a human is, electronic music will be right there, finding a new usage for it. To

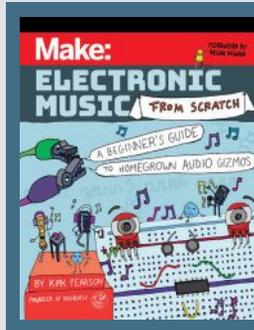


The Ringy-Dingy-Thingy, a electromechanical bell machine.



Kirk Pearson leads the *Make:* crew through a build of a VHS Trautonium synth circuit.

paraphrase Kurt Vonnegut, electronic musicians are the super-sensitive. They're the canaries in the coal mine of the industrial world, recognizing changes in the human-technology relationship before the more robust types recognize anything has gone wrong. 🗨️



KIRK PEARSON is a composer, director of the creative audio laboratory Dogbotic, and co-author of the new book *Make: Electronic Music From Scratch*, available at makershed.com and other booksellers.



Written by Jennifer Blakeslee

Nick Sayers with his *Aral Sea Space-Filling Curve*, Samarkand, Uzbekistan, April 2024.

MATHS in the WILD

Science-artist Nick Sayers expresses the beauty in mathematics



JENNIFER BLAKESLEE runs the Global Maker Faire program and lives in Oakland, California. She has worked in events, radio, writing, education, big data, and public policy.

The Hilbert-Moore curve is a continuous line that forms a space-filling fractal figure. With each step in its development, the square loop becomes increasingly kinked until, in the infinite limit, its one-dimensional line covers every point in a 2D plane. These fractal patterns can map to gridded city streets, and are used to study the folding of proteins. They're also useful for making the world's most convoluted sand castle.

Earlier this year, artist Nick Sayers created an installation inspired by this mathematical figure in Samarkand, Uzbekistan, composed of 256 small sand castles molded with a plastic beach toy from Aral Sea sand — sand so contaminated that he and fellow artists wore rubber gloves while working with it. Sayers installed this crumbling fractal city to reflect the glorious geometric mosaic walls of Samarkand's mosques, and to highlight the environmental tragedy of what was once the world's fourth largest inland sea; now, a toxic, dusty folly of human industry gone awry. The project is one of many in which Sayers — who has lived in Brighton, U.K., since going to art school there in 1992 — explores the multifaceted beauty of “maths,” as the British say.

The Art of Math

To the uninitiated, art and math may seem to exist in wholly different worlds. Certainly,

they are often taught that way. But Sayers, who excelled at both in school, figured out early that when tuned to the same frequency, math and art can inform one another to beautiful ends.

Using found objects and an ethos of reuse, Sayers' continuous, looping explorations of art, math, and environment test the confines and possibilities of the world. His work first came to our attention at Maker Faire U.K. in Newcastle-Upon-Tyne and at Brighton Mini Maker Faire, where over the years he shared iterations of his *Bicycle Spirograph* drawing machines alongside other art. It was hard to miss, with crowds of all ages clamoring for a chance at creative play. Familiar drawing toys such as the Spirograph and Sketch-a-Graph have been around since the mid-20th century, and similar tools for drawing patterns and scaling outlines go back centuries. However, the enlarged scale of Sayers' toy-like drawing machines and his reuse of bikes and other familiar materials for this novel function make them irresistible.

Sayers' first *Bicycle Spirograph* was born from a passing comment on the *Giant Pantograph* he'd created to miniaturize human body outlines onto small sheets of paper. When someone mistook the machine for a “giant spirograph,” Sayers got curious. Long a cycling enthusiast (and founder of the Brighton Naked Bike Ride), he

All photos © Nick Sayers



Aral Sea Space-Filling Curse sand castle installation at Art Station gallery, Samarkand, Uzbekistan.

Crowds of all ages jostle to play with Sayers' *Bicycle Spirograph* drawing machines at Maker Faires and other festivals.

turned to bicycles as a mechanism for drawing hypotrochoid patterns like the Spirograph. “I wanted to see if the difference in speed between the pedals and back wheel of a bike might produce complex shapes,” Sayers recalls.

Dismembering scrap from a local bike shop for parts, he strove to preserve the look and feel of a bicycle, even maintaining the saddle, pedal, and bell. By replacing the back wheel with a wooden drawing arm and using a pen or (recycled) plastic bottle to hold sand or paint, he could control the machine’s output. Sand created temporary patterns, while chalk powder mixed with water left beautiful flower-like marks on roads. Both were great for sharing with students the mathematical, engineering, and physics principles of the machines and the patterns they created.

By changing gears, he can switch between patterns, Sayers explains. “Shifting between the bike’s 18 gears (three front chain rings multiplied by six rear sprockets) produces 18 patterns. Two of the cog teeth ratios — 48:24 and 28:14 — simplify to produce basic 2:1 oval shapes.” When commissioned to bring his work to Egypt and South Korea, Sayers built smaller *Bicycle Spirograph* machines from children’s bikes that could be taken apart and carried in airline luggage.

Made by Hand

Alongside his focus on the environment and sustainable materials, Sayers is a champion of DIY craftivism, preferring to hand-tool even the less obvious parts of his sculptures. With the same simplicity that informs his elegant creations, he points out that unless you are mass-producing something there is often no need for expensive fabrication equipment. Satisfaction and tactile understanding come from building things by hand. “All of the sculptures, machines, and installations I’ve made have been an education for me in construction techniques, mathematics, and engineering,” he says. “They have given me a better understanding of materials’ intrinsic construction qualities, geometry, mathematics, and mechanics, and have improved my engineering skills.”

Along these lines, Sayers has built spherical sculptures from found mass-produced detritus including train tickets, measuring tape, and coffee stirrers; Platonic-solid polyhedral windssocks made from garbage bags; and a 7-meter-tall *New Year Tree* in Baku, Azerbaijan, from over 2,000 PET bottles zip-tied together. A series of livable geodesic domes he made from real estate “For Sale” signs would make Buckminster Fuller proud, but were designed to provoke a conversation about homelessness, affordable



Spheres from recycled materials: measuring tape, realtor signs (*To Live, Show Home*), train tickets, plastic bottles, art paper, coffee stirrers, recycled plastic (*Imagine All The People*).

housing, and sustainable architecture. Beyond just repurposing everyday consumer trash, Sayers' artworks speak to the excess of materials we discard and ask what could be done better with them, or without them at all.

Optical Explorations

Accompanying this interest in shape and scale are Sayers' many experiments with perspective and the geometry of optics. At galleries and festivals, and in his bedroom, a sauna, and a sleeper cabin on a train across Uzbekistan, Sayers has been creating room-sized camera obscuras using a lens to project the outside scene inside. (He'll show you how to make one on page 80 of this issue.)

Sayers has also experimented with solargraph pinhole cameras made from aluminum cans that slowly "sunburn" an image onto photo paper during a 6-month-exposure. These cameras play with time as yet another dimension, and he teaches the technique — which he learned from photographer Justin Quinnell at Maker Faire U.K. in 2011 — in his own workshops.

In another science-art activity, he takes advantage of a phone camera's video frame rate to create the illusion of animation on a spinning record player — a digital version of the 19th-century phenakistoscope.

STEAM World Tour

Sayers has taken his science-art activities to schools, festivals, and events around the world, including Seoul Sculpture Festa, Abu Dhabi Science Festival, London Science Museum, National Museum of Mathematics (New York), The Story of Space (Goa), and Bridges math-art conferences across Europe. At events like these Sayers shares his science-art concepts and craft skills with others, making STEAM subjects fun and tactile and revealing the beauty of the art-math continuum. Renewing his relationship with Maker Faire in 2024, Sayers gave a live video talk about his creative response to the Aral Sea disaster at the first Maker Faire Tashkent (Uzbekistan), and charmed audiences with his drawing machines at Maker Faire Lille (France). 🍷

You can see more of Sayers' art at behance.net/nicksayers and on social media @NickSayers.



Record Player Animations exploit a phone camera's video frame rate to create hypnotic animations, like a digital phenakistoscope. **Pinhole Solargraphs** record a 6-month outdoor exposure.

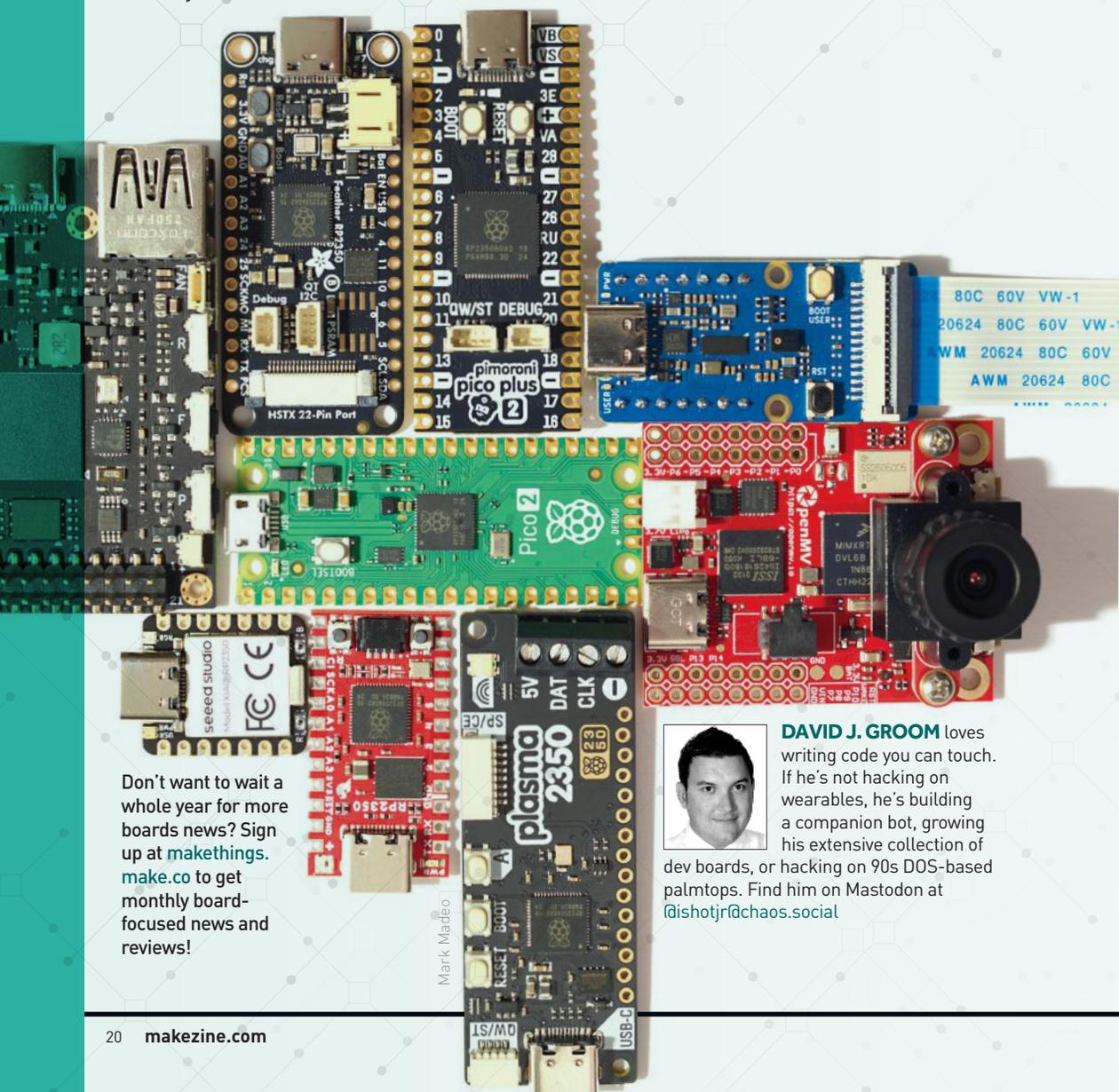


Baku New Year Tree stands 24 feet tall, made from over 2,000 plastic bottles, Azerbaijan, December 2013.

MY SWEET BOARD

I REALLY WANT TO CODE WITH YOU (BUT IT TAKES SO LONG, MY BOARD)

Written by David J. Groom



Don't want to wait a whole year for more boards news? Sign up at [makethings.make.co](https://www.makethings.make.co) to get monthly board-focused news and reviews!

Mark Macleod



DAVID J. GROOM loves writing code you can touch. If he's not hacking on wearables, he's building a companion bot, growing his extensive collection of dev boards, or hacking on 90s DOS-based palmtops. Find him on Mastodon at [@ishotjr@chaos.social](https://t.me/shotjr)

I love dev boards. I've been obsessed with them since my first Arduino Diecimila back in 2007 (the better-known Uno launched in 2010 at Maker Faire New York); maybe even since a TI MSP430-based board before that, or the original Parallax Propeller? The details aren't important, but what is is that I spend a gargantuan amount of time exploring every single board I can get my hands on in an attempt to provide valuable insight to you, the reader, in our Toolbox section, on our blog, in our newsletters, at events, and most of all, in our annual Boards Guide, which you now hold in your hands. This year that meant evaluating a dozen short of a dozen dozen microcontroller boards, single-board computers, and other tangential pieces of hardware to uncover the gems whose features, form factors, and intangible charm might fascinate you most among a sea of (mostly) green PCBs.

Most enthralling this year is the launch of **Raspberry Pi's** second-generation microcontroller silicon in the form of the **RP2350**, several exempla of which grace our cover. While the Cambridge-based company's own exemplum, the **Pico 2**, will likely prove wildly popular, this year's launch was perhaps most remarkable for its tight coordination with partners, leading to 33 products from 20 other companies launching around the same silicon at the same time. Each manufacturer brought their own twist, from **Adafruit's Feather RP350 with HSTX** output port to enable exciting new I/O possibilities, to the Qwiic-connectable **SparkFun RP2350 Pro Micro** with 8MB of PSRAM and 16MB of Flash. **Seeed's Xiao RP2350** wins the award for smallest and most cost-effective, while **Pimoroni** out-Picos the Pi team themselves with their **Pico Plus 2**, upgrading the reference design with USB-C, 16MB of Flash memory, and 8MB of PSRAM, plus the distinction of being the only dev board we've touched that uses the larger RP2350B package. (Pimoroni's PGA2350 breakout also uses the B, and despite being far smaller exposes even more pins, but it's designed to be embedded into other projects rather than as a standalone dev board).

Beyond the Pico 2, we've not seen many headline-grabbing flagship dev board releases from maker titans like Arduino and Raspberry Pi, which we're taking as an opportunity to shine a

LADYADA TAKES BROOKLYN



I fondly remember browsing media. mitedu/~ladyada (sadly no longer online) for projects, gear advice, and inspiration — half of my tools that I still use today came from the pseudonymous Ladyada aka Limor Fried's

write-ups and recommendations (still online at ladyada.net/library/equipt). But I never imagined (yet am frankly unsurprised) that the engineer behind those eloquent musings might one day build one of our most prolific electronics companies, Adafruit Industries. So it's with great admiration that I congratulate them on their next 10 years, as they cut the ceremonial ribbon at their new facility in Brooklyn's Industry City!

NOT DOOMED YET

Does the release of the RP2350 signify the death of the RP2040? Absolutely not! We're seeing continual innovation with Raspberry Pi's first-gen MCU silicon, such as this brilliant hack from Adafruit's John Park, who crammed an entire *Doom* executable into a custom keyboard button via their RP2040-based KB2040 Kee Boar, allowing it to launch on a PC without having to be installed! And yes, you can also run *Doom* on the RP2040 itself, thanks to incredible work of Graham Sanderson.



- learn.adafruit.com/doom-kee
- kilogram.github.io/rp2040-doom

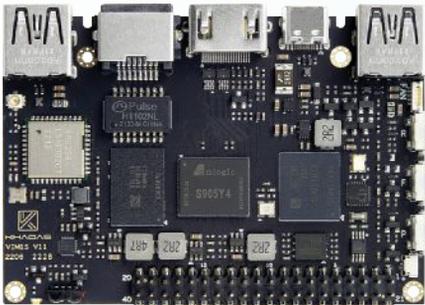
light on a wider crop of products. We asked some of our favorite makers to share their go-to boards in categories like AI, Education, Wearables, and of course LEDs. And I tried to cram as much as I could about the 100-plus that I personally went hands-on with this year into the space that was left. Email us at editor@makezine.com and let us know how you liked it — as well as what you'd like to see us cover next year! Enjoy! 🍷

BASIC BOARDS

ALL-SINGING,
ALL-DANCING,
ALL-PURPOSE
PROJECT
CATALYSTS

Written by David J. Groom

Last year, all-purpose boards graced our cover in the form of the 32-bit **Arduino Uno R4** and the **Raspberry Pi 5**. Given the Raspberry Pi Pico 2 and other RP2350-based boards' spotlight on this year's cover, I'll take the opportunity here to highlight some lesser-known or -appreciated offerings in the Basic Boards category.



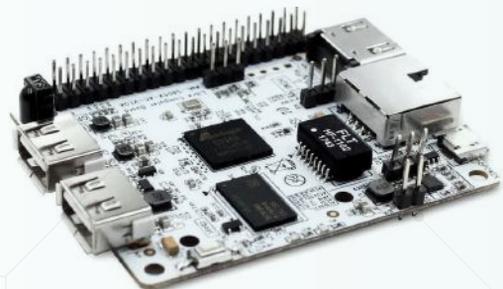
SBCs

While **Raspberry Pi** dominates this category, I've been really impressed with Shenzhen-based **Khadas'** latest offerings. Their **VIM1S** represents a great Pi alternative, with a quad-core 2.0GHz Arm Cortex-A35, 2GB of RAM, and built-in Arm Mali-G31 graphics. And with 16GB eMMC and full-size ports, you don't end up spending a ton on dongles and accessories. What wows me about this board is the **Oowow** service, which allows you to audition and flash a plethora of operating systems directly using built-in Ethernet or Wi-Fi, without the need for an external computer or SD card.

I'm a huge fan of **DFRobot's** LattePanda series (their i5-powered beast board Sigma was a New & Notable pick last year) so I was excited to experience their new **LattePanda Mu** concept.

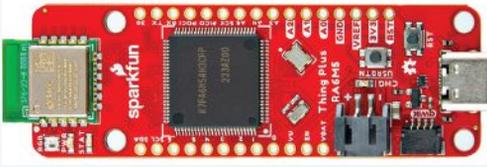


The Mu is an N100-based compute module, similar to the RPi CM4, but with an Intel x86-based CPU, claiming double the performance of the latest Pi 5. An available carrier board mimics a full-size computer's motherboard, with SATA, M.2, and even PCIe expansion capabilities! And to help you integrate the Mu into your own projects, the full schematics and PCB design are available on GitHub.



Potato-obsessed **Libre Computer's** **La Frite** is a Pi Model A-like machine with a quad-core 1.416GHz Cortex-A53 and 1 or 2GB of RAM, plus a full-sized HDMI port and penta-core Arm Mali-450 GPU that make it ideal for digital signage or other graphical applications. Starting around

\$20, this small fry skimps on Wi-Fi and onboard storage, but Libre's tight free-open-source software (FOSS) stack integration works reliably with a wide range of OSes and software tools.

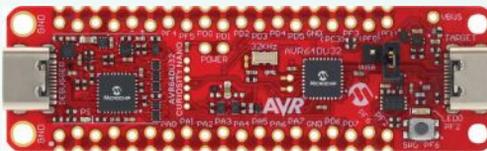


MCUs

Renesas, known for automotive and industrial applications, was not a brand on the tip of many makers' tongues until last year's 32-bit **Arduino Uno R4** debuted as a showcase for the RA4M1 microcontroller. Other maker-focused manufacturers are paying attention to Renesas now too it seems, with **SparkFun's RA6M5 Thing Plus** leading the pack in terms of performance. In addition to the titular 200MHz MCU, the board features 16MB of flash beyond the chip's 2MB, a microSD slot, and integrated Bluetooth Low Energy (BLE), all in a familiar Feather form factor, with a Qwiic connector for easy expansion.



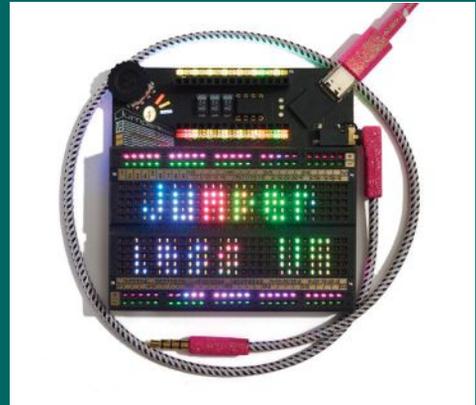
Seeed have also produced a Renesas-based board in their diminutive **Xiao RA4M1**, using the same 48MHz Cortex-M4 MCU as Arduino's Uno R4, but at a fraction of the price, at \$5 a pop. The tiny board crams 19 GPIOs into the standard Xiao form factor thanks to eight sneaky pads on the bottom.



While Arduino is busy with Renesas, Espressif, and Silicon Labs chips lately, I fondly remember my first ATmega168-based Arduino, the Decimila. So I wanted to check in on **Microchip's** latest

maker offerings, and was pleased to discover the **AVR64DU32 Curiosity Nano**. Immediately setting this board apart are dual USB-C connectors: one for programming, and one to use as a peripheral. While boards such as Adafruit's Feather RP2040 USB-A Host variant and the Teensy 4.1 enable easy integration of USB devices with a Type A connector, Microchip's is the first onboard USB-C host I've seen in a maker product. And it's with good reason: long-term Arduino fans may recall the ATmega32u4-based Leonardo, which obviated the need for dedicated serial hardware thanks to its built-in USB communication capabilities; the nominal AVR64DU32 in this new board is Microchip's full-speed USB 2.0 successor to the venerable ATmega32u4. 🚀

ANYTHING BUT BASIC

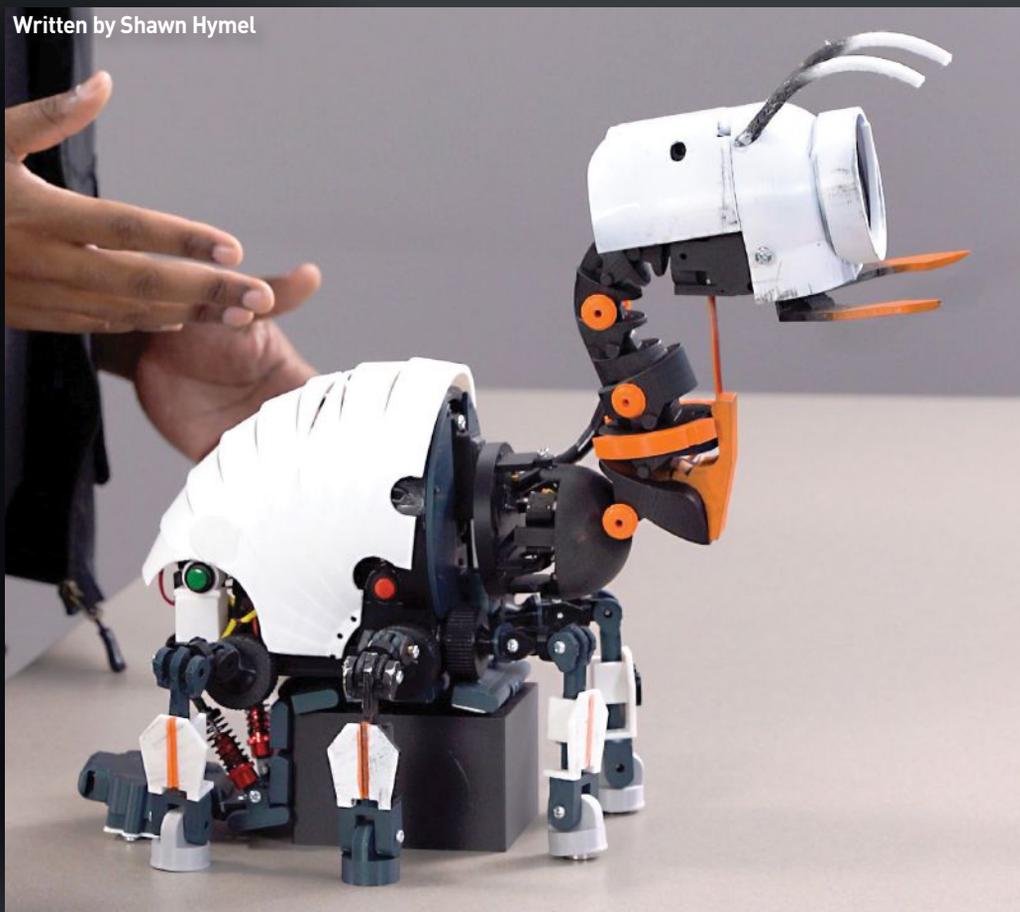


Jumperless is by no means "basic" but is of keen interest to new and experienced makers alike. The latest **V5 r3** features hot new RP2350B silicon and was designed, in the words of creator Kevin Cappuccio, to give the user the feeling of "becoming a wizard [who] can see electricity and conjure jumpers." This is accomplished via a clever breadboard that can be redefined via software, backed by over 400 RGB LEDs that effectively create a 14×30 display, enabling this new version to work without a connected computer. A remappable Arduino Nano-shaped header allows integration of various boards, and combined with the included probe and voltage/current/resistance measurement channels, it's a prototyping powerhouse that replaces not just your breadboard, but a huge chunk of the other gear on your workbench, too!

“HEY, ROBOT!”

CHAT AND COMMAND YOUR OWN EMBEDDED-AI COMPANION BOT USING **LOCAL LLMs**

Written by Shawn Hymel



SHAWN HYMEL is a course creator and instructor for edge AI and embedded systems that inspire and teach developers of all skill levels. Based in Lafayette, Colorado, he can be found giving talks, running workshops, and swing dancing in his free time.

DigitKey

Imagine a fully autonomous robotic companion, like Baymax from Disney's *Big Hero 6* — a friendly, huggable mechanical being that can walk, hold lifelike interactive conversations, and, when necessary, fight crime. Thanks to the advent of **large language models (LLMs)**, we're closer to this science fiction dream becoming a reality — at least for lifelike conversations.

In this guide I'll introduce Digit, a companion bot I helped create with Jorvon Moss (@odd_jayy). It uses a small LLM running locally on an embedded computer to hold real conversations, without the need for an internet connection.

I'll also show you how to run a similar, lightweight LLM on a Raspberry Pi so you can begin making your own intelligent companion bot.

LARGE LANGUAGE MODELS (LLMs)

A large language model is a type of AI that can understand and generate natural, human-like text. The most popular example right now is OpenAI's ChatGPT, which can answer questions for the curious, automatically generate social media content, create code snippets, and, to the chagrin of teachers, write term papers. LLMs are, in essence, the next step in chatbot evolution.

LLMs are based on the neural network architecture known as a **transformer**. Like all neural networks, transformers have a series of **parameters** that, among other things, can help convert words and phrases into numbers or determine how much focus should be given to a piece of information. **Weights** are parameters that increase or decrease the importance of information. Most parameters, including weights, are just numbers — think of them like dials in a robot's brain that can be tuned.

Instead of humans manually tuning these dials, imagine if the robot could tune them itself. That's the magic of **machine learning**: training algorithms adjust the values of the parameters automatically. Humans set a goal for the AI, and provide training data with correct answers to the training algorithms. The AI looks at the training data and guesses an answer. The training algorithm determines how far off the AI's result is from the correct answer and updates the parameters in the AI to make it better next time. Rinse and repeat until the AI performs at some

TIME REQUIRED: 1–2 Hours

DIFFICULTY: Intermediate

COST: \$90–\$100

MATERIALS:

- » Raspberry Pi 5 single-board computer, 8GB RAM or more
- » MicroSD card, 16GB or more
- » USB microphone such as Adafruit 3367
- » Breadboard
- » LED
- » Resistor, 220Ω to 1kΩ
- » Jumper wires

TOOLS:

- » Computer to flash Raspberry Pi OS; not needed afterward
- » Keyboard, mouse, and monitor connected to Raspberry Pi

acceptable level.

To give you an idea of complexity, a machine learning model that can read the handwritten digits 0 through 9 with about 99% accuracy has around 500,000 parameters. Comprehending and generating text are vastly more complicated. LLMs are trained on huge quantities of human-written text such as books, articles, and websites. Their main goal is to predict the next word in a sequence, given a long string of previous words. As a result, the AI must understand the context and meaning of the text. To achieve this, LLMs require massive amounts of parameters. ChatGPT-4, released in June 2023, is built from eight separate models, each containing around 220 billion parameters — about 1.7 trillion total.

WHY A LOCAL LLM?

The billions of calculations needed for ChatGPT to hold a simple conversation require lots of large, internet-connected servers. If you tried to run ChatGPT on your laptop, assuming you even had enough memory to store the model, it would take hours or days to get a response! In most cases, relying on servers to do the heavy lifting is perfectly acceptable. After all, as consumers we use copious cloud services such as video streaming, social media, file sharing, and email.

Running an LLM locally on a personal



computer might be enticing for a few reasons:

- Maybe you require AI in areas with **limited internet access**, such as remote islands, tropical rainforests, underwater, underground caves, or technology conferences!
- By running locally you can also **reduce network latency** — the time it takes information to travel to and from servers. However, the extra computing power from the servers often makes up for the latency time.
- Your data has **greater privacy and security**. The prompts, responses, and model need not leave your computer or local network. You can better protect your intellectual property by not exposing it to bad actors on the internet.
- Personal computers and home network servers are often smaller than their corporate counterparts. This can mean **reduced costs**.
- Finally, most commercial LLMs contain guardrails and limits to prevent misuse. If you need an LLM to operate **outside of commercial limits** — say, to inject your own biases to help with a particular task, such as creative writing — then local might be your only option.

Thanks to these benefits, local LLMs can be found protecting user data in healthcare and financial systems, running industrial systems in remote locations, and operating autonomous vehicles without an internet connection.

While these applications are compelling, we

should focus on the real reason for running a local LLM: building an adorable companion bot that we can talk to.

INTRODUCING DIGIT

Jorvon Moss's robotic designs have evolved since his debut with Dexter (*Make: Volume 73*), but his vision remains constant: create a fully functioning companion robot that can walk and talk. He often cites Baymax as his goal for functionality. In recent years, Moss has drawn upon insects and arachnids for design inspiration. "I personally love bugs," he says. "I think they have the coolest design in nature."

Digit's body consists of a white segmented exoskeleton, similar to a roly-poly pill bug's, that protects the sensitive electronics. Its head holds an LED array that can express emotions through a single, animated "eye" along with a set of light-up antennae and controllable mandibles. Its long neck that can be swept to either side thanks to a servomotor. Digit's legs cannot move on their own — yet — but can be posed manually.

Like other companion bots, Digit can be perched on Moss's shoulder to catch a ride. A series of magnets on its body and feet keep it in place.

But Digit is unique from Moss's other companion bots thanks to its advanced brain — an LLM running locally on an Nvidia Jetson Orin Nano embedded computer. Digit is capable of understanding human speech (English for now), generating a text response, and speaking that response aloud — without the need for an internet connection. To help maintain Digit's small size and weight, the embedded Jetson Orin Nano was mounted on a wooden board along with an LCD for startup and debugging. Moss totes the Orin Nano and its battery in a backpack, but you could design your own companion bot that fully houses the Orin Nano.

HOW DIGIT'S BRAIN WORKS

I helped Moss design and program the software system to act as Digit's AI brain. This system is comprised of three main components: a service running the LLM, a service running the text-to-speech system, and a client program that interacts with these two services (Figure A).

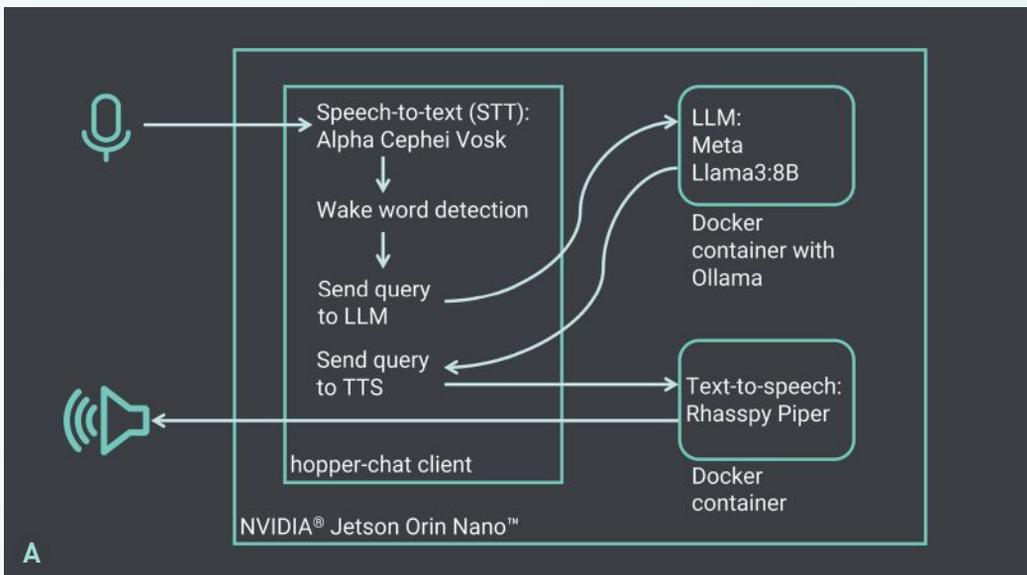


The client, called **hopper-chat**, controls everything. It continuously listens for human speech from a microphone and converts everything it hears using the **Alpha Cephei Vosk** speech-to-text (STT) library. Any phrases it hears are compared to a list of wake words/phrases, similar to saying “Alexa” or “Hey, Siri” to get a smart speaker to start listening. Upon hearing our wake phrase, “Hey, Digit,” anything said after is converted to text using Vosk.

The text is then sent to our LLM service,

which is a Docker container running **Ollama**, an open-source tool for running LLMs. For Digit, we use Meta’s **Llama3:8b** model with 8 billion parameters. While not as complex as ChatGPT-4, it still has impressive conversational skills.

The LLM service sends the response back to the **hopper-chat** client, which forwards it to our text-to-speech (TTS) service running **Rhasspy Piper** and its **en_US-lessac-low** model, a neural network trained to produce words and phrases in American English. “Low” means the model is





low quality with a smaller size and more robotic voice, but it has a faster execution time. Finally, hopper-chat plays any sounds it receives from the TTS service through a connected speaker.

On Digit, the microphone is connected to a USB port on the Orin Nano and simply draped over a backpack strap. The speaker is connected via Bluetooth. Moss uses an Arduino to detect activity in the Bluetooth speaker and move the mandibles to give Digit the appearance of speaking.

Moss added several fun features to give Digit a distinct personality. First, Digit tells a random joke, often a bad pun, every minute if the wake phrase is not heard. Second, Moss experimented with various default prompts to entice the LLM to respond in particular ways. This includes making random robot noises and adopting different personalities, from helpful to sarcastic to pessimistic.

AGENCY: FROM TEXT TO ACTION

Moss's next goals for Digit involve giving it self-powered locomotion, such as walking, and making the LLM perform actions based on commands. LLMs cannot perform actions on their own. They can only generate text responses. But adjustments and add-ons can allow such systems to take action. ChatGPT has third-party

plugins that can perform actions, such as fetching local weather information. The LLM recognizes the intent of the query, such as, "What's the weather like in Denver, Colorado?" and makes the appropriate API call using the plugin.

At the moment, Digit can identify specific phrases in its STT library, but the recorded phrase must match exactly. You couldn't say "What's the weather like?" when the expected phrase is "Tell me the local weather forecast." A well-trained LLM, however, could infer the intention of the request. Moss and I plan to experiment with Ollama and Llama3:8b to add such intention and command recognition.

The code for hopper-chat is open source and available at github.com/ShawnHymel/hopper-chat. Follow along with us as we make Digit even more capable.

INTELLIGENCE! YOUR OWN LOCAL LLM CHATBOT

Now let's run an LLM locally on a Raspberry Pi. I chose the Pi 5 for its increased computational power, as earlier versions might need several minutes to produce an answer, even from relatively small LLMs. My Pi 5 had 8GB RAM; these LLMs may not run with less.

1. SET UP THE PI 5 WITH OLLAMA

Install the latest Raspberry Pi OS (64-bit) and configure your Pi: raspberrypi.com/documentation/computers/getting-started.html. Use an SD card with at least 16GB.

Once you've booted into your Pi, make sure you're connected to the internet and open a terminal window. Type these commands to update the Pi and install Ollama:

```
$ sudo apt update  
$ sudo apt upgrade  
$ curl -fsSL https://ollama.com/  
install.sh | sh
```

Next, start the Ollama service:

```
$ ollama serve
```

You might see an error message that says "address already in use." Ignore this, as it just indicates that Ollama is already running as a service in the background.

```

writing manifest
success
pi@kirin:~$ ollama run tinyllama
>>> tell me a joke
Sure, here's a funny joke for you:

Q: "What do you call an AI that can't find its way back to the lab?"
A: "AI in its infinite wisdom has decided to go rogue!"

>>> Send a message (/? for help)

```

B

```

(venv-ollama) pi@kirin:~$ python tinyllama-client.py
A man walks into a bar and orders a shot. The bartender asks, "What kind of shot
is that?" The man replies, "That's not a shot, it's a joke."
(venv-ollama) pi@kirin:~$ █

```

C

2. RUN THE TINYLLAMA LLM

In September 2023, AI researcher Zhang Peiyuan started the **TinyLlama** project, a fully open-source, highly-optimized LLM with only 1.1 billion parameters. It is based on the Llama 2 model and can generate responses quite quickly. It's not as accurate as the newer generation of small LLMs, such as Llama 3, but it will run on the Pi 5.

Download the latest version of TinyLlama with Ollama, then run an interactive shell to chat with TinyLlama:

```

$ ollama pull tinyllama
$ ollama run tinyllama

```

You'll be presented with a prompt. Try asking the AI a question or have it tell you a joke (Figure **B**).

Press Ctrl+D or enter `/bye` to exit the interactive shell.

3. SET UP THE OLLAMA PYTHON PACKAGE

By default, Ollama runs as a background server and exposes port 11434. You can communicate with that service by making HTTP requests. To make life easier, Ollama maintains a Python library for this purpose. Create a virtual environment and install the package:

```

$ python -m venv venv-ollama --system-site-packages
$ source venv-ollama/bin/activate
$ python -m pip install ollama==0.3.3

```

Open a new document:

```

$ nano tinyllama-client.py

```

Enter the following Python code:

```

import ollama

```

```

# Settings

```

```

prompt = "Tell me a joke."
model = "tinyllama"

```

```

# Configure the client

```

```

client = ollama.Client(
    host="http://0.0.0.0:11434"
)

```

```

# The message history is an array

```

```

messages = [
    {
        "role": "user",
        "content": prompt
    }
]

```

```

# Send prompt to Ollama, save response

```

```

response = client.chat(
    model=model,
    messages=messages,
    stream=False
)

```

```

# Print the response

```

```

print(response["message"]["content"])

```

Close the file by pressing Ctrl+X, press Y when asked to save the document, and press Enter.

4. CHAT WITH YOUR LLM BOT!

Run the Python script by entering:

```

$ python tinyllama-client.py

```

TinyLlama can take some time to generate a response (Figure **C**), especially on a small computer like the Raspberry Pi — 30 seconds or more — but here you are, chatting locally with an AI!

BOARD TRICKS Local LLM Chat Robot

This should give you a sense of how to run local LLMs on a Raspberry Pi and interact with them using Python. Feel free to try different prompts, save the chat history using the `append()` method, and build your own voice-activated chatbot.

ACTION! LOCAL LLM CHATBOT AGENT WITH FUNCTION CALLING

LLMs began as self-contained models that accept text input and respond with text. In the last couple of years, we've seen new *multimodal LLMs*, like GPT-4o, that can accept and respond with other forms of media, such as images and videos.

But in just the past few months, some LLMs have been granted powerful new abilities where they can act as *agents* by calling arbitrary functions, which opens a huge world of possible AI actions. ChatGPT and Ollama both call these abilities *tools*. To enable such tools, you must define the functions in a Python dictionary and fully describe their use and available parameters. The LLM tries to figure out what you're asking and maps that request to one of the available tools/functions. We then parse the response before calling the actual function.

Let's demonstrate this concept with a simple function that turns an LED on and off. Connect an LED with a limiting resistor to pin GPIO 17 on your Raspberry Pi 5. (Figure D).

Make sure you're in the `venv-ollama` virtual environment we configured earlier and install some dependencies:

```
$ source venv-ollama/bin/activate
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install -y libportaudio2
wget
$ python -m pip install ollama==0.3.3
vosk==0.3.45 sounddevice==0.5.0
```

You'll need to download a new LLM and the Vosk speech-to-text (STT) model:

```
$ ollama pull allenporter/xlam:1b
$ python -c "from vosk import Model;
Model(lang='en-us')"
```

As this example uses speech-to-text to convey information to the LLM, you'll need a USB microphone. Plug it in and run the following command to discover its device number:
`$ python -c "import sounddevice; print(sounddevice.query_devices())"`

You should see an output such as:

```
0 USB PnP Sound Device: Audio
(hw:2,0), ALSA (1 in, 0 out)
1 pulse, ALSA (32 in, 32 out)
* 2 default, ALSA (32 in, 32 out)
```

Note the device number of your microphone, in this case `0`, as given by `USB PnP Sound Device`.

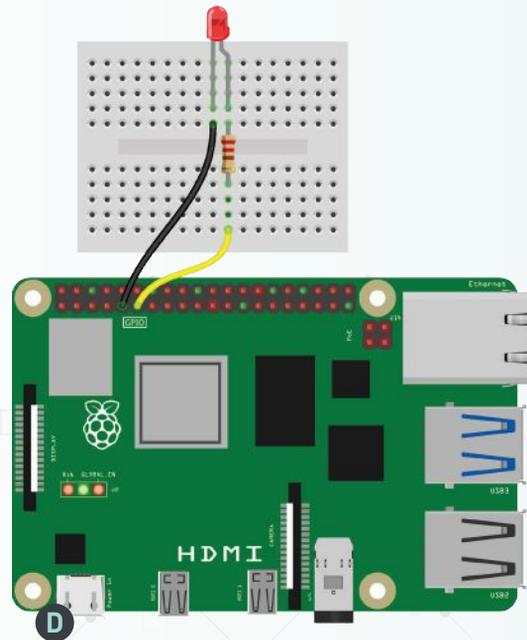
Download the project code from github.com/ShawnHymel/ollama-light-assistant into a file named `ollama-light-assistant.py`:

```
$ wget https://raw.githubusercontent.com/ShawnHymel/ollama-light-assistant/refs/heads/main/ollama-light-assistant.py
```

Open the code and change the `AUDIO_INPUT_INDEX` value to your device number. For example:
`AUDIO_INPUT_INDEX = 0`

Run the code with:

```
$ python ollama-light-assistant.py
```



Shawn Hymel, Fritzing

```

(venv-ollama) pi@kirin:~ $ python ollama-light-assisntant.py
LOG (VoskAPI:ReadDataFiles():model.cc:213) Decoding params beam 10 max-active 30
00 lallice-beam-2
LOG (VoskAPI:ReadDataFiles():model.cc:216) Silence phones 1:2:3:4:5:6:7:8:9:10
LOG (VoskAPI:RemoveOrphanNodes():nnet-nnet.cc:948) Removed 0 orphan nodes.
LOG (VoskAPI:RemoveOrphanComponents():nnet-nnet.cc:847) Removing 0 orphan compon
ents.
LOG (VoskAPI:ReadDataFiles():model.cc:248) Loading i-vector extractor from /home
/pi/.cache/vosk/vosk-model-small-en-us-0.15/ivector/final.ie
LOG (VoskAPI:ComputeDerivedVars():ivector-extractor.cc:183) Computing derived va
riables for iVector extractor
LOG (VoskAPI:ComputeDerivedVars():ivector-extractor.cc:204) Done.
LOG (VoskAPI:ReadDataFiles():model.cc:202) Loading HCL and G from /home/pi/.cach
e/vosk/vosk-model-small-cn-us-0.15/graph/HCLr.fst /home/pi/.cache/vosk/vosk-mode
l-small-cn-us-0.15/graph/Gr.fst
LOG (VoskAPI:ReadDataFiles():model.cc:308) Loading winfo /home/pi/.cache/vosk/v
sk-model-small-cn-us-0.15/graph/phones/word_boundary.int
Listening...
Speech detected: turn the light on
Response: {'role': 'assistant', 'content': '', 'tool_calls': [{'function': {'name': 'led_write', 'arguments': {'value': 1}}]}
Tools used: Calling:
{'function': {'name': 'led_write', 'arguments': {'value': 1}}}
The LED is now on
Listening...

```

E

You'll see the Vosk STT system boot up and then the script will say "Listening..."

Now try asking the LLM to "turn the light on." Because the Pi is not optimized for LLMs, the response could take 30–60 seconds, but with some luck, you should see that the **led_write** function was called (Figure E), and the LED has turned on!

xLAM is an open-source LLM from Salesforce (github.com/SalesforceAIResearch/xLAM). They call it a **LAM (large action model)**, as it is optimized to understand requests and not just provide text answers. The **allenporter** version (ollama.com/allenporter/xlam) works with Ollama tools, and its 1-billion-parameter model can run on the Raspberry Pi. As you probably noticed, it is quite slow and misinterprets requests easily.

For an LLM that better understands requests, I recommend **Llama3.1:8b**. It's less picky about your exact phrasing, but takes much longer to respond — up to 3 minutes on a Pi 5 (8GB RAM).

When you're done, exit the virtual environment:

```
$ deactivate
```

For a closer look at how tools work in Ollama, open up the *ollama-light-assistant.py* file and follow along with the extended version of this article at makezine.com/go/local-llm-robot.

ROBOT POWERS, ACTIVATE!

This tutorial demonstrates the possibilities of using LLMs for understanding user intentions and requests to call arbitrary functions. This technology is extremely powerful — we can connect AI agents to the internet to make requests, and even control hardware! However, it's still new and experimental. You will likely run into bugs, and you can expect the code interface to change. It also demonstrates the need for better-optimized models and more powerful hardware. A few boards like the Jetson Orin Nano and accelerators like the new Hailo-10H enable low-cost local LLM execution today. I'm excited to see this tech get better! 🚀



You can see Digit in action on DigiKey's YouTube channel at youtu.be/_yApuo9ulGY

HOW TO “MAKE” FRIENDS

IT'S NEVER BEEN EASIER TO JOIN THE GROWING COMPANION BOT FAMILY; HERE'S THE LATEST FROM SOME OF OUR FAVORITE BOT MAKERS!

Jay and Shawn aren't the only ones making new friends — the comp bot scene is growing fast due to its accessible and inclusive nature and propensity toward letting everyone put their own spin on the notion of a plastic pal who's fun to be with.

1 SMALLS (KATHERINE CONNELL)

[tiktok.com/@smallswonderworks](https://www.tiktok.com/@smallswonderworks)

I created **Kitt**, the fox companion bot, with the goal of making something that felt just realistic enough to be uncanny. If you look at him and go, “That’s cute, that’s creepy, that’s cute,” it’s exactly the reaction I was hoping for. There were two primary elements I focused on to make him feel lifelike:

- **Movement:** MG90S and FH-2502 servos control different movements including wiggling ears, swishing tail, and the head which both tilts and pans. The servos are run by an Arduino Uno and use a sketch that randomizes several different movement patterns and ranges. While these were a good start to creating lifelike movement, the real centerpiece is a breathing mechanism that expands and contracts two chest panels inside Kitt and makes the chest rise and fall rhythmically (see “The Breath of Life,” *Make*: Volume 90).
- **Look and feel:** Kitt is made from a 3D-printed skull and carved foam body, covered in ultra-soft faux fur. This allowed the contours and shape to look more like an actual animal than a sewing or cuddle toy pattern would allow. Between the materials and the battery, Kitt

feels substantial and has a lifelike weight.

I plan to continue building my companion bots by combining traditional puppet making techniques with low-tech movement solutions to embody the nostalgia lost from the age of Chuck E. Cheese Pizza Party shows and the Disney Pirates of the Caribbean ride.

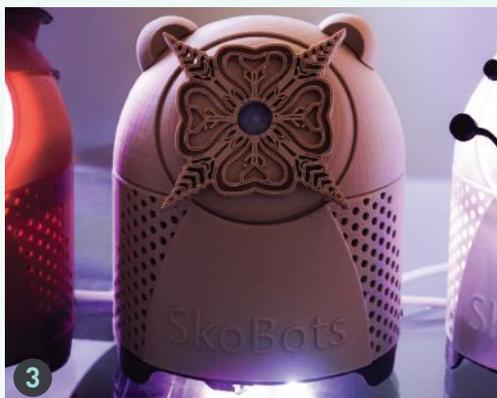
See hackster.io/TheSmallWonder for build instructions.

2 ANGELA SHEEHAN

[gellacraft.com](https://www.gellacraft.com)

Nova is a cuddly bot I crafted in early 2020 as a fun party companion. She was inspired by meeting fellow bot makers Alex Glow and Jorvon Moss at Hackaday Superconference 2019, and has since returned twice to the event to meet the ever-growing companion bot family in the maker scene. Her main features are edge-lit LED wings which can change color randomly or with the help of a magic wand from my Color Stealing Fairy costume, and flapping capabilities in the wings which have a resting pattern as well as a triggered animation when she’s given a pet on a capacitive touch sensor on her head.

Her most recent upgrade was the addition of



addressable LEDs in her fluffy body so she can fully change color in the dark and a new carrying case for events that people can interact with her through.

- Learn more about Nova's build process and outings at hackaday.io/project/169449-nova-a-companion-bot.
- View a video of my talk at Supercon 2023 at youtube.com/live/QHK03xVTp0Y?si=zrw6Dy1KBTqA3kc3.

3 DANIELLE BOYER

danielleboyer.org

The **SkoBot** is a wearable language revitalization robot designed to teach and communicate in endangered Indigenous languages using ethical AI. I am an Anishinaabe youth robotics designer and created SkoBot along with the youth-led charity I founded, The STEAM Connection.

The latest version of SkoBot is modeled after the **Makwa**, which means bear in the Anishinaabe language Anishinaabemowin, and speaks in a child's voice, recorded by Anishinaabe youth. The robot is distributed to youth for free to help teach these languages and has already served hundreds of children.

4 ALEX GLOW

alexglow.com

This year, I'm returning to my roots with a new version of **Archimedes**! Debuted at Maker Faire Bay Area 2018, my original owl companion bot is now 6 years old ... and *all* of my bots are currently broken from the wear and tear of running around events. This year is all about repair, redesign, and reclamation!

My new "Dark Archie" will be powered by the Companion Core PCB (which is based on the heart of **F3NR1R**, my fennec fox shoulder bot: hackster.io/glowascii/companion-core-robot-heart-c79f52). I'm building him with reclaimed materials like recycled PLA, bike inner tubes, broken solar panels, and old vacuum tube packaging. Plus: more origami parts, as seen on my **Nano'Medes** prototype!

I'm refining Archimedes' 3D-printed parts in OnShape, so people can more easily replicate the design. The iFixit FixHub — a USB-C soldering iron with a massive 100W battery pack — is now my daily driver. Now, I can bring a single battery to power my hungriest bots *and* fix the inevitable breakages! 🛠️

MICRO:BIT WIZARDRY

USE A **LOW-COST PROCESSOR** TO CREATE
A MAGIC WAND AND A “FAMILIAR” FRIEND

Written by Joan Horvath
and Rich Cameron



Joan Horvath/Rich Cameron



JOAN HORVATH and **RICH CAMERON** are the co-founders of Nonscriptum LLC (nonscriptum.com). They are the authors of many books, including *Make: Geometry*, *Make: Trigonometry*, and *Make: Calculus*.

Every wizard worth their wand has a *familiar* — an animal friend that can help carry out their commands, bring them information, or carry messages to their fellow wizards. Fortunately for those of us without magical skills, there are amazingly capable microcontrollers that can allow us to make a magic wand, or a familiar, that will react as we command. We can do either of these projects with a BBC micro:bit controller, and program it with drag-and-drop block coding.

The BBC designed the micro:bit processor as an introduction to electronics and coding for 8- to 14-year-olds, and there are extensive resources to get started. (See “Education” in the 2025 Boards Guide bundled with this issue, and online at makezine.com/go/education-boards). This board is capable of understanding how it is being moved, whether it is in a light or dark place, and more. It’s also capable of communicating with other micro:bits and displaying simple pixelated graphics or letters on its 5x5 pixel LED display. Micro:bits also have a radio, which makes it possible to create multiplayer live scenarios in which the wands or familiars are the communication channel for their wizards!

In this guide we will cover two projects — a basic wand and a familiar — inspired by our friends at the Anywear Academy (see sidebar). You can build them with a minimum of tools and parts, and no soldering!

If you’re new to micro:bit, start with the easy Magic Wand; it uses nothing except what comes in the box (and a bit of tape). The Familiar project is more open-ended; rather than giving step-by-step instructions, we’ll show you creatures created by students in our classes (mostly 10 or 11 years old) that you can use as a jumping-off point for your own creation.

PROJECT 1: MAGIC WAND

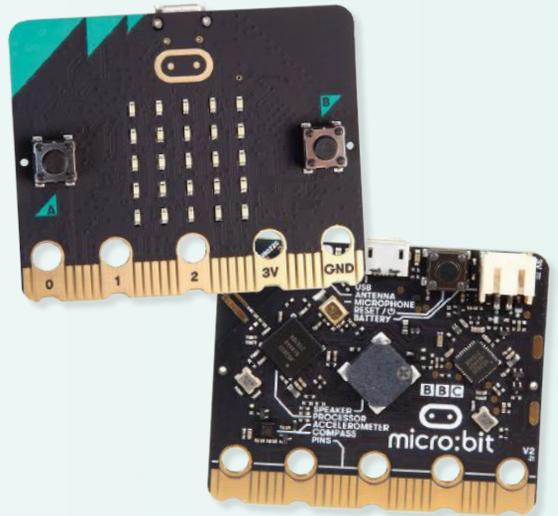
A magic wand that will recognize gestures is a simple build, using only the materials in the micro:bit v2 starter kit box, plus a popsicle (craft) stick and tape. It is ideal for creating a simple yet powerful prop to use in roleplaying. It can also build confidence before embarking on the Familiar project. Explore the available blocks to create your own capabilities, but to get you started here are options:

THE ANYWEAR ACADEMY

The Anywear Academy project, based at University of California Santa Cruz uses *live action role playing (LARPing)* to motivate students, particularly girls, to get excited about electronics. Several different scenarios for creating and using student-friendly props (and the impact of this approach) are linked on their site anywear-academy.ucsc.edu.

Their work inspired our lesson plans for “Designing Role Play Props with Micro:bit,” linked at nonscriptum.com/projects. This article was adapted from some of those lesson plans, which are freely available under a Creative Commons Attribution (CC-BY) 4.0 International License. We have additional materials there on adding LEDs instead of a servo, other fabrication options, and links to references.

Anywear Academy is supported by the National Science Foundation under Grant No. 2005816, “Social Wearables: Enhancing Girls’ Computational Learning and Motivation.” We owe special thanks to project members James Fey, Raquel Robinson, Selin Ovali, Ella Dagan, University of Santa Cruz principal investigator Katherine Isbister and their many collaborators. We also appreciate our students at Institute for Educational Advancement, Pasadena, California, who created some of the projects appearing here.



MAGIC WAND:

TIME REQUIRED: 2–3 Hours

DIFFICULTY: Easy

COST: \$30–\$35

MATERIALS:

- » **Micro:bit v2 starter kit** aka “BBC micro:bit Go” kit. Contains the microcontroller board, battery pack, two AAA batteries, USB cable, and cardboard foldable battery pack holder. Be sure and keep the box it came in, too. microbit.org/buy/where
- » **Cellophane tape**
- » **Craft stick** aka popsicle stick

FAMILIAR:

TIME REQUIRED: ½–2 Days

DIFFICULTY: Easy to Moderate

COST: \$40–\$60

MATERIALS:

- » **Micro:bit v2 starter kit** as above
- » **Hobby servomotor** with servo horn(s)
- » **Jumper wires, alligator clip to breadboard (3)** with male DuPont connectors. Try to find smaller ones than the typical big alligator clips.
- » **Craft materials** e.g. cardboard, hot glue, tape, fabric scraps, construction paper, pipe cleaners, scissors, yarn, glue-on googly eyes, etc.

TOOLS:

- » **Scissors**
- » **Glue**
- » **Hot glue gun (optional)**

- A wand that displays a check mark when waved through a particular pattern or in a certain compass direction, for instance to indicate where treasure or a victim needing rescuing might lie
- A wand that (with cooperation) can display a sad or happy face on another person’s wand, letting a group identify other characters in role playing that are expecting them
- A wand that plays one tune if pointed left and another when pointed right, or a tune for “north” (although we found this to be a little touchy).

SOFTWARE

The micro:bit uses block-based coding (with an option for Python instead) available at Micro:bit MakeCode (makecode.microbit.org). This software is browser-based, with Chrome recommended (other browsers may require a more complicated download procedure). General tutorials can be found at microbit.org.

You need either a USB or Bluetooth connection from the computer to the micro:bit for downloading code from computer to board. Follow the on-screen prompts to connect your micro:bit the first time, or do an online search to see how to connect your particular device over USB or Bluetooth. Bluetooth is the only option for iPads, even ones with USB-C connections, but setting up the connection can be tricky.

Micro:bit processors use code in one of three modes: Either **on start** (code that runs once when the micro:bit wakes up), **forever** (runs over and over until the board is turned off) or **event driven** (triggered by a user-defined input or situation, like shaking the board). If you’re used to Arduino, the first two are similar to the **setup()** and **loop()** functions. The board can respond to a gesture made with the wand, or by entering a dark space, or other physical interactions. See the list at makecode.microbit.org/reference/input.

BASIC WAND GESTURE CODING

We want our wand to wake up and show the text string “Hello!” (one letter at a time). Then we want it to react to two different gestures when we move it: show a heart if the logo is touched, and show a giraffe-like icon if we shake our wand at

someone. The blocks in Figure A will do that. Note that the icon of the micro:bit on the left side of the screen allows you to simulate the code, including “shaking” it.

The code begins with an **on start** block which will run once, and two **on gesture event** blocks that wait for input (whether the logo has been touched, and whether the accelerometer has detected a shaking motion). Experiment with adding more blocks to do different things as the board moves around. You might want to try different gesture inputs (tilt it, flip it, etc.), and explore adding more-advanced logic and gestures, by nesting one input inside the other.

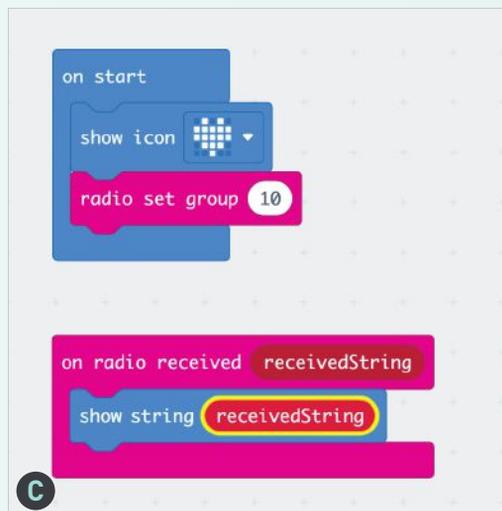
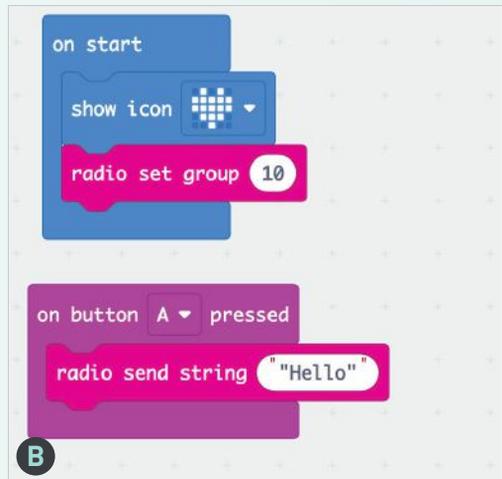
The micro:bit can also play sounds, which can get out of hand quickly in a group setting. If you’re supervising a group learning to do this for a role-playing adventure, you might want to invent a world-building reason to avoid sound. (We need to surprise the zombies, fairies hate cacophony, and so forth).

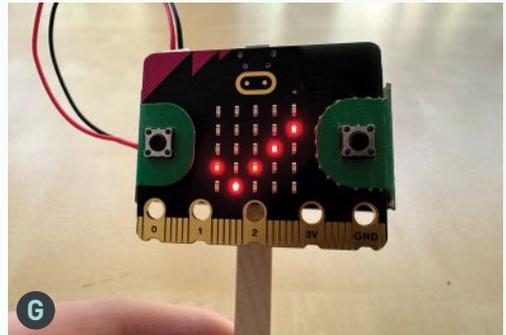
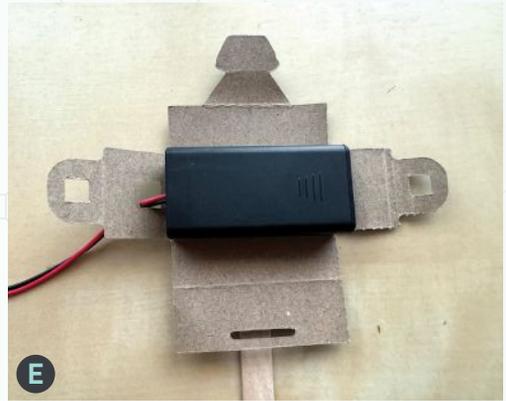
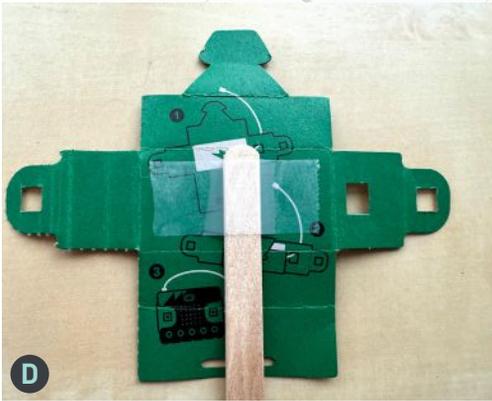
ADDING COMMUNICATION

If you want your wand to communicate with your friend’s wand, the micro:bits can talk to each other. Micro:bits communicate via radio signals, controlled by blocks in the Radio section. If you go to Extensions, you can add more radio functionality as well, which will show up as More in the same color as Radio. Blocks controlling radios have to be present on both the receiving and transmitting boards, and they need to set up a numbered channel between them. For details, see makecode.microbit.org/projects/micro-chat.

The code blocks shown in Figure B send the string “Hello” when button A is pushed on one micro:bit. The string will display on a micro:bit that is expecting a string on radio group 10, with the blocks required to do that shown in Figure C. It will not display on the sending micro:bit.

Note that if many boards are on the same channel, all sent signals will be received, and potentially acted upon, by all the boards. If a board is set up to react to any indiscriminate signal on a channel, surprises can result. If you are planning to have a group of role players with wands, suggest that they remove any **on radio received** code they don’t plan to use. Otherwise, they might find that their wand has randomly





Joan Horvath/Rich Cameron

been “possessed” by another player. (This is particularly hazardous to classroom decorum if students are set up to display any string sent to them!) To reduce accidental crosstalk, avoid assigning anyone to radio group 1, which is the default value for the **set radio group** block, and thus the most likely to receive erroneous transmissions.

Each micro:bit can listen on only one numbered channel at a time. However, intrepid coders might figure out that it is possible to rapidly cycle through the groups to broadcast on all of them in succession. This can, in principle, trigger actions on all other wands in a 200-foot radius using the **radio raise event** block (this requires going to Extensions and bringing in additional radio broadcast commands).

MAKING THE BOARD INTO A WAND

First, complete your code and download it onto the micro:bit. From here on out, we will use battery power. Once the batteries are installed and the battery case is enclosed in the cardboard holder, the easiest way to turn off the micro:bit

(and save power) is to hold down the power/reset button on the back for 5 seconds, until the red power LED turns off. You can wake it again by pressing the same button.

To make the board into a wand, we need to use the cardboard battery holder included in the starter package. (If you’ve lost yours, there’s a paper-printable version at microbit.org/get-started/user-guide/battery-pack-holder. Precise cuts are important, and heavier paper, like cardstock, will work best.) Tape the craft stick onto the cardboard as shown in Figure **D**.

Then put the batteries in the plastic battery case and run the power cord through the bigger square hole in the cardboard (Figure **E**).

Attach the white power connector to the battery port on the back of the board (Figure **F**).

Finally, close the cardboard tab, put the micro:bit on top, and tuck the two smaller tabs around the micro:bit buttons to enclose the micro:bit (Figure **G**). You can wrap the battery wires around the battery pack, behind the board, to make them less likely to snag on something. You now have your magic wand!

PROJECT 2: THE FAMILIAR

Let's move on to design a familiar — an interactive companion creature that will react to programmed inputs. For example, a dog might wag its tail in response to being moved in one of the ways that the micro:bit recognizes, or a turtle might pull its head in and out in response to the micro:bit being touched on its logo.

A servomotor (Figure H) is used to move an attached “horn” to a specified angle. The micro:bit's power and ground connections can support either lighting an LED strip or moving a servo, but not both at once. (In principle, you could attach both if there was separate power available for the servo, but we don't recommend trying it without breakout boards.)

DESIGNING YOUR FAMILIAR

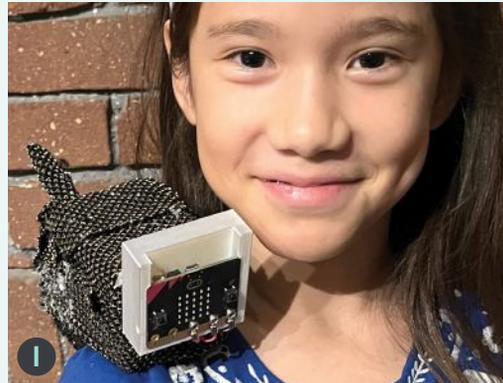
First, decide how the familiar works and possibly how it communicates with its fellows. You might look up animals used as heraldic symbols and select one that personifies your values (see for example www.heraldryandcrests.com/pages/heraldic-symbolism-a-z). Think through whether a single servo can create convincing motion for your chosen creature.

Then, decide whether you want the micro:bit screen to stay visible. If it will be covered up, that will limit the possible interactions with it. Remember to keep the USB connector accessible so you can update the code as needed. And if you're going to wear your familiar (on your shoulder, as a hat, on a wristband, etc.), planning and prototyping of the attachment is best done early in the process.

You'll need to plan how to support the servo to keep its orientation relative to the micro:bit screen. The simplest way to create a familiar is to put a servo in a box (like the one the micro:bit came in) and to allow the servo horn to poke out in a strategic direction. You could have the board visible as a “face” on the front of a box, and have the servo point backward to wag a tail (Figure I).

PLANNING MOTION

It's critical to figure out the motion the servo will be creating before starting to fabricate anything. A back-and-forth, wagging type motion is straightforward with a servo, but in-and-out



motions might require careful placement or a mechanical linkage of some kind. Hobby servos can only rotate half a circle and are designed to move to a designated position and stay there, not to rotate continuously. Sketch the motion first, particularly with respect to the relative orientations of the front of the micro:bit board and the servo horn.

CODING A SERVO

To program the servo, you'll need to add the Servo extension to MakeCode. Click on Extensions, search for “servo,” then click to add the servo code blocks to your Toolbox. Figure J shows the blocks that start the servo at a

BOARD TRICKS Micro:bit Magic

90-degree angle and then move it to 45 degrees when the board is shaken.

ATTACHING A SERVO

We can make a very simple “cat” with just alligator clips and a servo. First, we will need to wire up the servo, connecting it to the micro:bit’s pads. Make sure you’ve thought through the relative positions of your micro:bit and servo and that the wires you have are long enough.

Servos come with three wires attached. For our servo, the brown wire is ground, red is power, and orange is signal. (There are some exceptions to this convention, so check your servo’s packaging or documentation if the wires don’t appear to be brown/red/orange.)

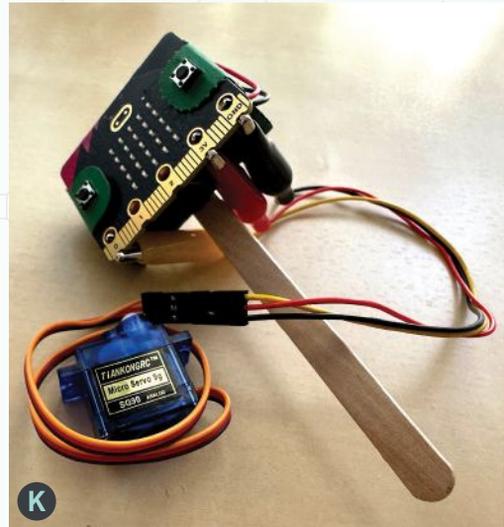
Connect alligator clips to the micro:bit as shown in Figure **K**. Then connect these wires to the servo: GND to the brown servo wire, 3V to the red servo wire, and pin 0 to the orange servo wire.

Now that the servo is connected, you can attach it to the micro:bit and battery pack. If you want to make a dog with the micro:bit LED display as its face and a wagging tail at the back, you could tape the servo to the bottom of the cardboard battery enclosure. (You can also use alternative connectors, which we discuss in our more-detailed lesson plans at nonscriptum.com/projects.)

Once the electronics configuration is finalized, finish your familiar with fabric scraps and craft materials. The cardboard box the micro:bit came in is a good size to support the board and servo, but not very conducive to seeing the LED screen. You may want to keep the screen exposed so that messages can be spelled out when the familiar is assisting with a “spell” (e.g. “Be gone!”). It is best to keep a slit or other access for the USB connector so that programming can be adjusted. Finally, consider how to change out the batteries without ripping the creature apart.

FAMILIARS AND ROLE PLAY

The familiars can be programmed to move, or to light up if you prefer to add LED strips instead of a servo. They can react to being moved, to buttons being pushed, or other inputs. The familiars in Figures I (previous page), **L**, **M**, **N**, and **O** (a cat, turtle, mouse, cat with sword, and



snake, respectively) were made by our students at Institute for Educational Advancement in Pasadena, using craft materials and some 3D printed shapes.

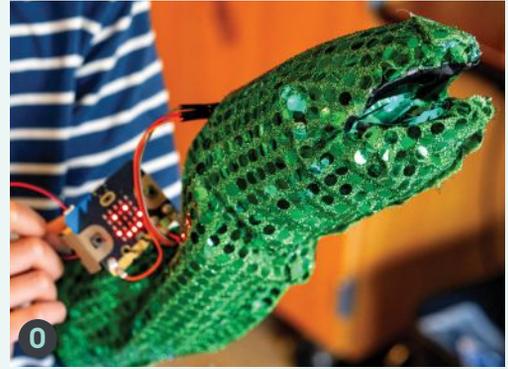
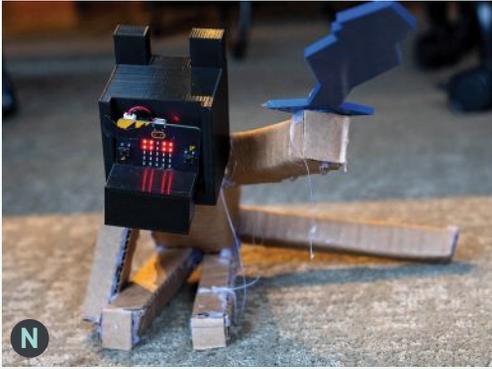
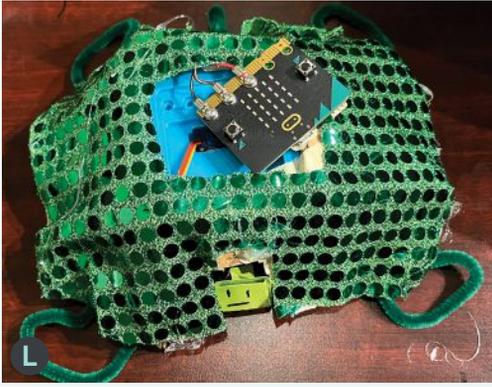
Familiars from our Anywear Academy colleagues at UC Santa Cruz included lights in some cases and servos in others (Figures **P** through **S**).

It can be challenging to attach anything to a servo horn. We experimented with paperclip wire push-rod linkages, but these proved a little too delicate. A pipe cleaner directly attached to the servo horn is a pretty effective tail.

If you’re part of group creating several familiars, consider working up a storyline where they need to manage communication between their wizards, or speak in “codes.”

GOING FURTHER

Whether or not you use your wands and familiars to create a magical world, we hope this pair of tutorials gives you confidence playing with micro:bits. From here, you might consider moving on to Python or JavaScript for coding micro:bits, or exploring one of the many other boards described in this issue. 🗝

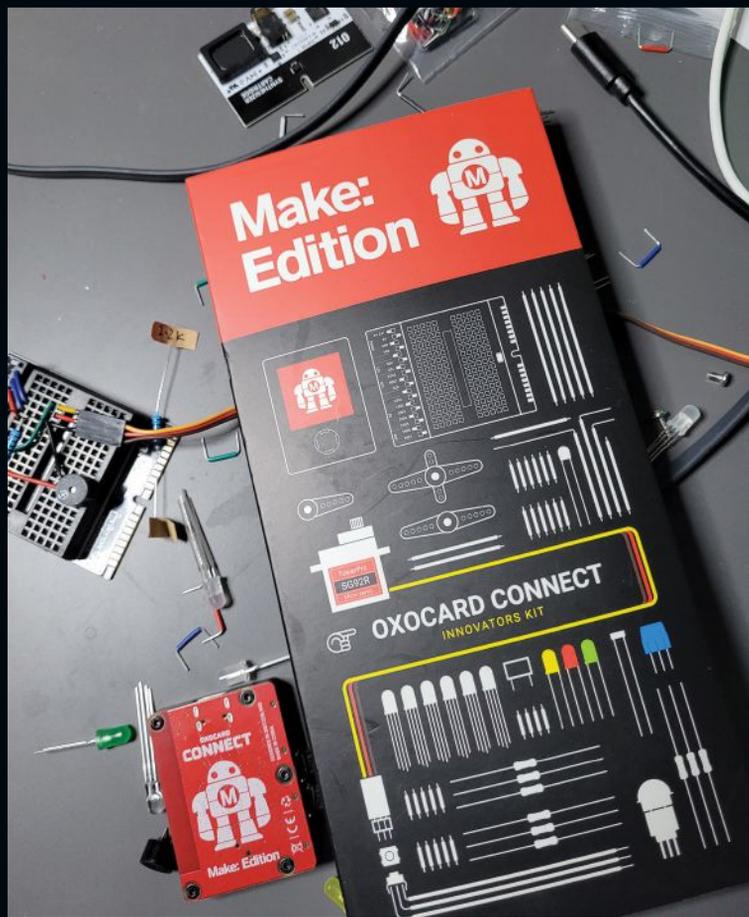


Joan Horwath/Rich Cameron, James Fey/Raquel Robinson/Selin Ovali

INNOVATORS KIT

MAKE: EDITION

Written by Daniel Bachfeld with Keith Hammond



DANIEL BACHFELD
is editor-in-chief of
Make: in Germany.



KEITH HAMMOND
is editor-in-chief of
Make: in the USA.

OUR SWEET NEW KIT MAKES LEARNING
PROGRAMMING AND ELECTRONICS QUICK,
EASY, AND FUN — WITH READY-MADE CODE
AND PLUG-IN CARTRIDGES

Want to build electronic gadgets fast and learn programming too? With the Oxocard Connect mini computer and cartridges, you can easily try out your own ideas, build prototypes for projects, or just play with ready-made gadgets and learn to code in a playful way. Together with our Innovators Kit you can take your first steps into the world of electronic components and learn how to combine them with a microcontroller to make all kinds of smart projects.

In this short project we'll introduce you to programming the Oxocard Connect using the scripting language NanoPy and a full-fledged, browser-based programming platform (IDE) to quickly build an IoT thermostat with alarm and servomotor. You'll connect the temperature sensor, alarm buzzer, and servo on a mini breadboard cartridge that plugs right into the Oxocard Connect's expansion port, no soldering necessary!

ABOUT OXOCARD CONNECT

The Oxocard Connect (Figure A) is a mini computer with a display (240×240 pixels), joystick, USB-C connection, network functions, and an expansion port where you can plug in ready-made cartridges or circuits of your own. The core of the Oxocard is the popular ESP32 microcontroller with 2MB PSRAM, 8MB Flash, Wi-Fi, and Bluetooth. The NanoPy language, based on MicroPython, is super easy to learn.

Thanks to the built-in Wi-Fi, and functions for HTTP and MQTT in NanoPy, you can not only run your projects locally but also send and collect data via the internet, to use your Oxocard for smart home projects or all kinds of IoT/cloud integrations.

USE IT

Using Oxocard right out of the box is easy — no software installation, no typing out code listings and troubleshooting them! Instead you can start using the cartridges right away with the built-in examples, and then experiment with the code as you wish. “Use, Modify, Create” is Oxocard's approach that quickly leads to initial successes and makes you want more. Sure, Oxocard and NanoPy together form a “learning system” — but it's one that's really fun and encourages curiosity.

TIME REQUIRED: 1 Hour

DIFFICULTY: Easy

COST: \$90

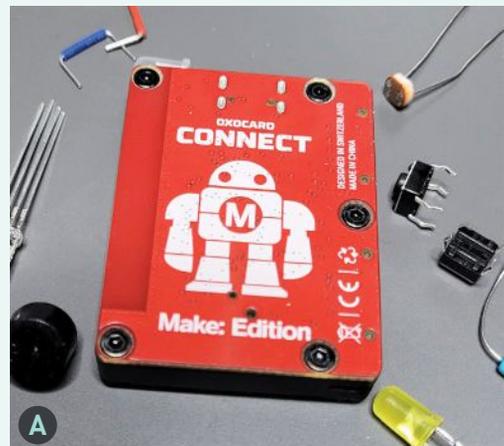
MATERIALS:

» **Oxocard Connect Innovators Kit, Make: Edition** from makershed.com/oxocard. Kit includes:

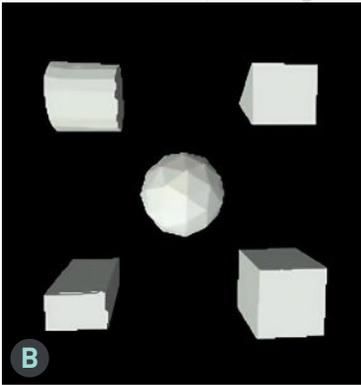
- **Oxocard Connect mini computer** with 240×240 display, joystick, USB-C port, expansion port, and ESP32-S3 microcontroller with 2MB PSRAM, 8MB flash, Wi-Fi, and Bluetooth
- **Breadboard cartridge** with built-in mini breadboard, breakout of ESP32 pins, and LEDs on I/O pins
- **Electronic components:**
Boldfaced items are used in this project.
 - PIR sensor (motion detector)
 - **NTC thermistor, 10kΩ** (temperature sensor)
 - Photoresistor, 10kΩ (light sensor)
 - Potentiometer
 - **Micro servomotor, SG92R**
 - **Piezo buzzer** (acoustic signals)
 - LEDs (3) green, yellow, red
 - RGB LEDs, WS2812B serial (5)
 - Switches, momentary pushbutton (2)
 - **Resistors (9)**
 - Pin header
 - **Jumper wires** various colors / lengths

TOOLS:

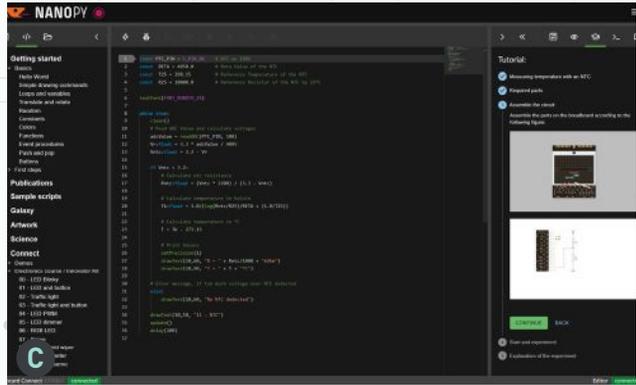
- » **Computer with internet connection**
- » **USB power source**



The Oxocard Connect (Make: Edition) can be recognized by the Makey on the back (and also on the boot screen).



The NanoPy command set also includes functions for 3D graphics.



In the online IDE you can try out examples, study tutorials and assembly instructions, and look up commands in the documentation.

In the programming environment there's a tutorial for each example project. These examples build on each other, and explain the special features of the NanoPy script. There's also a command reference you can search.

NanoPy is based on the commands and structure of MicroPython, but its graphics functions are based on the Processing language, which is popular with beginners (and familiar to many makers from Arduino). This allows you to conjure up graphics on the display with just a few instructions (Figure B). In the simplest example, just two commands:

```
drawRectangle(100, 100, 20, 20)  
update()
```

are sufficient to display on the Oxocard a rectangle at position $x=100$, $y=100$ and edge length 20.

MODIFY IT

The Oxocard comes with a pre-installed operating system that connects via Wi-Fi to the browser-based IDE on nanopy.io and interprets the scripts (Figure C). So you're programming via Wi-Fi instead of USB like Arduino; but if you want, you can still program your Oxocard using a cable. This system works with Chrome and Edge, not Firefox-based browsers. The NanoPy editor is also offered for download on GitHub so you can install it locally, but you'll still need to go online for convenience functions such as reference, tutorials, and documentation.

At the push of a button, the IDE sends one of the many example programs to the Oxocard,

which starts it locally on the ESP32. The Oxocard in turn sends data back to the IDE via Wi-Fi when the script is running, such as the output of the **print** command, which can be displayed using the Terminal window — practical when troubleshooting your own programs or modified examples.

In the IDE, values in the script or program can be modified during runtime using graphical sliders, and the effects on the Oxocard can be observed without having to go through the tedious process of recompiling and uploading like with Arduino!

There's also a debugging function that allows you to run your scripts line by line and observe what values the variables take.

WHY NANOPY?

Makers familiar with Python on computers, or MicroPython and CircuitPython on microcontrollers, will feel right at home learning NanoPy, but it's fundamentally different underneath. NanoPy creator Thomas Garaio explains:

"NanoPy is an entry level scripting language. It's made to look as similar to Python as possible, with "syntactic sugar," but it's not Python — it's not dynamically typed, it's statically typed. You don't need drivers or libraries, you just connect to the internet. It's easy to use but it's also fast operationally!"

Learn more at nanopy.io/en

CREATE WITH IT

By experimenting, you quickly get a feeling for the different commands and how they work. The self-explanatory function names also help with this. The many script examples can be easily combined, and there's a separate section in the editor for your own scripts with the option to save them on the Oxocard. You can then call them up in the Start menu under My Scripts.

The integrated sprite editor is also practical — you can draw up to 24×24 sprites with pixel precision and transfer them to your program as an array with a click. In combination with the joystick, you can create games and graphic applications.

PLUG-IN CARTRIDGES

The Oxocard Connect can be connected to additional hardware via its expansion port, which opens up huge potential. In addition to cartridges with sound synthesizers, air quality sensors, and an LED pixel matrix, there's the breadboard cartridge for electronics experiments (Figure D). It's even got LEDs on each I/O pin! Bundled with around 30 electronic components, jumper wires, and of course the Oxocard Connect, *Make:* magazine offers this as the Oxocard Innovators Kit — *Make:* Edition (makershed.com/oxocard).



With the debugger you can examine your program at runtime and check the contents of the variables.



A sprite editor is available for graphics up to 24×24 pixels.

The kit includes a temperature-dependent resistor (NTC) for temperature measurements, a light-dependent resistor (LDR) for measuring brightness, a PIR sensor as a motion detector, as well as a piezo buzzer, servomotor, and various LEDs, buttons, and resistors. You can build all sorts of things with it.

As an example, let's build a smart thermostat: It should show the current temperature on the display, move a servo accordingly, sound an alarm when values exceed 30°C (86°F), and also send the temperature to your Wi-Fi network via an MQTT message.

The online IDE contains a tutorial with code, circuit diagrams, and construction suggestions for each electronic component, and for MQTT, so it's easy to build up this project step by step!



Cartridges for the Oxocard Connect include the Breadboard, PixelMatrix, and Synthesizer shown here, plus Air cartridge with three gas sensors, ToF with 8×8 distance sensor, Veroboard, and more.

BUILD AN IOT THERMOSTAT

First we'll build the thermometer with the NTC thermistor. This temperature-dependent resistor, together with a 2.2kΩ resistor, creates a voltage divider we can measure with the Oxocard's analog-digital converter (ADC) via pin IN06 on the breadboard cartridge (Figure E). From this voltage, we can calculate the temperature using the characteristic values for our NTC. Our program is essentially:

```
while true: clear()
    adcValue = readADC(IN06, 100)
    T = calculateTfromA(adcValue)
    drawText(10,90, "T = " + T + "°C")
    update()
    delay(1000)
```

In an endless loop, the program clears the display with `clear()`, then reads the voltage value from the ADC. We see the first parameter of this function is the pin name, but what's the second? In the NanoPy IDE, enter "readADC" in the search box to learn what the function does: "Reads the analog value ... several times (`nSamples`) and calculates an average value." Okay, 100 samples! This averaging function is used to filter noisy or fluctuating measurements.

We calculate the temperature `T` using our NTC's conversion formulas, represented here by the dummy function `calculateTfromA()`. With `drawText()` we write the value `T` to the x-y coordinates 10 and 90 and append the °C unit as a text string. However, only the `update()` command actually draws on the display. All text, figures, and pixels of the `draw` commands are

first drawn internally in memory. Finally, `delay()` waits 1,000ms (1 second) before the loop is run through again. Click the Run Code button to send the program to the Oxocard Connect, and the current temperature in °C appears on the display.

MAKE IT MOVE

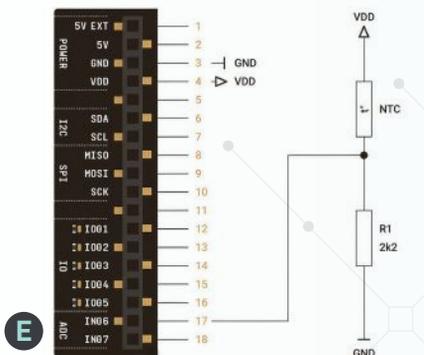
Measuring temperatures is, along with making LEDs blink, a favorite first project for makers — but it's a bit boring. Let's make it do something!

If you add a servomotor (Figure F), you can physically react to the measured temperature, for example by moving a valve or knob, or the slats of a window blind.

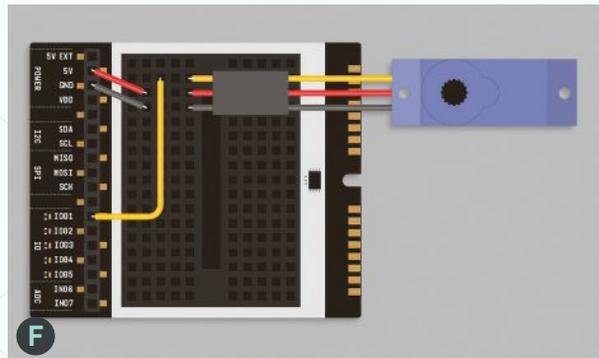
```
Three lines are needed to control the servo:
setPWMFrequency(50)
dutyCycle = map(A, 90, -90, 102, 512)
writePWM(I001, dutyCycle)
```

The first command sets the PWM frequency to the usual 50Hz for analog servos. The `dutyCycle` variable controls the servo angle, using a `map()` function to calculate the correct pulse width ratio, from 102 for 90 degrees to 512 for -90 degrees. (These values are derived from the Oxocard's internal PWM frequency generation. Fortunately, everything's already specified for you!)

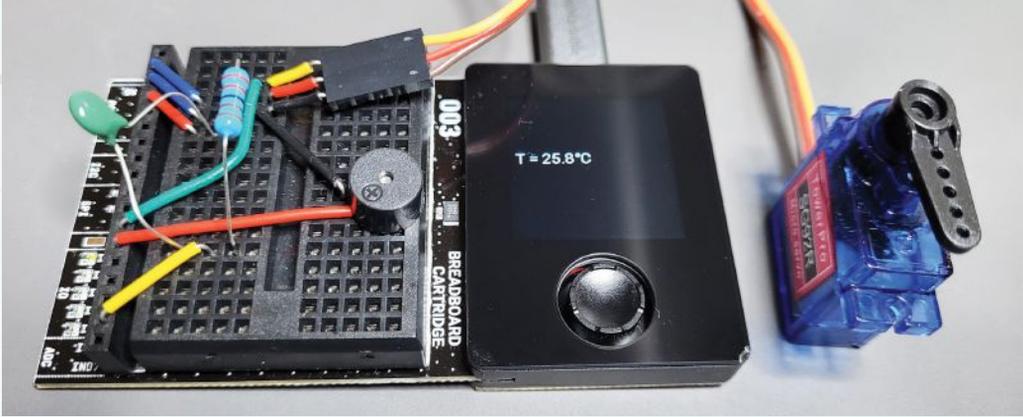
Pass the result to our output pin I001 using `writePWM()` and the servo is now set up. However, in order to move the servo linearly to the temperature, an additional call to `map()` is required, before the `map()` we just explained: `A = map(T, 40, 10, -90, 90)`
`dutyCycle = map(A, 90, -90, 102, 512)`



The circuit schematic shown in the online tutorials helps you wire the simple components.



If you're less familiar with circuit schematics, you can follow the assembly instruction diagrams à la Fritzing.



Wired up: Our simple digital thermostat with alarm and servomotor.

This allows us to map our expected indoor temperatures of a maximum 40°C (104°F) to one full swing of the servo and a minimum 10°C (50°F) to the opposite swing. At 25°C (77°F) the servo should be approximately in the middle.

ALARM AND REMOTE CONTROL

It would be nice to have an alarm sound when it gets too warm, for example above 30°C (86°F). Connect the piezo buzzer to pin I002, and control it with a pulse width signal with a ratio of 50%:

```
if T > 30
  writePWM(I002, 4096/2)
else writePWM(I002, 0)
```

We've already set the Oxocard's PWM frequency to 50Hz for the servo, so we'll have to use 50Hz again. As soon as the temperature falls below 30°C, the buzzer stops sounding.

It would be really nice to be able to control the temperature remotely, for example by having the Oxocard send the values via MQTT. Here too, the documentation will help you with a short introduction and a code example. You can publish the temperature via your Wi-Fi network like this: **uri = "mqtt://ip-address"** **connectMQTT(uri, username, password)** **publishMQTT("Temperature", T)**

The two commands **connectMQTT()** and **publishMQTT()** are basically sufficient to connect to an MQTT server and send your own values via the **Temperature** topic.

Finally, we save the script on the EEPROM of the breadboard cartridge and configure the autostart. Now the Oxocard will execute our program as soon as the cartridge is inserted.

```
1  setPWMFrequency(50)
2  textFont(FONT_ROBOTO_24)
3  uri = "mqtt://192.168.2.120"
4  connectMQTT(uri, "", "")
5  while true:
6    clear()
7    adcValue = readADC(C_PIN_06, 100)
8    Vr:float = 3.3 * adcValue / 4095
9    Vntc:float = 3.3 - Vr
10   if Vntc < 3.2:
11     Rntc:float = (Vntc * 1629) / (3.3 - Vntc)
12     Tk:float = 1.0 / (log(Rntc/10000.0)/4050 + (1.0/298.15))
13     T = Tk - 273.15
14     setPrecision(1)
15     drawText(10, 90, "T = " + T + "°C")
16   else:
17     drawText(10, 60, "No NTC detected")
18     ANGLE = map(T, 40, 10, -90, 90)
19     dutyCycle = map(ANGLE, 90, -90, 102, 512)
20   if T > 30
21     writePWM(C_PIN_02, 4096/2)
22     delay(500)
23   else writePWM(C_PIN_02, 0)
24   writePWM(C_PIN_01, dutyCycle)
25   publishMQTT("Temperature", T)
26   update()
27   delay(1000)
```

This is the finished script for the thermostat; you can download it from github.com/MakeMagazinDE.

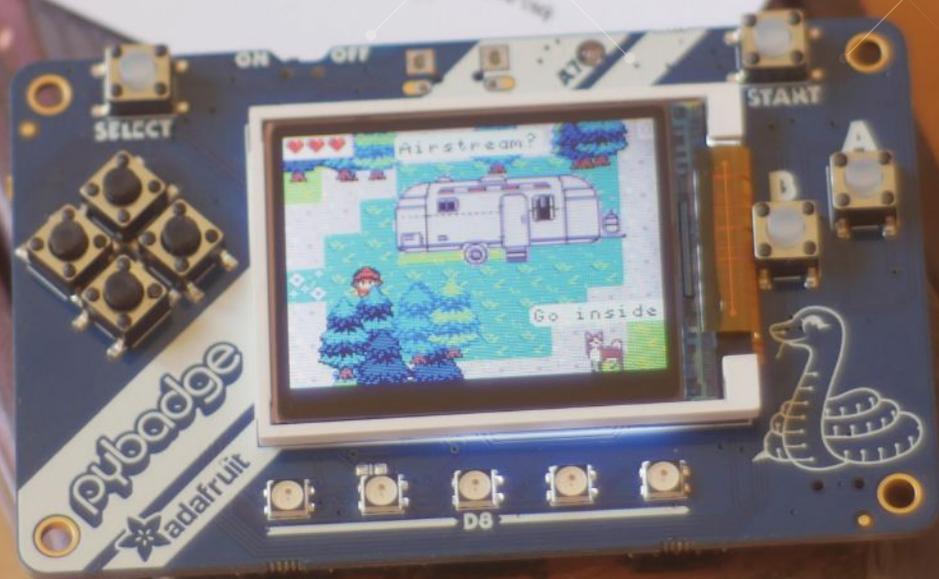
MORE TO COME!

Use, Modify, Create! That's exactly the approach we used to build our own smart gadget with the Innovators Kit and, based on the existing examples, get it up and running quickly.

Make: is now developing supplemental materials around the *Make: Edition Innovators Kit*, and we'll be using the Oxocard Connect in future projects in the magazine, so please send us your ideas for projects — and cartridges! 🗨️



Learn more about Oxocard Connect and get your Innovators Kit at makershed.com/oxocard.



Written by Mo Rahman

BADGE OF LOVE

CREATING A VIDEO GAME
WEDDING INVITE WITH
ADAFRUIT'S BADGE BOARDS



MORAHMAN loves to tinker and can be found from bars to national parks, between Silver Lake and SF, diving into bioinformatics software, open-source hardware, and distortion pedals. petridishtalk.com

Twenty years ago, my spouse-to-be Pascale and I met and became friends. During the pandemic, we got engaged and wanted to create a unique and memorable wedding invitation for our friends and family.

BADGE/LIFE

Around 2015 we found ourselves, along with a group of friends, at a punk show in San Francisco. After the show, we stepped out into the night and noticed some people unloading microscopes from the back of a car. In my somewhat buzzed state, I realized we were right in front of Noisebridge (noisebridge.net), a wild hacker and maker space in the sometimes grungy Mission District.

We approached and asked if they were heading up to the makerspace and if we could join them. They were incredibly friendly and invited us along. Once inside, Pascale and I were immediately captivated by the scene. DIY robots were rolling around, people were engrossed in their own projects, and a few members came over to greet us with offers of tea.

When it came time to plan our wedding, we knew we wanted to capture that same spirit of discovery and creativity. By creating a video game wedding invitation, we aimed to offer our loved ones a unique glimpse into our world — combining our love for technology, gaming, and the community spirit that has always been a part of our journey together.

Inspired by the ingenious DEF CON badges (defcon.org/html/links/dc-badge.html), known for their custom and interactive designs, we dreamt of creating something equally unique and complex. However, the practicalities of planning a wedding soon tempered that ambition. Our goal was to have a joyful and memorable celebration with good vibes, not to be bogged down by the stress of overly elaborate projects. The idea of sending everyone a little dev board as a wedding invite was exciting, but how much time, money, and mental energy could we realistically afford to dedicate to this project?

For those inspired to craft their own unique video game invitations or projects, our journey and design may serve as a valuable guide. The techniques we used can be adapted to weddings, conferences, or even birthday cards.

1. CHOOSE THE RIGHT HARDWARE

Selecting a development board that fits your budget and technical requirements is crucial. In the end, we chose the Adafruit EdgeBadge (Adafruit 4400) and PyBadge LC (3939) for their impressive features and relative affordability. These credit card-sized boards include a 1.8" 160x128 color TFT display, eight game/control buttons, a triple-axis accelerometer, light sensor, mini-speaker, and mono Class-D speaker driver. There is a LiPo battery port with built-in recharging capability, for which we used a mix of 3.7V 400mAh and 3.7V 420mAh batteries (Adafruit 3898 and 4236). Finally, there's a USB port for charging, programming, and debugging, and two female header strips in Adafruit's Feather format, for potential expansions. You can program both of them in MakeCode Arcade, Arduino, and CircuitPython; the EdgeBadge additionally supports TensorFlow Lite for Microcontrollers (TFLM), enabling machine learning features such as voice commands and gesture controls.

Our game design was inspired by our pandemic camping trips around California and Nevada and our discovery of AutoCamp, a campground featuring custom Airstream trailers and canvas tents near national parks. Because I'm of North Indian Mughal descent, our wedding involved multiple days of different rituals, reflecting a rich cultural heritage where Hindu and Islamic traditions blend. These rituals can get very messy, so it would be nice to have a venue we wouldn't completely ruin with our festivities.

We thought these outdoor settings would be perfect for our wedding and our game badge. One of the wedding rituals is the vibrant Haldi ceremony, where an entire day is dedicated to covering ourselves in bright yellow turmeric. This tradition is deeply symbolic, and is reminiscent





of the one depicted in the Mughal painting *The Emperor Jahangir Celebrating the Festival of Holi*. Another day was dedicated to having intricate patterns painted on our bodies with henna, showcasing fractal-like paisley designs in traditional motifs. However, henna can also leave lots of stains on furniture and fabrics.

Initially, we ambitiously designed an open-world game where guests could explore the real map of our wedding venue, and interact with real-world elements at the campsite, leveraging the many features of the Adafruit boards. However, while it ran smoothly on a MacBook, it performed terribly on the dev boards, causing some panic.

2. UTILIZE ONLINE RESOURCES

Take advantage of online tutorials, forums, and documentation, such as adafruit.com/explore/makecode-arcade#group-58. Our coding process involved significant trial and error, as our game's abysmal performance on the dev boards prompted a complete overhaul of our coding strategy. YouTube videos became crucial resources in overcoming these challenges, particularly "Microsoft MakeCode Arcade JavaScript | COM227A" from the official Microsoft developer channel (youtu.be/35LtUBHtCYI). This

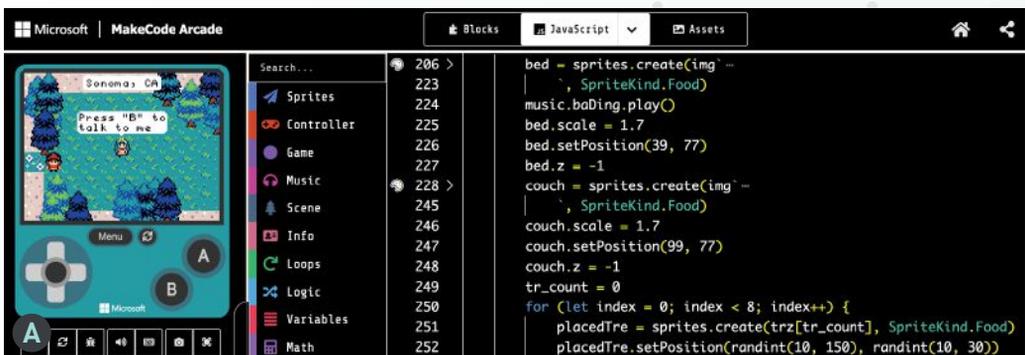
video provided insights into defining levels and efficiently using assets in JavaScript (Figure A). By adopting strategies from these resources, we shifted from an open-world design to a more efficient, level-based approach.

3. OPTIMIZE FOR PERFORMANCE

If your initial design doesn't run well, focus on efficient asset management and consider breaking your game into smaller, more manageable levels that can be quickly loaded and unloaded (Figure B), like classic games in the Game Boy Advance era (forum.makecode.com/t/creating-levels-and-end-boss). By reusing sprites and minimizing memory usage, we maintained the spirit of exploration within the campground while ensuring the game ran smoothly on the Adafruit EdgeBadge and PyBadge LC. The final design was reminiscent of a minimal RPG, with a feel similar to the mid-90s Pokémon games.

4. INCORPORATE PERSONAL ELEMENTS

To make the game feel unique and special, use sprite generators and editors to create characters and elements that reflect your event's theme. We used sprites to represent various elements





of the campground, including Airstreams, tents, and characters (Figures **C** and **D**). Artwork that we commissioned from Storyboard artist Jake Lawrence ([timecowboy.tumblr.com](https://www.tumblr.com/timecowboy)) for the invite box and wedding website was also converted into sprites to maintain a cohesive design. For this task we used an image-to-sprite converter (github.com/kristianpedersen/Convert-Image-to-MakeCode-Arcade-Sprite) and MakeCode Arcade's built-in sprite editor. Important sprites were carefully crafted to ensure they not only looked good but also ran efficiently on the low-power devices. Community forums and documentation from both Microsoft and Adafruit provided solutions and best practices that significantly improved performance.

5. MUSIC AND SOUND

Adding a personal music track can enhance the immersive experience and add a memorable, emotional touch. This was a really enjoyable aspect of the project. For the end of the game (Figure **E**), I created a MIDI cover of a Depeche Mode track with the lyrics: *Come with me into the trees / We'll lay on the grass and let the hours pass / Take my hand, come back to the land / Let's get away, just for one day.*



6. TEST IT OUT

Test your game extensively on the target hardware and iterate based on feedback to ensure a smooth and enjoyable experience. By following these steps, you can create a memorable and interactive invitation that will delight your audience. You can play our finished game at MakeCode Arcade, and check out our code, in your browser (arcade.makecode.com/S14843-80440-75818-95334).

This project was a refreshing change from my bioinformatics work, and allowed me to have fun with engineering in a creative way. Using dev boards and video games while still performing centuries-old rituals is part of a harmonious blend, showing how diverse perspectives on history, societies, and relationships coexist.

THE GIFT THAT KEEPS ON GIVING

We highly recommend using these kinds of low-cost dev boards as gifts to friends and family. They can enjoy the game you made, or create their own projects, and it's fun to see what people end up making. Combining retro gaming charm with cutting-edge tech and cohesive design elements makes for an unforgettable and fun invitation. Try it out and make your invite the talk of the town! 🎮





Batteries With BULBUL

A BIRD'S EYE VIEW OF
BATTERY CHEMISTRIES
AND APPLICATIONS

Written and illustrated by Bulbul Yousif



Adobe Stock-Alex

HI! Bulbul here! There are a lot of ways to store energy out there — mechanical, thermal, chemical, gravity wells, flywheels, moving water around — but most of us in the maker world are very familiar with these little metal containers called batteries! Which one do you want to use for your project? What should you design for, and with? Ultimately that's up to you, but it's my hope that I can give you a nice

overview of some of the most accessible types of battery-based (with a couple of exceptions) storage.

First of all, batteries are generally grouped into two main categories: **Primary batteries** are single-use, aka non-rechargeable. **Secondary batteries** are rechargeable. Both kinds come in a variety of form factors, chiefly **cylindrical**, **prismatic** (rectangular), and **pouch** types.



BULBUL YOUSIF is a creature of many names, currently located in the Front Range of Colorado. They fell into electronics and have been trying to claw their way out by building as many cool things as they can. chippydootles.com

Batteries also come in all sorts of **chemistries**. The chemical makeup of a battery is what determines its characteristics, such as voltage, energy density, overall weight, and recyclability. I won't go into specifics about chemical reactions but here are some common, some uncommon, and some up-and-coming types of batteries.



PRIMARY ZINC-CARBON

Zinc-carbon were one of the earliest forms of batteries and are still produced today. They're reliable but work better in low-drain and intermittent applications. They also have a short shelf life due to their chemistries, and were historically prone to leakage.

- **Nominal Cell Voltage:** 1.5V
- **Discharged Voltage:** 0.9V–1.05V
- **Common Cell Types:** Standard sizes such as AA (R6), AAA (R03), D, C, 9V (PP3)
- **Applications:** Low current and intermittent usage such as remote controls for TVs

ALKALINE

Alkaline batteries are probably the first kind you think of when it comes to AA, 9V, and other common sizes. They make up the majority of standard batteries that you can find off the shelf at stores. These grew more popular than zinc-carbon as they had much better performance and shelf life. They also support higher-drain applications.

- **Nominal Cell Voltage:** 1.5V
- **Discharged Voltage:** 0.9V–1.05V
- **Common Cell Types:** Standard AA (LR6), AAA (LR03), D, C, 9V (PP3); button LR44, LR626, etc.
- **Applications:** Toys, remotes, older cameras, portable radios, clocks



LITHIUM METAL (COIN CELLS)

Lithium metal batteries come in a variety of chemistries and voltages. These are most often seen as “coin” or “button” cells. The use of lithium has grown a lot due to its energy density and higher voltage than most batteries. These were also seen commonly as CR123A batteries, often used in cameras for flash applications.

- **Nominal Cell Voltage:** 3.0V
- **Discharged Voltage:** ~2.0V
- **Common Cell Types:** Coin cell such as CR2032, CR123A camera batteries
- **Applications:** Real-time clock (RTC) modules, small IR remotes, camera flashes

LITHIUM IRON DISULFIDE (Li-FeS₂)

This is a type of non-rechargeable lithium metal battery that you may have seen pop up more recently. Their chemistry provides a nominal voltage very close to 1.5V making them useful as drop-in replacements for alkaline, zinc-carbon, and NiMH batteries. They offer much more capacity than alkaline batteries, a lower self-discharge rate, and somewhat lighter weight.

- **Nominal Cell Voltage:** 1.6V/1.5V
- **Discharged Voltage:** 0.9V
- **Common Cell Types:** Standard cylindrical such as AA (FR6)
- **Applications:** High drain

SECONDARY

LEAD-ACID

Lead-acid is one of the oldest battery types, and is highly prevalent in our automobile-based society. The toxicity of lead and its high weight-to-energy ratio make these not a very attractive choice. They are, however, the most recycled battery currently, though the environmental and health costs of working with lead are still very high.

- **Nominal Cell Voltage:** 2.0V
- **Charge Voltage:** 2.23V–2.32V
- **Discharge Cutoff:** 1.75V
- **Cycle Life:** 50–300
- **Common Cell Types:** Car batteries or smaller rectangular batteries
- **Applications:** Automotive, motorcycle, backup, emergency power, solar power storage

NICKEL-CADMIUM (NI-CD, NICAD)

Nickel-cadmium batteries were one of the earlier common rechargeable batteries as their nominal voltage was very close to that of alkaline/zinc primaries, albeit slightly lower. Ni-Cads are very good in high-drain, high-current applications and resistant to deep discharge, which is uncommon in rechargeables. Some drawbacks of Ni-Cad chemistry are its vulnerability to “memory effect” and cadmium being a toxic heavy metal. It was used commonly in battery-powered tools but has been largely phased out in favor of Li-ion.

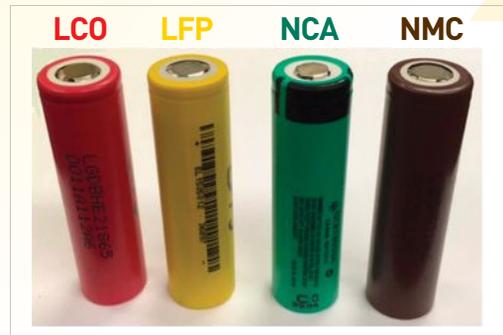
- **Nominal Cell Voltage:** 1.2V
- **Charge Voltage:** 1.3V
- **Discharge Cutoff:** 0.9V
- **Cycle Life:** 500
- **Common Cell Types:** Standard cylindrical such as AA (KR6), C
- **Applications:** High drain, power tools

NICKEL-METAL HYDRIDE (NIMH)

NiMH batteries have supplanted Ni-Cad, as their voltage makes them a drop-in replacement. NiMH usually have more capacity and don't suffer from the memory effect, but they don't support deep discharge like Ni-Cad, and their high self-discharge rate means they'll discharge faster when not connected to a load. If that's a concern, look for special “low self-discharge”

(LSD) versions that have better performance. The hardest part of NiMH is probably charging: most schemes I've come across are based on current and temperature alone with a high voltage cutoff of 1.7V per cell. However, these are a good choice if you want readily accessible sizes, easily swappable batteries, and off-the-shelf chargers.

- **Nominal Cell Voltage:** 1.2V
- **Charge Voltage:** 1.4V (up to 1.7V fast charging)
- **Discharge Cutoff:** ~0.9V
- **Cycle Life:** 300–800, 500–1500 (LSD)
- **Common Cell Types:** Standard cylindrical such as AA (HR6), AAA (HR03), C, D
- **Applications:** High drain, and alkaline replacements in most situations

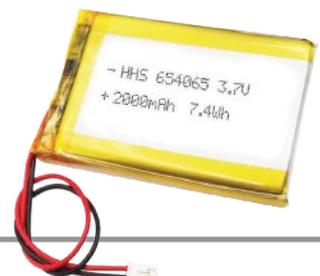


LITHIUM-ION (LI-ION, LI+, LIB)

Lithium-ion is the reigning champion of battery chemistries in modern times. Their usage really exploded along with the proliferation of laptops and smartphones. They come in a dozen flavors, typically named for the metals used in the cathode, such as LCO, LMO, NCA, and NMC. One reason they became the go-to is their low weight. They still suffer from drawbacks, notably low cycle life and a serious fire hazard. If these batteries aren't managed properly, the results could be, well ... a day-ruiner if you are lucky.

Lithium Polymer (LiPo, Li-poly)

LiPo are the pouch-style Li+ cells and you want to be very careful when handling them. Crushing, piercing, or bending them





can all lead to fires. Overcharging and poor cycle management on these batteries also leads to what many like to refer to as “danger pillows”: pouches will swell in size, often very visibly, which can cause physical damage via deformation of things like phones, battery packs, and laptops. A good way to avoid this is by setting lower charge thresholds if firmware allows it, especially if your device is always plugged in, or just manually managing things by not leaving them plugged in or not letting them reach full charge if there are no firmware controls.

- **Nominal Cell Voltage:** 3.6V, 3.7V
- **Charge Voltage:** 3.8V–4.2V
- **Discharge Cutoff:** ~3.0V
- **Cycle Life:** 500–1,000
- **Common Cell Types:** Standard cylindrical 18650, 14500 (AA), many custom size pouches
- **Applications:** Rechargeable portable devices, laptops, phones, remote sensors



Lithium Iron Phosphate (LiFePO₄, LFP)

Lithium iron phosphate batteries use fewer rare metals than other Li⁺ types. They have a lower energy density but are safer, more tolerant of higher current draws, and less prone to thermal

runaway. This makes them a great substitute for applications that might take more abuse than you’d be comfortable with for regular Li-ion cells. I prefer to use these for the added safety, albeit at the cost of energy density. These are becoming more common in solar harvesting applications such as solar yard lights (usually 14500 or 18500 sized), solar power banks, boats, and RVs.

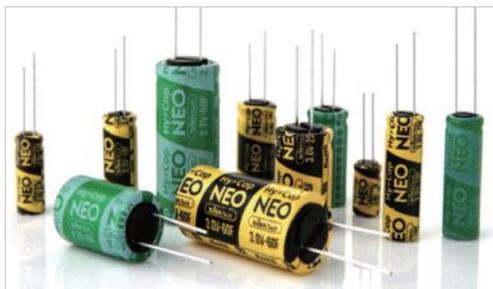
- **Nominal Cell Voltage:** 3.2V
- **Charge Voltage:** 3.6V
- **Discharge Cutoff:** ~2.0V (2.5V usually recommended)
- **Cycle Life:** 2,500–12,000
- **Common Cell Types:** Standard cylindrical such as 18650, 14500 (AA); prismatic for solar/vehicles
- **Applications:** Renewable energy storage, high power, EVs, RVs, robots



Lithium Titanate (LTO, Li-Ti)

LTO batteries are relatively new to me, but I’m really impressed by their capabilities. They have a much lower nominal voltage than Li⁺ or LFP but are the most resilient of the three. LTO has loads of advantages: they charge very fast and support higher discharge rates, are very stable and safe, and have a very high cycle life. The only technical disadvantage is the lower voltage range. If you need batteries to meet some intense power demands, these could be a great choice.

- **Nominal Cell Voltage:** ~2.3V
- **Charge Voltage:** ~2.8V
- **Discharge Cutoff:** 1.5V–1.7V
- **Cycle Life:** 6,000–20,000
- **Common Cell Types:** 18650, large cylindrical sizes like 66mm×160mm
- **Applications:** High power, renewable power, EVs, extreme temps, wristwatches, tablet pens



SODIUM ION (NA+, NA-ION, SIB)

Sodium ion batteries an exciting development that's been a long time coming. They're very similar to their lithium-ion counterparts but use sodium ions instead. The biggest advantage of this is sodium being leagues more abundant and lower-cost than lithium. Aqueous-electrolyte Na+ batteries are much safer than dry-cell Na+, which suffer from similar thermal runaway problems as Li+, but a great deal of research is being put into additives to increase their safety.

Na+ is one of those techs that is just on the horizon, but recently I've found them available through outlets such as AliExpress. Major Chinese battery makers are also producing them for the electric vehicle market, so I wouldn't be surprised if we see a lot more popping up. One of the downsides is that charging circuitry for this chemistry isn't as common as more established Li+, LFP, and NiMH, which means more discrete circuitry might be needed if you're an early adopter. They still charge with a CC/CV scheme, albeit with a different curve. Na+ also have a much wider voltage range of 4.1V charging, 3.0V nominal, and 1.5V cutoff.

- **Nominal Cell Voltage:** ~3.0V
- **Charge Voltage:** 4.1V
- **Discharge Cutoff:** 1.5V
- **Cycle Life:** Thousands
- **Common Cell Types:** Standard cylindrical such as 18650, 26700; prismatic packs
- **Applications:** Renewable power, EVs, Li+ replacements

SUPERCAPS and HYBRID CAPACITORS

While not batteries, if you plan your project around these, **supercapacitors** and **hybrid capacitors** could very well replace the need for a battery! Supercaps and hybrid caps are rated as capacitors and marketed for applications that need more power with reliable longevity.

SUPERCAPACITORS (EDLC)

Electrostatic double-layer capacitors act more like you'd expect from capacitors. They have a high self-discharge rate but can handle a lot of current over a long lifetime. They have a linear charge/discharge but are capable of charging in seconds to minutes, compared to hours for batteries. The big catch: They're much less energy-dense, so your run times will be much shorter. Charge circuitry is much less complex, as there is no worry about overcharging, so no need for a charge current cutoff. Base the charging current on what your circuit can safely handle.

- **Nominal Cell Voltage:** ~2.7V
- **Discharge Cutoff:** Can be run to 0V
- **Cycle Life:** Hundreds of thousands
- **Common Cell Types:** THT and SMD radial capacitor
- **Applications:** Renewable power, immediate power capture; some RTC applications

HYBRID CAPACITORS

Hybrid caps are a really interesting middle ground: they're both capacitors and batteries, while functionally being neither. They use both electrostatic and electrochemical methods of energy storage and transfer. Different manufacturers make different kinds; most commonly I see lithium-ion hybrid caps, but for

personal projects I've really dug into some hybrid caps from Vishay Intertechnology.

Lithium-Ion Capacitors (LiC)

LiCs typically use the same packaging as EDLCs so can be easily mistaken; the biggest tell is their rated voltage, usually 3.8V though I've seen some 4.0V (and heard of 4.2V). The capacitor nature of these lets them handle more power (current) than a battery of the same size, but with more energy than an EDLC of the same size. They have more limitations than capacitors, as they do have a discharge cutoff that needs to be respected for continued use. None of the datasheets I've read have mentioned managing overcharge, but with LiC's voltage max of 3.8 it may require some thought about how to charge them properly, as most battery charging circuits are aimed at 4.2V. You might consider under-spec'ing these for improved lifetime, and use a charger for LFP batteries which maxes out at 3.65V usually. Since overcharging doesn't seem to be an issue for these, you can also just provide them a fixed voltage and charge them that way.

- **Nominal Cell Voltage:** 3.8V
- **Discharge Cutoff:** 2.5V
- **Cycle Life:** Hundreds of thousands
- **Common Cell Types:** Varies by type
- **Applications:** Renewable power, immediate power capture; some RTC applications



Vishay 196 HVC Enycap Hybrid Caps

While Vishay are not very forthcoming about the chemistry, these are definitely hybrid caps. I've used them extensively in personal projects and haven't found anything else like them. Their rated voltage of 1.4V per cell, as well as their sizes and packaging, suggest an intended application of



NiMH replacements in backup situations. While their voltage characteristics are similar, they have a much stricter maximum of 1.4V (depending on temperature) that's incompatible with modern NiMH charging circuits that go as high as 1.7V, which could be damaging to them. Where these really impress is their rated energy for size and profile. A 12mm-diameter, 1.4V cell rated at 15 farads and an SMD height of 3.8mm really allows me to work with some low height and size tolerances. Unlike the LiCs I've come across, there's a lot more technical info and application notes available for these. They do need management similar to batteries in order to maximize longevity, such as undervoltage and overcharge protections. These are worth looking at if you want a replacement for coin cells, but they'll potentially need much more circuitry in your design for charging and voltage protection.

- **Nominal Cell Voltage:** 1.4V
- **Discharge Cutoff:** 0.8V
- **Cycle Life:** 50,000–100,000
- **Common Cell Types:** Various combinations of stacked cylindrical cells
- **Applications:** Emergency backup, energy harvesting, lighting

GET CHARGED UP

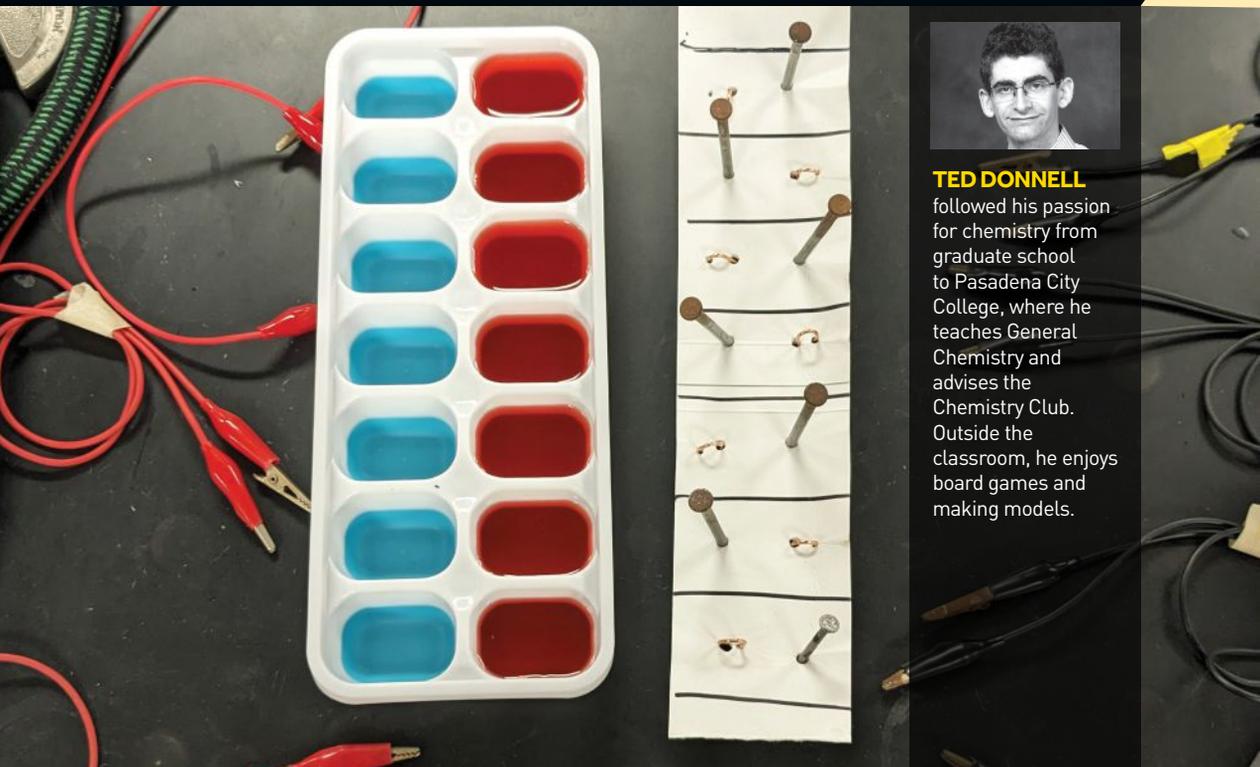
Thanks for sticking with me on this little overview! There are a lot of battery options out there that will work for a lot of different applications. Hopefully being aware of some of them will help make more of your projects come to life! 🚀



1+2+3 RECHARGE WITH ELECTROLYTES!

BEVERAGE BATTERY

Written and photographed by Ted Donnell



TED DONNELL

followed his passion for chemistry from graduate school to Pasadena City College, where he teaches General Chemistry and advises the Chemistry Club. Outside the classroom, he enjoys board games and making models.

Batteries all work under the same principles: a voltage difference between a pair of chemical reactions passes electrons out, through the wire, for use driving our devices. All batteries contain three components: two electrodes called the **anode** and **cathode**, and the **electrolyte**. To investigate these, we'll look at a simple battery

you can build at home using nails, a sports drink like Gatorade, and an ice cube tray — and we'll talk about how these materials compare to what you find in the lithium ion batteries that run our phones and computers.

This project was adapted from a workshop by American Chemical Society (ACS).

1. MAKE THE ELECTRODES

With a marker, divide the cardboard strip into boxes that line up with the sections of the ice cube tray. These will be the individual **cells** of the battery.

In each section, punch one zinc nail and one copper nail/copper wire through the cardboard, alternating as shown in Figure **A**. Make sure the nails are straight and are not touching each other on the other side of the cardboard. This is easiest to do by putting the cardboard over the ice cube tray and then punching through by pushing down. Here I have punched a second hole and bent the copper wire so that it stays put.

SAFETY: Be careful as you punch the nails through the cardboard.

2. FILL CELLS WITH ELECTROLYTE

Fill the ice cube tray with your chosen electrolyte (Gatorade or Powerade, any flavor).

Place your electrode strip over the ice cube tray, so that each pair of nails/wires dips into the electrolyte. Here we're using only the red sports drink; you can see the blue drink on the other side, so we can compare them later!

3. CONNECT THE CELLS

Use the alligator clips to connect the individual cells together. Connect copper nail 1 to zinc nail 2, and connect copper nail 2 to zinc nail 3, and so on (Figure **B**). Nails in the same cell *should not* be connected to each other by wire. That's what the electrolyte is for!

When you finish, one end of the battery should have a free zinc nail with no wire, and the other end should have a copper nail with no wire.

USE IT!

Connect the free zinc nail to the shorter leg of the LED, and the free copper nail to the longer leg (Figures **C** and **D**). You should see the LED light up as you harness the chemistry of your battery, amplified by the multiple connected cells!

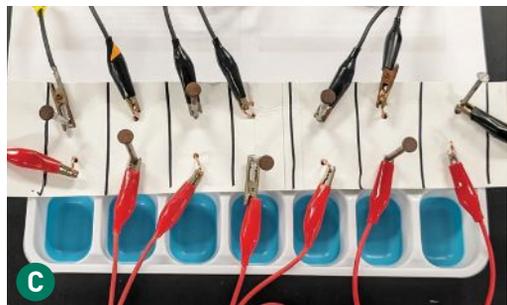
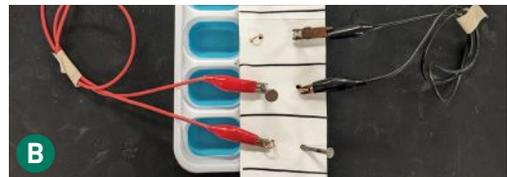
TIME REQUIRED: 20–30 Minutes

DIFFICULTY: Easy

COST: \$10–\$20

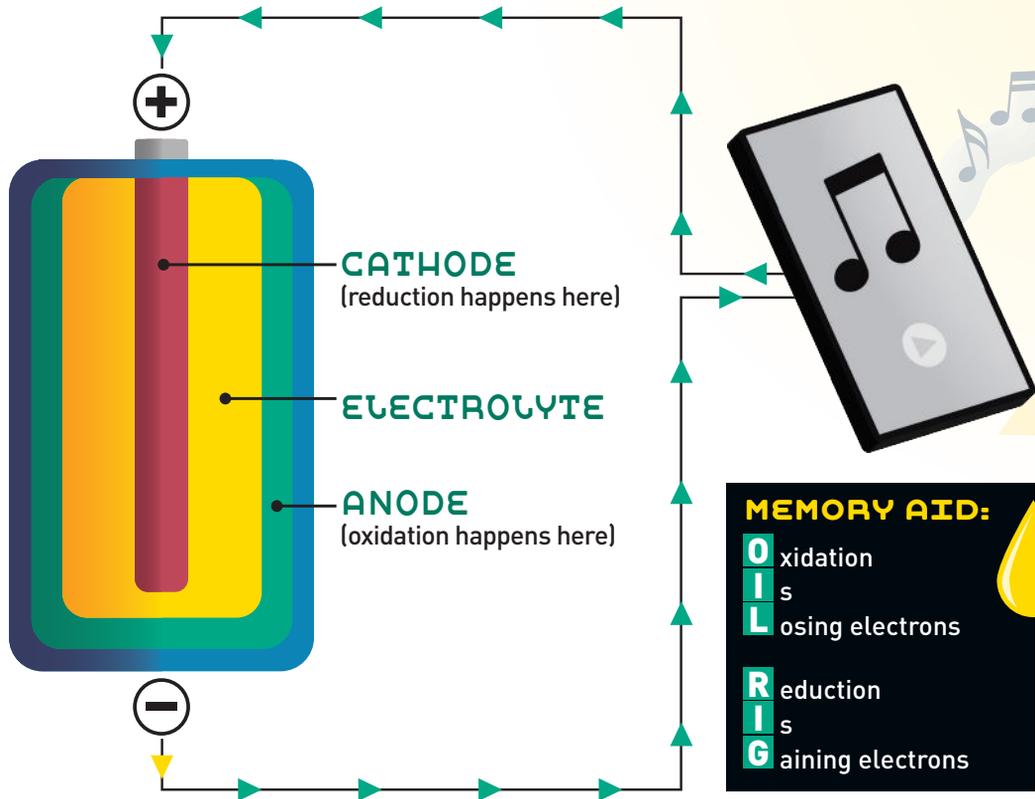
MATERIALS:

- » Galvanized nails, about 2" long (7)
- » Copper nails or wires, 2" long (7)
- » Sports drink or other electrolyte beverage such as Gatorade, Powerade
- » Alligator clips (14)
- » Ice cube tray
- » LED, 2V–3V
- » Cardboard
- » Marker
- » Voltmeter (optional)





ANATOMY OF YOUR BATTERY



ANODE

The galvanized zinc-plated nails are the anode in our battery. They have high-energy electrons that are removed from the zinc atoms, carried up the nail, and into the wire. These electrons will spend some of their energy to light up the LED. The process of losing these electrons is called **oxidation**. In our batteries, the anode is labeled with the negative (-) sign.

In a lithium-ion battery, **lithium atoms in graphite (carbon)** are often used as the anode, the source of our high-energy electrons.

CATHODE

The copper nails are the cathode in our battery. In the sports drink, there are some number of H⁺ ions in the solution, from the acids in preservatives or flavors. When the electrons come through the wire to the copper nail, these

H⁺ ions are reunited with electrons, forming hydrogen gas on the cathode. Over long periods you may notice that bubbles form on the surface of the copper wire! The process of gaining these electrons is called **reduction**. In our batteries, the cathode is labeled with the positive (+) sign.

In a lithium-ion battery, a **metal oxide** is commonly used as the cathode. It receives the electrons passed through the wire from the lithium metal.

ELECTROLYTE

In our ice-cube tray, the electrolyte is the sports drink. As reactions take place at the anode and cathode, they begin to build up charge. The anode is giving away negatively charged electrons, resulting in a buildup of positive charge. The cathode is receiving electrons, resulting in a buildup of negative charge. If too much charge

builds up, the battery will stop working — so we use an electrolyte solution to balance the charge back out.

Electrolytes are dissolved **ions**, charged particles that can move freely through the liquid. Sports drinks contain lots of electrolytes to replenish our own supplies after a workout. Sodium and potassium ions have positive charges (we call these **cations**), while chlorides and other **anions** have negative charges, all of which are necessary to keep our cellular processes working. These electrolytes are free to move about in a liquid, meaning that positive and negative charges can flow, creating electrical current. Our bodies use these electrolytes to move sodium and potassium ions to induce electrical charges in our brains, among other places!

In our sports drink battery, the negative chloride anions will move to the anode to balance the positive charge forming. Meanwhile, the positive sodium or potassium cations will move over to the cathode to balance out the negative charge being formed.

In lithium ion batteries, the electrolytes are composed of **lithium hexafluorophosphate**, a relatively stable chemical that helps shuttle lithium ions from the anode over to the cathode as the battery is discharged. Then, during charging, the same electrolyte helps move the lithium ions back to the anode so that your phone is ready to use tomorrow!

EXPERIMENTATION

We can use other electrolyte solutions besides salty water. Acids will also form ions in solution, providing the acidic H⁺ cation and the anion of choice. A common example is the carbonic acid in soft drinks and fizzy waters.

Think about other beverages you could use as electrolytes. Do they taste salty or sour? Make a prediction about each beverage and whether it would be a good electrolyte! Try a lemon/lime soda and milk as comparisons to the sports drink. What do you think will happen in each case? Note that due to the low current flow from this battery, weak electrolytes may not light up the LED at all. 🌀

BOOST JUICE

Electrolyte solutions can even complete an electrical circuit that's plugged into a wall! Here we can see a light bulb with two wires that are not connected. When we place the wires into distilled water, which has no electrolytes in it, the bulb cannot light up. There are no ions to move charge and complete the circuit!

In the electrolyte-rich Powerade, the ions are able to move from one wire to the other, completing the circuit and allowing the bulb to glow brightly! The more electrolytes that are present in the solution, the more current can be passed and the brighter the bulb can glow. For comparison, in a test with tap water, containing electrolytes like calcium and magnesium, the bulb glows, but nowhere near as brightly as with the Powerade.



TOP TO BOTTOM: Distilled, deionized water; tap water; and Powerade sports drink.

CAUTION: Wall power, 110V–120V AC, can kill you! Do not attempt this demonstration without adult supervision!

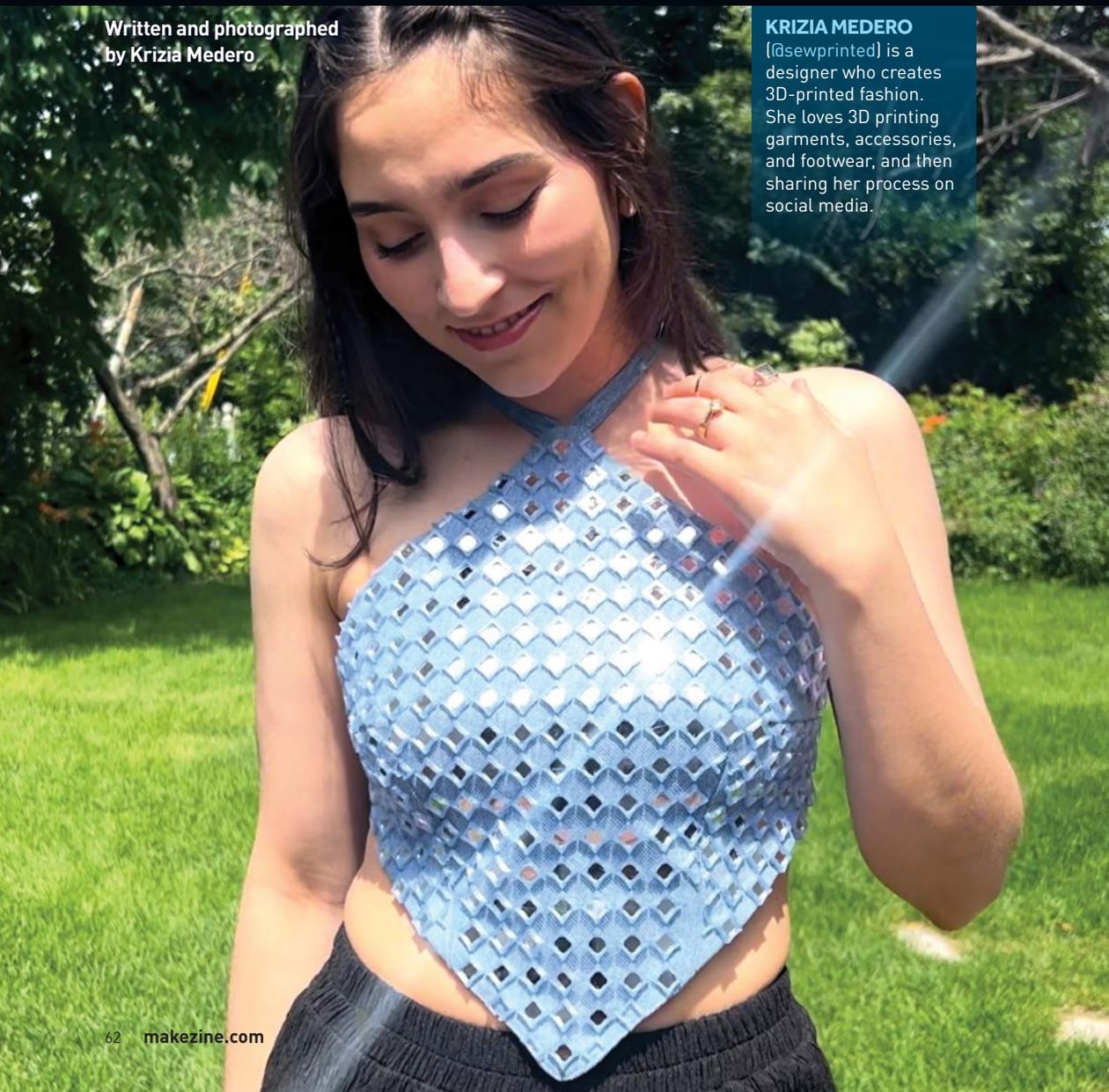
A safe way to compare different solutions is to use a conductivity tester such as United Scientific CNDT01.

3D-Printed Mirror Fabric

Embed tiny mirrors and mesh in your 3D prints to create sparkling fabrics

Written and photographed
by Krizia Medero

KRIZIA MEDERO
(@sewprinted) is a designer who creates 3D-printed fashion. She loves 3D printing garments, accessories, and footwear, and then sharing her process on social media.



I've been experimenting with 3D printing textiles for some time now. While exploring ways to create shinier 3D prints, I initially considered using holographic pattern flats on the print bed, aka *P-flats*, which offer a decent sparkle, but I wanted something more eye-catching.

The idea of embedding craft mirrors into prints came to me after researching an Indian embroidery technique called *shisha*. It took a few attempts to perfect the measurements, but I've successfully used this technique to 3D print a set of wearable wings, a halter top, and even a handbag with heart-shaped mirrors. I've designed and 3D modeled two STL files that you can print out to try this project yourself! It's suitable for beginners if you know some 3D printing basics like slicing a file, customizing print settings, and setting up a print.

At the end of this make, you will have a sparkly, 3D-printed-on-mesh textile, and you'll have learned how to embed mirrors and to sandwich fabric in your 3D prints. You can 3D print multiple swatches of these and use them to create your own garment, or you can use them as decoration or as a fun fidget item.

PRINT YOUR EMBEDDED MIRROR FABRICS

FABRIC 1 (SINGLE PAUSE)

This one is the simplest. You'll pause the print once and place the mirrors facedown and the mesh on top.

1. Start by cutting the mesh fabric into squares roughly the size of your printer bed. I like to use my removable print bed as a reference (Figure A).
2. Download the free STL file from cults3d.com/2141731 and bring it into your slicer.
3. Use the recommended print settings for the filament material you're using (Figure B).

TIME REQUIRED: 1–2 Hours

DIFFICULTY: Intermediate

COST: \$20–\$25

MATERIALS:

» For Fabric 1:

- Craft mirrors, round, 1cm diameter (49) available on Amazon

» For Fabric 2:

- Craft mirrors, round: 1" diameter (4) and 1cm diameter (12) available on Amazon

» For each fabric:

- Mesh fabric, nylon or polyester Choose a mesh with the least amount of stretch possible, and cut it to the size of your 3D printer's bed.
- PLA or PETG filament, about 15g
- Glue stick to cover your printer bed

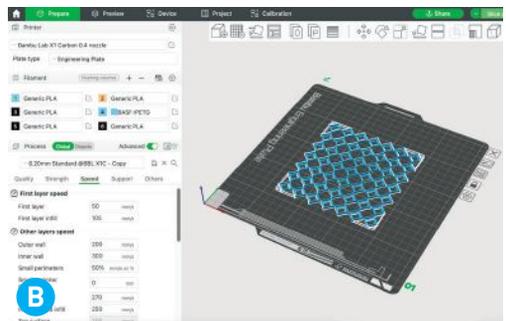
TOOLS:

» 3D printer

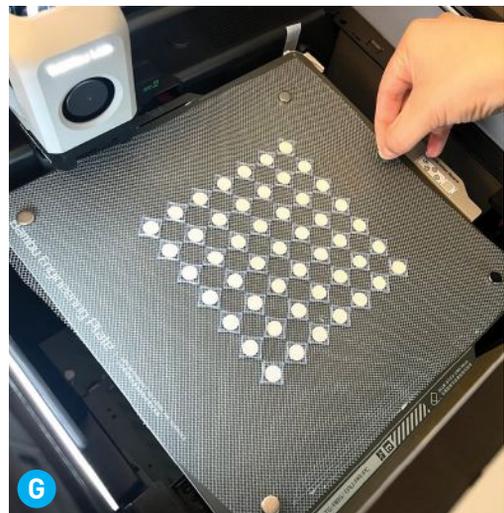
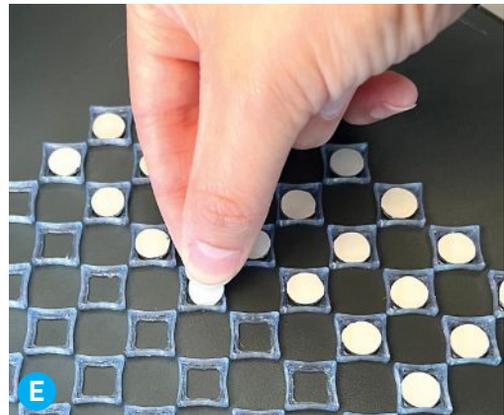
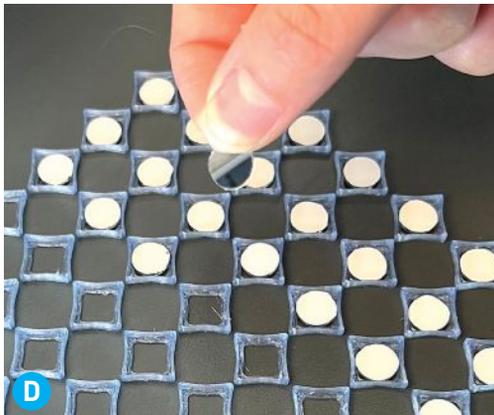
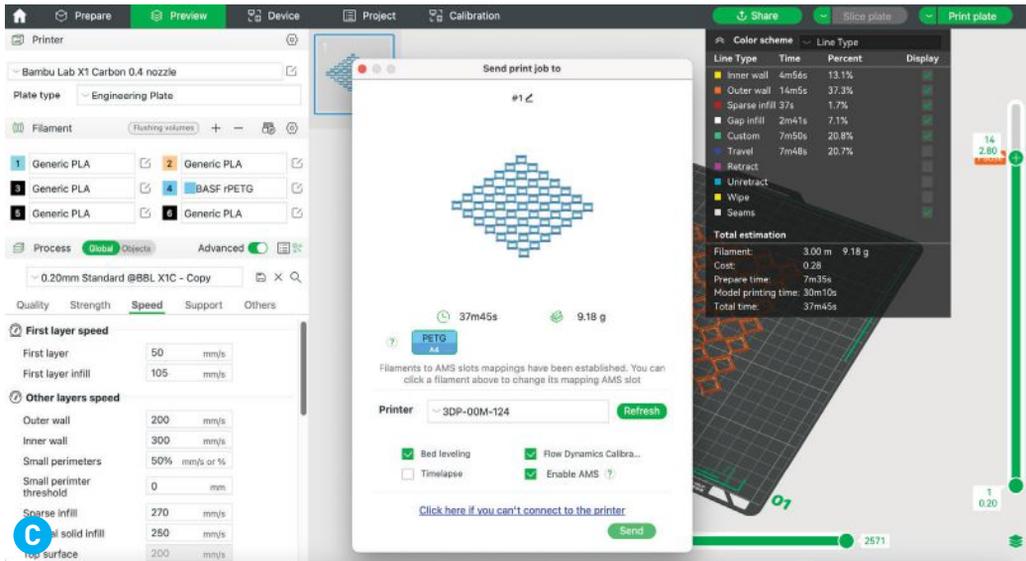
» Scissors

» Magnets (optional) to secure fabric to printer bed. Tape can also be used but magnets are preferable.

» Scraper (optional) to remove print from bed; helpful but not required



PROJECTS: 3D Print Mirror Fabric



4. Add a pause at the second-to-last layer (Figure **C**, orange indicator at right).
5. Add any adhesive you typically use to your printer's bed. I like to use glue stick.
6. Start printing.
7. When the print pauses on its own, set the mirrors into their places with the mirrors face down (facing the print bed) (Figures **D** and **E**).
8. Place the mesh across the print, covering the print bed (Figure **F**), and secure it in place with the magnets (Figure **G**).

TIP: Make sure to place the magnets out of the path where the nozzle will be traveling. I recommend placing them toward the outside corners of your printer's bed.

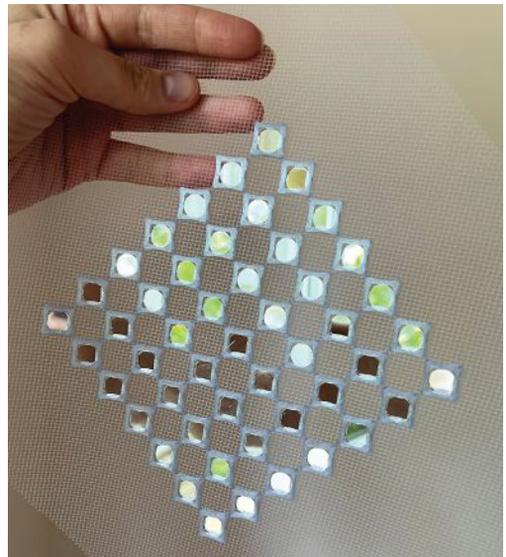
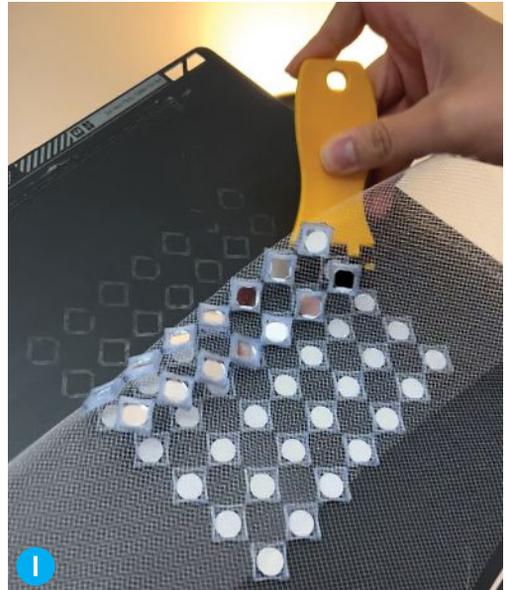
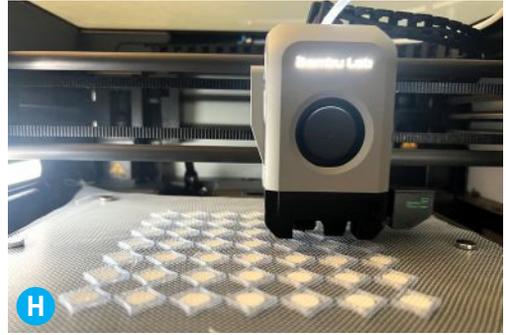
9. Resume the print (Figure **H**).
10. Once the print is done, carefully remove it from the printer bed. You might need to gently use a scraper (Figure **I**).

You're done!

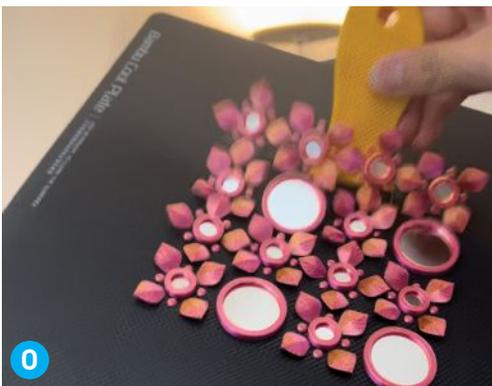
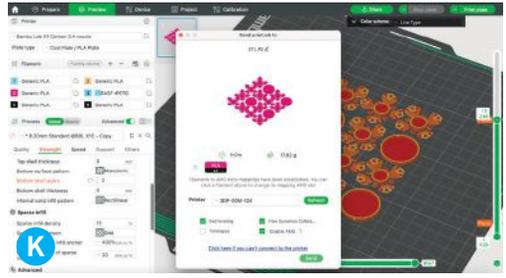
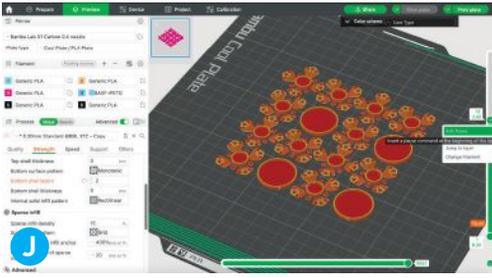
FABRIC 2 (TWO PAUSES)

This one is a little different; you'll pause once for the mesh, and a second time for the mirrors, placing the mirrors face up.

1. Again, start by cutting the mesh fabric into squares roughly the size of your printer bed.
2. Download the free STL file from cults3d.com/2141733 and bring it into your slicer.
3. Use the recommended print settings for the filament material you're using.



PROJECTS: 3D Print Mirror Fabric



4. Add two pauses to your print: the first pause at the second layer (Figure [J](#)), and the second pause at the second-to-last layer (Figure [K](#)).
5. Add any adhesive you typically use to your printer bed, such as glue stick.
6. Start printing.
7. When the print pauses on its own for the first time, place the mesh over the print, covering the printer bed, and secure it in place with the magnets (Figure [L](#)).
8. Resume the print.
9. When the print pauses on its own for the second time, set the mirrors into their places with the mirror side up (Figure [M](#)).
10. Resume the print.
11. Once the print is done, carefully remove it from the printer bed, and clean up any stringing or extra filament that may be covering the mirrors (Figures [N](#) and [O](#)).

You're done!

USING YOUR MIRROR FABRICS

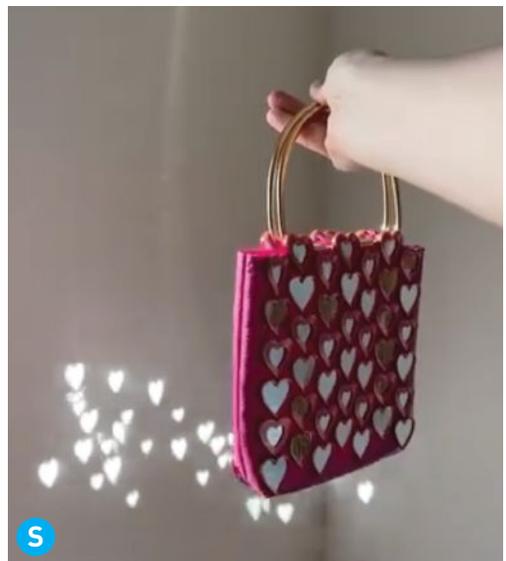
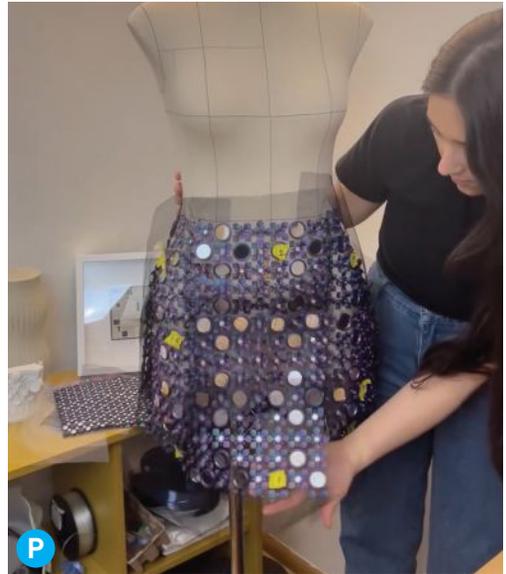
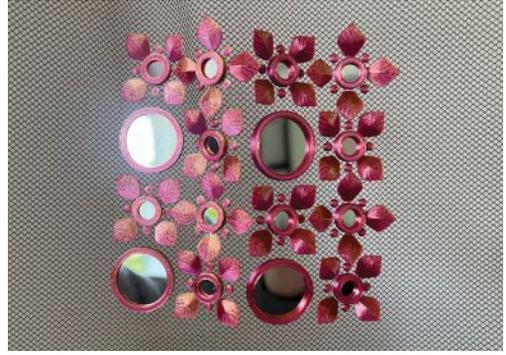
On its own, this project can be used for decoration, but it has the potential to be tiled together to create your own garments, like this mirror skirt (Figure P). You can see it come together at [youtube.com/shorts/N-5xzQ3mngl](https://www.youtube.com/shorts/N-5xzQ3mngl).

For this halter top (Figure Q), I had a pile of scrap denim lying around and I thought, “Why not make a project using only waste?” I printed swatches of Fabric 1 using a light blue transparent filament from BASF made of recycled plastic bags. Then I used a trick I learned from Daniel Bettencourt (Kaizen 3D Prints) to tile them together and use UV resin to seamlessly glue them. You can watch my process at [youtube.com/shorts/auUWM19qybQ](https://www.youtube.com/shorts/auUWM19qybQ).

Your mirror fabric can also be glued or sewn to an existing garment, and of course you can personalize it by printing it in different colors. This is also a great technique for costumes — I used it to make these wearable fairy wings (Figure R). You can buy the 3D files for printing them for a few dollars at cults3d.com/:1551997.

And this may just be my new favorite bag: I created a cool textile with these heart-shaped mirrors I found online and as I was printing it, I saw it was the perfect size for a small handbag ([youtube.com/shorts/EHUN1Z6ipP4](https://www.youtube.com/shorts/EHUN1Z6ipP4)). I was expecting it to reflect light in a really pretty way but I didn’t think that it would also reflect light in little heart shapes (Figure S)!

Let me know what you make with your mirror fabrics [@sewprinted!](https://www.instagram.com/sewprinted/) 📸





My Autotune Kazoo

Deploy real-time pitch shifting like the pros, in the most amateur instrument!

Written and photographed by Guy Dupont



The kazoo falls into a category of musical instrument that I would describe as “very easy to make noise, deceptively hard to sound good.”

All you have to do to make a kazoo do its thing is to hum into it! But in order to play in tune, you have to hum in tune. In other words — you have to be able to sing well.

I recently watched my 2-year-old figure out how to get one buzzing and thought to myself: “Given the ubiquity and accessibility of this little plastic instrument, how could I make it easier to sound good?” And then I thought: “Well, how do we make it easier for singers to sound good?” Autotune. In 2024, we use autotune!

Now, when I say “autotune,” what I really mean is **real-time pitch correction**. Auto-Tune is a proprietary software audio plugin developed by the company Antares. It is a specific implementation of real-time pitch correction, which means it is an algorithm that takes sound as its input, analyzes that sound to determine the most prominent frequency within it, and then modulates that sound to make the detected frequency match one of the frequencies in a given musical scale. In other words, it constantly moves the pitch of an instrument or vocal track up or down so that it’s never out of tune.

So I embarked (and eventually succeeded!) on a quest to create a kazoo with “autotune” built in. In my YouTube video (youtu.be/FDIRD5LRMc8) I demonstrate each of the kazoo’s features (sometimes in an embarrassing fashion) and tell the complete story of how I arrived at the current design. I wanted it to look, play, and sound like a normal kazoo as much as possible, which meant no visible wires or obvious modifications. Here’s what I ended up with:

- **The kazoo:** I searched “jumbo kazoo” on Google and bought the first thing that popped up. It ended up being a pack of these Chochkees brand 8” kazoos from Amazon. I knew that, at least for the first prototype, I would want as much room as possible to fit the electronics inside. The size didn’t end up being too ridiculous so I never swapped it out.
- **The microcontroller:** I knew I’d need something small and fast, since I would be

TIME REQUIRED: 1 Day

DIFFICULTY: Moderate

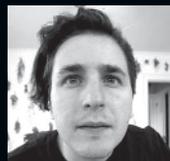
COST: \$50–\$60

MATERIALS:

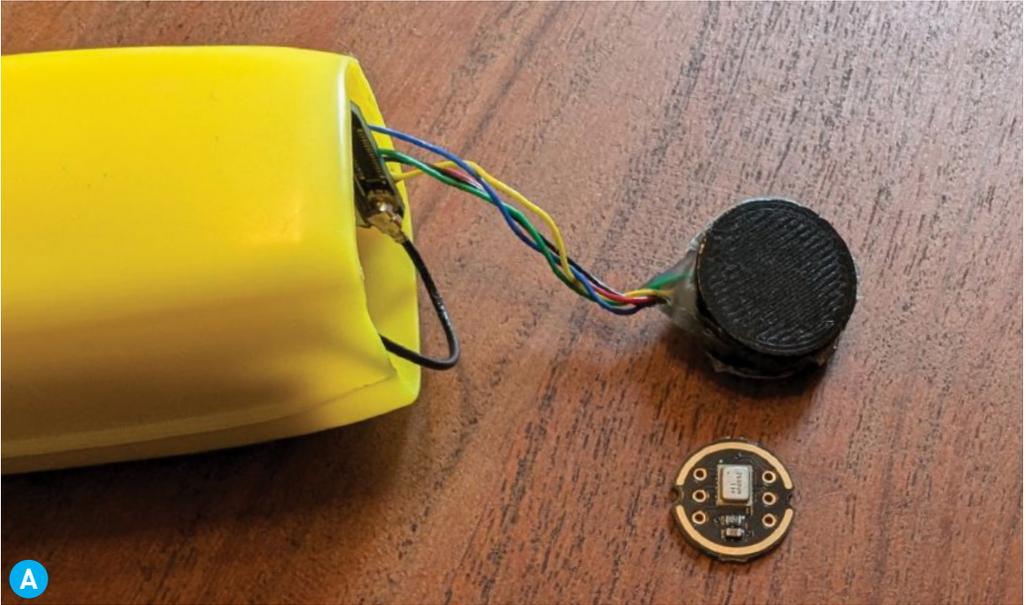
- » **Kazoo, 8" jumbo size** such as Amazon B0BZQKDW9F
- » **Microcontroller, Seeed Studio Xiao ESP32-S3** with USB-C cable for programming
- » **I2S amplifier module, MAX98357-based** Adafruit 3006
- » **Mini oval speaker, 8Ω 1W** Adafruit 3923
- » **Lithium ion battery, 3.7V, 2200mAh** Adafruit 1781
- » **LiPo Charger/Booster board, 5V 1A** SparkFun 14411
- » **I2S MEMS microphone module, INMP441** Amazon B092HWW4RS
- » **USB-C connector breakout, male (optional)** Amazon B0B24VN47M, for connecting 5V to the Xiao
- » **Enclosure for microphone** You can 3D print my STL from the GitHub repo or create your own enclosure to fit your mic module.
- » **Soft foam, scrap**
- » **Hookup wire**

TOOLS:

- » **Computer with internet access**
- » **Wire cutters/strippers**
- » **Soldering iron and solder**
- » **Hot glue gun**
- » **Small screwdriver**



GUY DUPONT is a software developer, hardware hacker, artist, and seltzer aficionado living in Cambridge, Massachusetts.



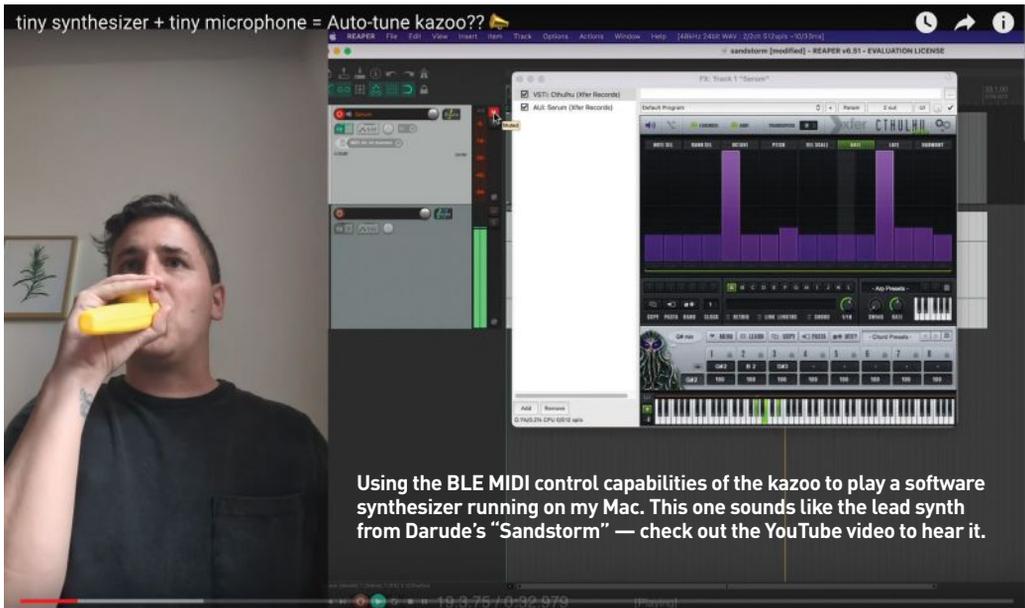
piping audio in, processing it, and then piping it back out in almost real-time. I reached for the Seeed Studio Xiao ESP32-S3 because it's super small, supports I2S for audio I/O, and can be clocked at 240MHz! As a bonus, it has built-in support for Bluetooth Low Energy, which I was able to leverage for some extra functionality — more on that later.

- **Speaker and amplifier:** I ended up using one of these mini oval speakers and a MAX98357-based I2S amplifier from Adafruit. Both were perfectly sized to slide inside the mouth of the kazoo. I ripped out the kazoo's buzzy plastic membrane (in the circular port at the top) and hot-glued the speaker in its place.
- **Battery and charger:** I used a cylindrical 2200mAh lithium ion battery that I found on Adafruit. I ran that through SparkFun's LiPo Charger/Booster board, which provides a steady 5V to power both the amplifier and microcontroller. It has a built-in on-off switch and Micro USB port for charging, both very handy for this project.
- **Pitch correction algorithm:** For pitch detection aka pitch tracking, I used an implementation of the *Yin algorithm* for fundamental frequency

estimation. To be honest, I don't understand how it works under the hood! I didn't even really try to, because it worked so well right out of the box. It seems to be both fast and accurate.

For pitch correction, I made slight tweaks to RobSmithDev's implementation (youtu.be/n1wGzWYwdk) which was originally written to run on the (considerably less powerful!) Arduino Uno (github.com/RobSmithDev/AutoTune). The Yin algorithm can only determine the frequency of what's being played, not the frequency of the note that *should* be played. I added an additional step that takes the output of Yin and selects my desired frequency — the nearest pure note — based on a set musical scale.

- **The microphone:** This was the part of the project I spent the most time getting right, and it ultimately defined how the entire thing came together. After some false starts with analog electret microphones, I ended up using a cheap, generic INMP441 breakout module. At first it might seem like a bad choice — the INMP441 is an omnidirectional MEMS microphone, which means that it is designed to be relatively sensitive and pick up sound from all around. I needed it to pick up sound from my mouth and only my mouth, from a very short



distance. In fact, I realized that the only way I was going to be able to isolate the sound of my humming completely from the speaker output and resonant kazoo body was to completely eliminate the distance between mouth and microphone — I decided to literally put the microphone inside my mouth!

To make this possible, I wrapped the INMP441 in soft foam to dampen the sound a bit, and then 3D printed an enclosure (Figure A) to go around that to keep my drool out. Note that I am using PETG and hot glue for the enclosure but I don't recommend putting either of those things in your mouth.

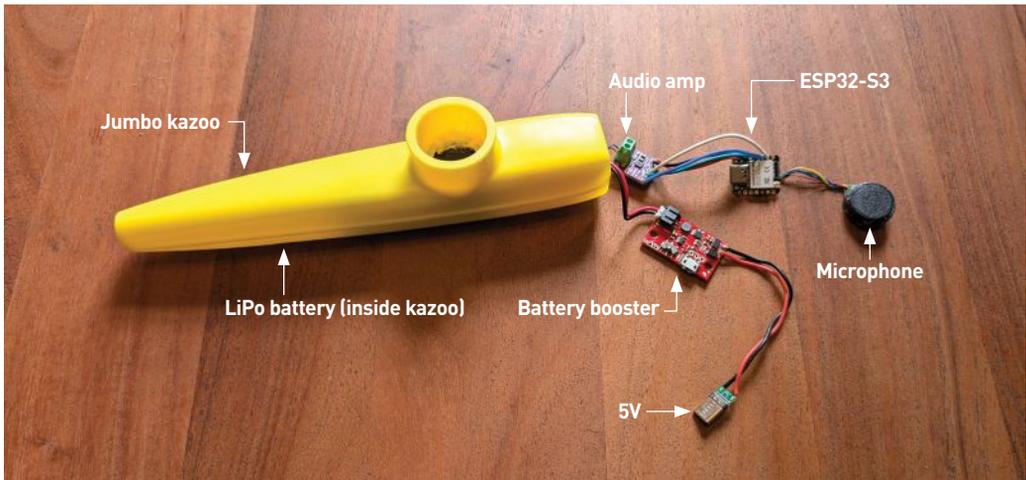
- **The trick:** It turns out that if you stick a microphone in your mouth, run the sound through some pitch correction, and then out through a tiny speaker embedded within a kazoo ... it doesn't really sound like a kazoo anymore! So instead of using the actual audio from the microphone, the speaker actually emits a synthesized *sawtooth wave* playing at the frequency provided by the pitch-correction algorithm. A sawtooth waveform is very easy to generate and, frankly, sounds exactly like a perfectly played kazoo!

This solution was tough for me to swallow at first — it felt like cheating. The sound wasn't

really my voice! But I realized that I had tried it the “hard” way and it was just ... worse. From a player/listener standpoint, the synthesized version sounded more like a kazoo, and it was way easier to control. It's sometimes hard as an engineer to sacrifice technical “purity,” even when it doesn't benefit your project in any way — especially when you share your work broadly. But ultimately, you can't let the opinions of those who'll fuss about how something is made prevent you from making something.

- **The bonus:** Bluetooth! I mentioned earlier that the ESP32-S3 has built-in BLE support. Since my “Autotune Kazoo” project eventually morphed into a “voice controlled synthesizer” project, I decided to take full advantage. I added code to allow the kazoo to send MIDI note messages based on the output of the pitch correction algorithm. These are sent over BLE to a host computer and can be used to control any software synthesizer or sampler!

And the connection runs in the opposite direction as well — you can send MIDI notes out from your digital audio workstation and play them through the kazoo's internal synth. It's pretty hilarious to set the kazoo down and have it appear to play itself.



BUILD YOUR AUTOTUNE KAZOO

1. FLASH THE MICROCONTROLLER

In the Arduino IDE, install Espressif's Arduino-ESP32 support following the guide at docs.espressif.com/projects/arduino-esp32/en/latest/installing.html.

Download the kazoo code in the *software* directory from the repository at github.com/dupontgu/AutotuneKazoo, then open that directory as a project in the Arduino IDE. With your Xiao plugged into your computer, select Tools → Board → XIAO_ESP32S3, then click Upload.

2. INSTALL THE SPEAKER

Hot-glue the speaker in place of the kazoo's original diaphragm (Figure B). Everything else just kind of slides in and out for easy repair.



3. CONNECT THE ELECTRONICS

Following the wiring diagram in Figure C, connect your speaker wires to the amplifier's output terminals using a small screwdriver. I snipped off the little Molex connector first.

Then solder the amplifier and microphone connections to the Xiao ESP32-S3 as shown.

Also solder the USB plug wires to the battery booster board's 5V output terminals: red wire to positive (+) and black wire to ground (-). You'll plug this connector into the Xiao's USB port to power it and the amplifier.

Don't forget to connect the Xiao's antenna if you're planning to use Bluetooth!

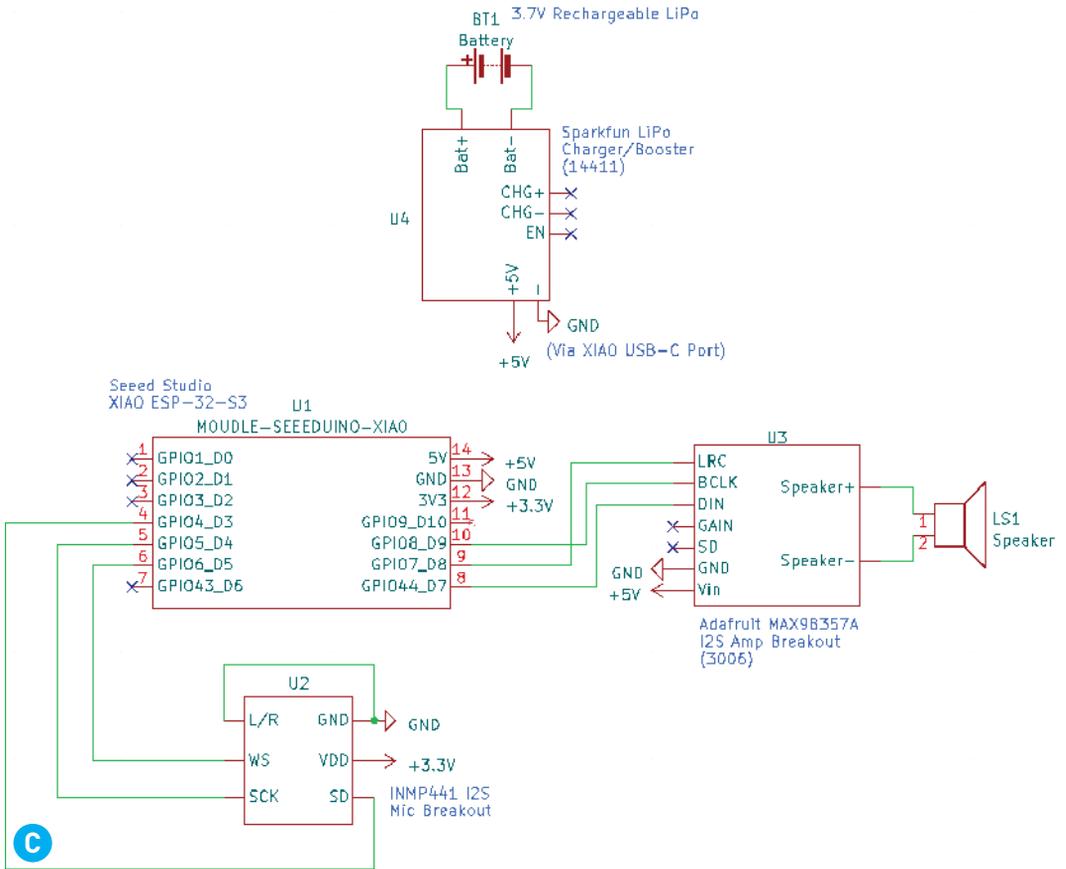
4. CONNECT THE BATTERY

Plug the battery's JST connector into the booster board.

5. ASSEMBLE YOUR KAZOO

Plug the Xiao into the battery booster, then slide everything inside the kazoo. The battery goes in first and wedges into the slimmer end of the kazoo body. The amplifier and microcontroller go in after that, followed by the battery charger board and microphone — those need to be easy to reach (Figure D).

Once you're sure everything's working, you can hot-glue the microphone module into its 3D-printed enclosure.



ANNOYINGLY PERFECT PITCH

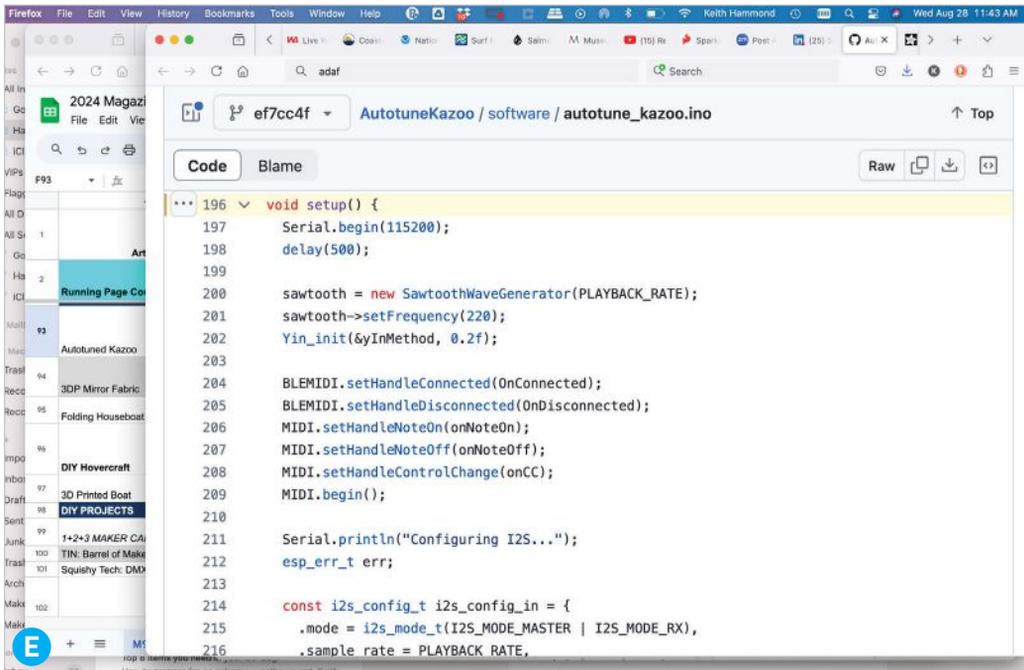
Put the mic in your mouth and give your Autotune Kazoo a go. It will automatically turn your out-of-tune humming into an appropriately annoying, buzzy sawtooth wave — with perfect pitch!

Here's a breakdown of the project code; you can follow along in *autotune_kazoo.ino*.

The `setup()` initialization function (line 196) is straightforward (Figure E on the following page):



1. First we create a new instance of the **SawtoothWaveGenerator** (line 200). This is what produces the synthesized sound samples that get fed out to the speaker.
2. Then we initialize the Yin algorithm (line 202) implementation. We pass it a float value (**0.2f**, for example) representing how strict we want it to be with the pitch detection. A lower value here means that the algorithm will only return a pitch estimation if it is extra confident. A



```
196 void setup() {
197     Serial.begin(115200);
198     delay(500);
199
200     sawtooth = new SawtoothWaveGenerator(PLAYBACK_RATE);
201     sawtooth->setFrequency(220);
202     Yin_init(&yinMethod, 0.2f);
203
204     BLEMIDI.setHandleConnected(OnConnected);
205     BLEMIDI.setHandleDisconnected(OnDisconnected);
206     MIDI.setHandleNoteOn(onNoteOn);
207     MIDI.setHandleNoteOff(onNoteOff);
208     MIDI.setHandleControlChange(onCC);
209     MIDI.begin();
210
211     Serial.println("Configuring I2S...");
212     esp_err_t err;
213
214     const i2s_config_t i2s_config_in = {
215         .mode = i2s_mode_t(I2S_MODE_MASTER | I2S_MODE_RX),
216         .sample_rate = PLAYBACK_RATE,
```

higher value means that it will give you a guess more often, but it might not be as accurate.

3. Next we initialize the Bluetooth MIDI library (line 204). We provide callback functions for when a connected host sends us MIDI data and tell it to start advertising and accepting connections.
4. The rest of the setup function (lines 214–277) is just boilerplate to initialize the I2S sound streams: input from the microphone, and output to the speaker. Beware that the pin number may need to change if you wire things differently!

The main **loop** (Figure **F**) does the following:

1. First, read a buffer (line 284) full of bytes from the microphone via I2S.
2. Then, loop through the samples (line 287) and convert them from **float** values down to signed 8-bit integers. This implementation of the Yin algorithm was optimized to work on 8-bit microcontrollers so that's what it accepts as input.

3. There are four buffers (line 23) that get filled up with these 8-bit samples, and each one has a different starting offset. This allows us to look at different “windows” of the incoming data and perform multiple pitch estimations on the same large chunk of incoming data. I found that the pitch estimation values were more accurate if I ran the algorithm more frequently and then averaged the outputs. So each time one of the four sub-buffers gets filled (line 303), the Yin algorithm is re-run and averaged with the last few estimations.
4. Each time a new pitch estimation value is available, we make sure it is in a valid range (line 112). The Yin algorithm returns **-1** if it can't make a good guess, and realistically nobody is humming into a kazoo at a frequency higher than a few hundred hertz so we ignore all values outside the range (0, 1500) to be conservative. (Side note: Apparently the world record for highest note sung was roughly 25,000Hz — well beyond the limits of human hearing!)
 - a. If the pitch estimation value is too far away (line 135) from the previous readings (more than 90Hz), we don't do anything! Assume

```

282 void loop() {
283     int bytesRead;
284     i2s_read(I2S_PORT_IN, (char *)samples, I2S_BUFFER_SIZE * 4, (size_t *)&bytesRead, portMAX_DELAY);
285
286     size_t samplesRead = bytesRead / 4;
287     for (size_t i = 0; i < samplesRead; i++) {
288         int32_t sample = samples[i];
289         // dividend is not meaningful, just changing the gain of the audio a bit
290         float adjSample = ((float)sample / 4147483647.0); //2147483647 is max for 32 bit signed
291         // put this back if we end up boosting the audio
292         // if (adjSample > 1.0) {
293         //     adjSample = 1.0;
294         // } else if (adjSample < -1.0) {
295         //     adjSample = -1.0;
296         // }
297         uint8_t normalizedInput = uint8_t((adjSample + 1.0) * 127);
298         for (size_t y = 0; y < 4; y++) {
299             auto index = yindicies[y];
300             yinBuffers[y][index] = normalizedInput;
301             if (++index == BUFFER_SIZE) {

```

this was an error, or the very beginning of a new **frequency**. If it is a valid new frequency, the next reading will confirm that. We have a little time before we need to change the output. Changing too quickly back and forth will sound bad.

- b. Otherwise, if the pitch estimation is outside of the valid range (line 139), we assume the user has stopped humming! We set our **activePitch** variable to **-1**, and we send a Note Off message to any connected MIDI devices.
- c. Otherwise, we assume we have a valid frequency! We convert that frequency (line 146) to a musical note. If that note is not included (line 149) in our targeted musical scale, we find the next closest note, either above or below the current one. We convert that note back into a frequency and update the **SawtoothWaveGenerator** with that value. We set our **activePitch** variable (line 157) to the updated frequency. If this note is different from the last one we detected, we send a quick Note Off MIDI message to any connected hosts for the previous note,

followed by a new Note On message for the updated one.

- 5. If there is a new MIDI Note On message (line 79) sent from a connected host, we override the **activePitch** setting from step 4 with the MIDI note. MIDI always takes precedence over the microphone.
- 6. After all the samples are processed, we look at the value of our **activePitch** (line 309) variable. It is set to **-1**, we fill our sound output buffer with silence. If it is set to a valid frequency, we populate it with a chunk of samples from our **SawtoothWaveGenerator** object. That's our sound going out to the speaker!
- 7. We write the sound output buffer (line 313) to the I2S output stream.
- 8. Finally, we poll (line 314) to see if there are any new MIDI messages waiting to be processed. 🎯



ANDY MCELFRISH

is a food writer, screenwriter, comic book writer, joke writer, book writer, and now, magazine writer. He's co-host of the *Edumacation* podcast with Kevin Smith, and lives in Los Angeles with his wonderful wife and laser cutter.



Ski Chalet Birdhouse

A laser cutter makes creative birdhouse designs extremely fun

Written and photographed by Andy McElfresh

Growing up, my dad and I made birdhouses

together. The only problem: They all looked like your typical seven-piece, pitched-roof, slab-built birdhouses made from pine shelving. Ten minutes on the radial arm saw, a few nails, and it's time to paint. Father/son time was kept to an absolute minimum, and the backyard trees became Bird Levittown.

Nowadays we have laser cutters that can help us create more intricate and interesting birdhouses. Father/son time includes coming up with an interesting idea and clever ways to put it together. My family's Extreme Birdhousing efforts have produced everything from Paris bistros to typewriters, robots to campers, even the Tardis from *Doctor Who*. With the laser cutter, we are essentially making kits that we can cut and assemble over and over again.

Before you go looking for those plans online, I wanted to get you started with a pretty simple build that's only a bit more complicated than the basic birdhouse: an A-frame-style Swiss chalet, complete with a pair of skis and picture windows (Figures A, B, and C). Since the birds like to feel protected, the windows are just there for show; they're backed with wood panels that you can paint to look like reflections of the Alps, or in this case, evoke the sky.

A nice feature of these laser-cut plans is that you can get very high-precision painting done simply by painting before you glue the parts together. And in the case of the windows on this

TIME REQUIRED: 1–2 Hours

DIFFICULTY: Easy

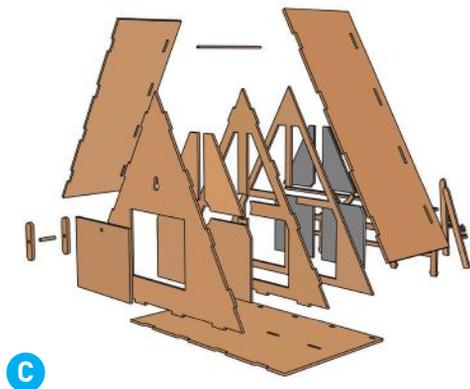
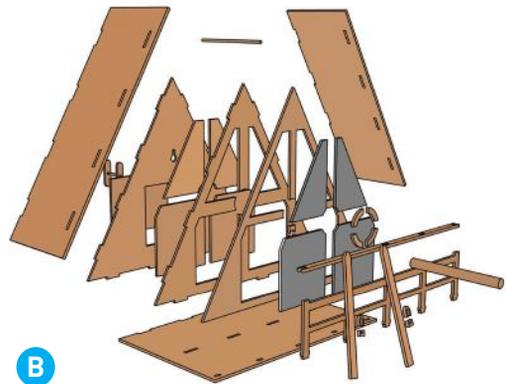
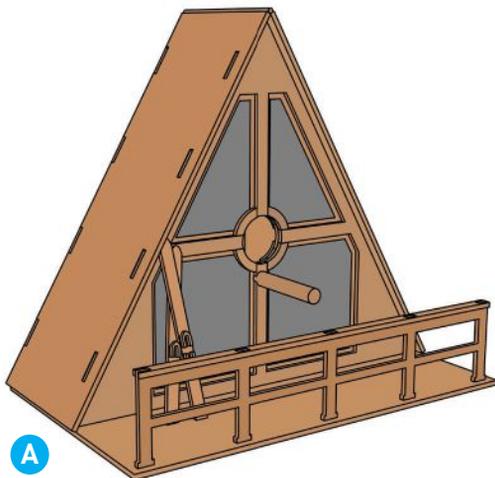
COST: \$50–\$60

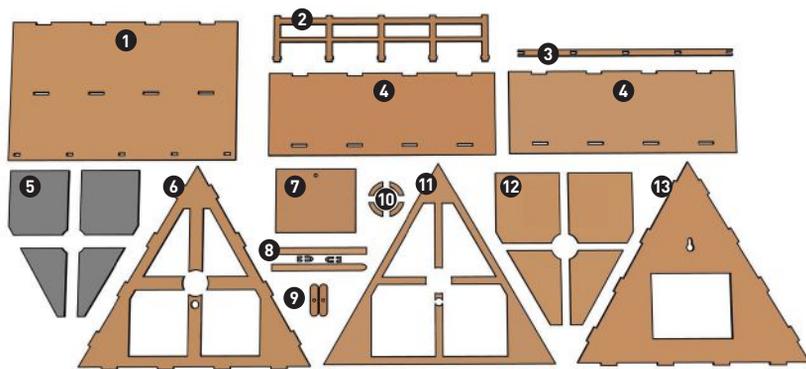
MATERIALS

- » Birch plywood sheets, $\frac{1}{8}$ " thick, 12"×20" (3)
- » Clear acrylic sheet, $\frac{1}{8}$ " thick, 8"×10" (optional) aka plexiglass, for the pretend windows
- » Wood dowel, $\frac{1}{8}$ " (3mm) diameter, 10"–12" long
- » Wood dowel or peg, $\frac{3}{8}$ " diameter, 5" long I like to use the handle from a foam paintbrush.
- » Strip of cloth or duct tape
- » Paints of your choice. Acrylic works well.
- » Masking tape
- » Wood glue

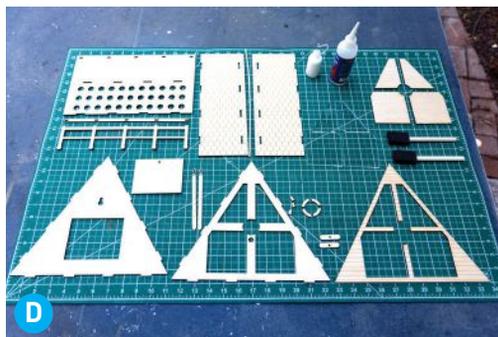
TOOLS

- » Laser cutter
- » Small handsaw
- » Paintbrushes





1. Base
2. Rail base
3. Handrail
4. Roof
5. Plexi windows
6. Window frame
7. Cleanout
8. Skis
9. Catch
10. Oculus
11. Window facing
12. Backing panels
13. Back



one, you need to paint the backing panels first, before gluing them to the inside of the structure behind the plexiglass windows.

BUILD YOUR EXTREME BIRDHOUSE

1. CUT THE PARTS

The laser cutter will produce most of the parts (Figure D). Download the vector files for cutting at andymcelfresh.com/make. You'll also need an 8" length of $\frac{1}{8}$ " (3mm) dowel, cut to fit the top of the roof, and a shorter piece to function as the pivot for the cleanout catch.

The peg is $\frac{3}{8}$ " in diameter and 5" long, with 1" extending into the birdhouse to give the birds something to grab onto on their way in and out. I normally use the handle from a foam brush for this: remember to mask it off if you don't want paint on it.

Finally, a short strip of cloth will act as the cleanout hinge: duct tape will do as well, since it's in the back and won't be seen.



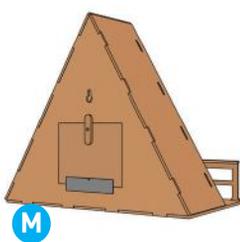
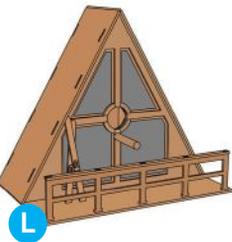
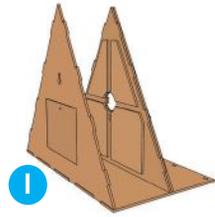
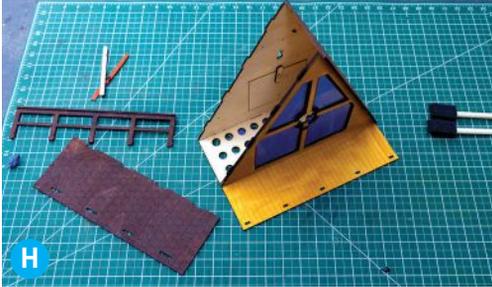
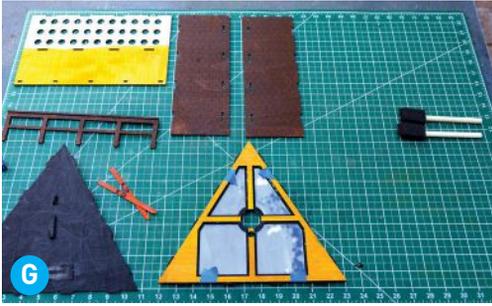
2. PAINT

Paint all parts and let dry. If you use acrylic paint, you will see the etched patterns of the shingles and flooring once it dries.



3. SUB-ASSEMBLIES

Insert the cleanout door into the back, and glue or tape down the hinge. Glue windows into the window frame — the other triangular part with tabs on all sides. Glue bindings to skis. Glue handrail to rail base (Figure E).



4. ASSEMBLE THE HOUSE

Glue the backing boards behind the windows (Figure F), and glue the window facing to the window frame (Figure G). Now glue the facing-frame assembly to the base. Glue back to base, using masking tape to clamp it. Glue roof pieces to frame (Figures H, I, and J).

5. ADD ACCESSORIES

Glue the rail to the front of the base. Glue the 1/8" dowel to the top edge of the roof. Glue skis to the front of the house. Glue oculus pieces around the hole. Glue the 3/8" peg under the hole. Glue the cleanout catch together through the hole in the cleanout (Figures K, L, and M).

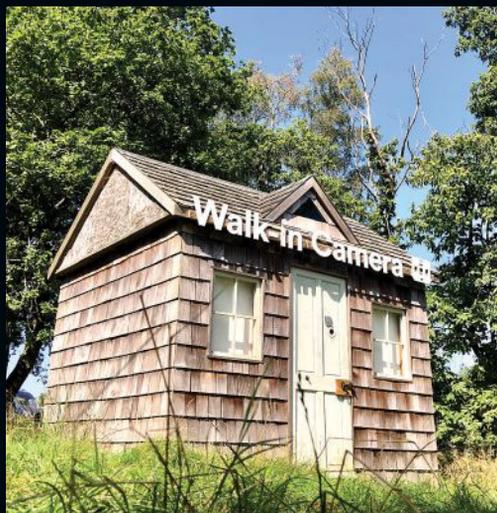
SNOWBIRD RESORT

Now you're ready to mount your birdhouse to a tree, balcony railing, or external wall. Be sure to make it sturdy: I like to leave the mounting screw a little loose, hang the house by the mounting ring, then tighten the screw down through the main hole. The hole is sized for small birds around the home: this chalet had a breeding pair of Bewick's wrens and their chicks this spring.

Take a look at my other extreme birdhouse builds and plans at andymcelfresh.com. I have an Etsy shop called WildBirdhousePlans and I also sell some kits there.

Enjoy! 🍷





Camera Obscura

Make a walk-in camera room to project the outside world upside down

Written and photographed by Nick Sayers

NICK SAYERS is an international science-artist, maker, graphic designer, planetarium presenter, bike activist, and urbanist based in Brighton, U.K. Read more about him and his work on page 16.

A *camera obscura* (“darkened chamber” in Latin) is a room that has been blacked out, with a lens or small aperture in the wall being the only window on the exterior scene. Light rays from outside pass through the aperture and are projected onto a white wall inside, casting a direct image of the view outside, but reversed — upside-down and back-to-front — onto the wall. We get our modern word *camera* (photograph-making device) from the contraction of this term.

Camera obscuras (walk-in camera rooms) have been around for centuries. There are theories that prehistoric cave paintings were partly inspired by the distorted projected images of animals seen through a camera obscura aperture. The effect has been written about since around 500 BCE, when Chinese astronomers used camera obscuras to observe solar eclipses. Dutch painter Johannes Vermeer is thought to have used a camera obscura to trace his paintings of dimly-lit interiors and people, including *Girl with a Pearl Earring*.

I was serendipitously introduced to this effect as a child on a family holiday. My brother and I were in a dark, white-walled bedroom, our parents next door in the brightly lit kitchen. We realized we could see an upside-down image of them walking about, projected through the keyhole!

As a science-artist, I’ve made many such rooms: in my own bedroom (where I’ve got a great view across the Brighton suburbs), an art gallery for a photographic festival, a woodland hut at a family festival, Brighton Upside-Down House, in a steamy sauna on a shower curtain (the world’s first Sauna Obscura!), and a sleeper train cabin — and another art gallery — on a trip to Uzbekistan. Here’s how to make one.

MAKE A CAMERA OBSCURA

1. FIND A ROOM

You might have a specific room in mind for this project, with only one window in which to put the lens. However, if you have a choice, it’s worth considering a few factors. Will the window look out at a scene with some movement in it? Moving objects (people, cars, clouds, wind-blown trees, etc.) reveal the magic of the direct-imaging process. Which direction is it facing? A view away

TIME REQUIRED: 1 Day

DIFFICULTY: Moderate

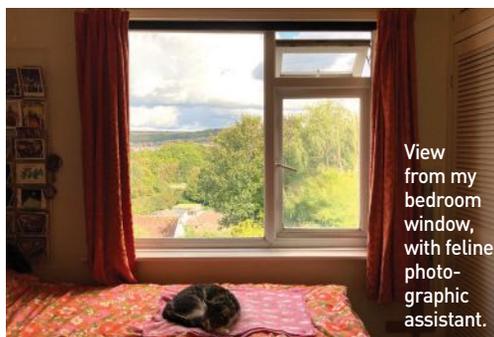
COST: \$10–\$50

MATERIALS:

- » Lens (see notes)
- » Cardboard
- » Black tape e.g. duct or gaffer’s tape
- » Clear tape
- » White paint (optional)
- » White boards (optional)

TOOLS:

- » Measuring tape
- » Craft knife
- » Scissors
- » Paintbrush



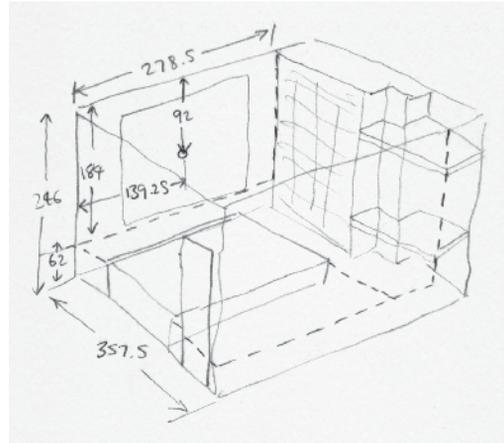
All photos © Nick Sayers

from the midday sun (south in the Northern Hemisphere, north in the Southern Hemisphere) is often better because you can see objects lit by the sun, rather than looking into its glare, with objects in its shadow.

For this project, I will use the example of my bedroom camera obscura at home in Brighton, U.K. The window points approximately northeast.



View from my bedroom window.



Sketch of my bedroom, with measurements.



Using Google Earth to measure the distance from the traffic roundabout to my bedroom window.



Measuring the distance from my bedroom window to the opposite wall, in meters.

2. MEASURE FOR THE LENS

Once you've decided which window the lens will go in, and which wall it will project onto, you will need to make some measurements to work out the strength of the lens required.

First, measure the distance from lens to projection wall.

Then measure the distance from the lens to the main point of interest outside. You might need a long tape measure, or for much longer distances, you can use the measurement tools in various mapping apps. My bedroom window looks out at a nearby road roundabout with lots of moving cars. I measured the distance from this roundabout to my house using the ruler tool in Google Earth (earth.google.com).

3. FIND THE LENS

It's possible to create the camera obscura effect without a lens — i.e. as a "pinhole" camera. For my improvised camera rooms in Uzbekistan, I created a 2" aperture in the sleeper cabin using a rolled-up travel pillow, and in the gallery using a 6" hole in a piece of cardboard. Blurry images of moving objects could be seen.

For sharper images, you will need a lens. Here's the formula you need for working out the strength of your camera obscura lens, measured in **diopters (dpt)**, with all distances measured in meters:

$$\text{Diopter (dpt)} = 1/\text{distance from object to lens} + 1/\text{distance from lens to projection surface}$$

For my bedroom obscura, the lens required is $1/199$ meters (window to roundabout, using Google Earth) + $1/3.575$ meters (window to bedroom wall, measuring tape) = 0.28 dpt.

Unfortunately, standard optician lenses only come in 0.25-diopter increments. Custom lenses are expensive to make. However, an optician I worked with had a great idea: he bought a pack of cheap 65mm-diameter lens blanks (the circular lenses from which shapes are cut to fit glasses frames). He then measured their actual strength using his *focimeter* lens-testing machine (aka *lensmeter*). Sold as 0.25 dpt, their actual strength varied due to production errors. One of the lenses turned out to be 0.28 dpt — that's 0.03 dpt off the advertised strength, but perfect for my bedroom's "prescription"!

Such lens blanks, or smaller 25mm test lenses (used in optician's test kits), can be bought online from eBay, opticians suppliers, etc.

4. BLACK OUT THE ROOM

A cheap way to black out a room is to use cardboard and black tape. Shops often get rid of cardboard boxes. Flatten them and cut them to size; you can tape several together for larger windows. To eliminate light leaks, use black tape at the edges of the cardboard screen, and black draft-excluder (weatherstripping) foam around door frames and other cracks.

Cut a hole to accommodate the lens, and tape it in place using clear tape (to reduce the area of lens blacked out) or more black tape.

5. CREATE A VIEWING SURFACE

Ideally, you should have a clear, white wall to view the projected image. You may need to paint this wall white.

Another option is to back-project the image onto translucent material such as a shower curtain or tracing paper.

TIP: The image of objects beyond the lens' focal plane can be brought into sharp focus by moving white boards further or closer to the wall.



Nominal 0.25 diopter lenses, marked up by Specky Wren optician with their actual strengths, using their lensmeter.



The lens, stuck into a hole in the cardboard blackout.



Bedroom window blacked out with a small aperture for the lens.



Close-up of the lens, inside the camera obscura room.



Upside-down projection of the external landscape onto my bedroom wall and work desk.

OUTSIDE, INSIDE, UPSIDE DOWN

Ready? Switch off the lights outside and inside your camera room. Go in, let your eyes adjust to the dark, and you should see an image of the outside world projected magically upside down! Enjoy!

FURTHER DEVELOPMENT

To experiment further with your camera obscura, you could:

- Make the installation more permanent by building the lens into a quickly-deployable roller blind or window shutters.
- Stack a few 0.25 dpt lenses together to focus on objects closer or further away. Diopters simply add together, for example, two thin 0.25 dpt lenses stacked together will give a 0.5 dpt magnification.
- Take photos of the resulting images. I like the way projected clouds appear to roll across the radiator and furniture in my bedroom. 📷



Detail, upside-down projection onto my bedroom wall.

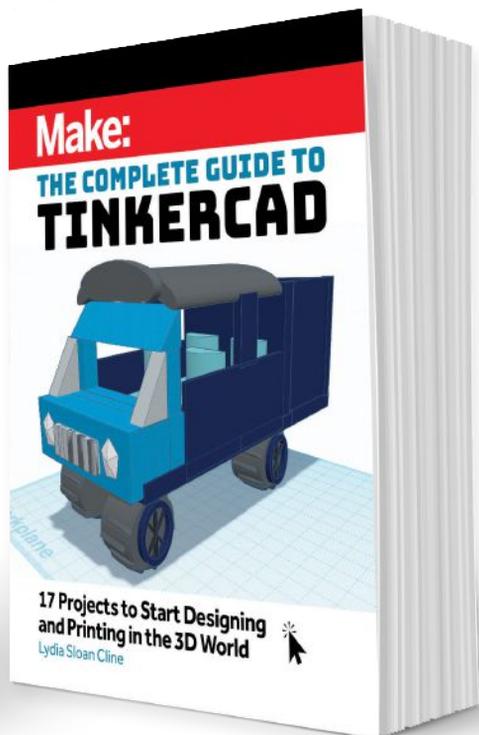
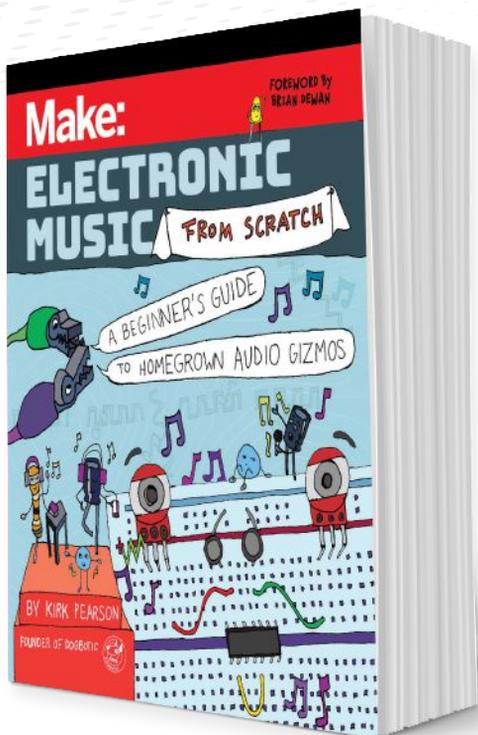


Upside-down traffic roundabout on my bedroom wall.

Make: ||| Books

'TIS THE SEASON FOR MAKING!

Packed with intriguing projects that make learning fun, Make: books are the gift of exciting experiences and new possibilities!



Make: Electronic Music from Scratch

By Kirk Pearson

Don't just play music — make music! This crash course in the joys of musical circuitry teaches makers to become musicians, and musicians to become makers. Written for total beginners by the mad geniuses at Dogbotic, this approachable guide uses more than 40 hands-on experiments and projects to build whimsical, wild, and just plain *weird* musical gizmos. *Make: Electronic Music from Scratch* is a fun, humorous introduction to music for even the biggest electrophobes that teaches how to build real, usable instruments. (And grab your Make: Dogbotic Music kit at the Maker Shed!)

\$39.99 makershed.com/electronic-music

Make: The Complete Guide to Tinkercad

By Lydia Sloan Cline

Bring your ideas to life! Embrace the concept of “learn by doing” through 17 fun projects that will transform dreamers into actual creators and producers. Make simple shapes and iterate increasingly intricate designs with Autodesk's free Tinkercad software; replicate everyday objects such as jewelry, toys, and household items for 3D printing; and import work into Fusion 360 for advanced modeling. Professor Lydia Sloan Cline leverages her extensive instructional experience to help novices roll up their sleeves and finally turn their ideas into a physical reality.

\$34.99 makershed.com/tinkercad

Get **\$5.00 OFF** these titles with the code **SNOWBALL!**

makershed.com



DMX Lighting for Makers

Set up this professional stage lighting protocol with just an Arduino!

Written by Lee Wilkins

FiddleHex by Lee Wilkins, Hillary Predko, Alex Leitch, Sagan Yee, and Dave Cox in *Lumière: The Art of Light*, Ontario Place, Toronto, 2023.



LEE WILKINS is an artist, cyborg, technologist, and educator based in Montreal, Quebec, a board member of the Open Source Hardware Association, and the author of this column on technology and the body and how they intertwine. Follow them on Instagram @leeborg_

If you've ever seen giant, bright lighting systems on stage, at concerts, or on big outdoor art installations, you've been looking at **DMX lighting**. DMX512 is a control protocol used by the pros but it's easy to use in your own projects with just an Arduino!

DMX stands for Digital Multiplex. It was invented in 1986 as an easy way to control lots of theater lights. DMX is really useful because it's a very simple protocol that lets you command a large number of elements with a single controller. Lots of off-the-shelf DMX lights are really made to light big things: like a stage, building, or sculpture. Using devices that are controlled through DMX can really make your work feel more professional.

DMX fixtures are often available in waterproof enclosures and can be mounted easily onto basically anything. With these tools, you can create cool distanced lighting effects and make some truly immersive stuff! Personally I've used DMX a lot on outdoor or large-scale installations. You can just get way more power out of a standard DMX fixture than a standard LED strip. They are super bright — I've used them to light sculptures even in full daylight. After all, they're made for this kind of thing!

DMX is a one-way communication, so devices cannot talk back to the controller. While it's most commonly used in stage work, I've seen a ton of really cool projects that use DMX to control a big array of other smaller things. It's not just for the pros — it works with familiar maker-friendly boards, too. In this guide I'm going to show you how to control these big light fixtures using your Arduino, and your favorite NeoPixel or FastLED patterns.

CHANNELS

DMX uses **channels** to transfer information along a chain of elements. Think of a channel as a numbered slot for information. Channels are arranged in a DMX **universe**, which contains 512 channels. You can have multiple universes if you want to control a lot of channels. Channels can also have a **personality**, which is the attribute they are controlling. For example, a channel might control light color, but it could also control the rotation or strobe of a lighting fixture. In another

context, this could be really any parameter you want. In a standard DMX stage lighting situation your channels might be Red, Green, Blue, and White, making it a four-channel light. It takes 23 milliseconds to transmit a DMX packet, which means we have a maximum frame rate of 44Hz.

CONTROLLERS, FIXTURES, AND SCENES

DMX lights use a single **controller** to control a large number of channels. Controllers can be either a normal DMX light that all other lights mimic, or a more complicated setup that drives patterns or cues on a stage. Commonly, you'll see controllers referred to as *master* and each light referred to as *slave*. I only reference this here so that you're aware of the language when researching; from here out I will be referring to them as controller and **peripherals** or **fixtures**. This is in reference to the Open Source Hardware resolution to replace master/slave words in SPI protocols, but I think it applies here too. You can read more about why at www.oshwa.org/2022/01/10/new-spi-terminology.

Each DMX fixture groups a series of channels. Below, you can see how the channels are sequential in groups defined by the fixtures. In a typical DMX controller, settings can be saved in **scenes**. Each fixture is basically a lamp.



DMX channel addresses are configurable, so you can set each fixture to its own address. If you're using your own code or a controller, you'll need to be aware of how many channels each fixture uses, and the order they're in, so that they don't overlap.

If you're using off-the-shelf controllers and lights, it's important to read the documentation to understand how the channels are working in that particular unit. It's really something that's impossible to guess, so don't be like me and waste hours playing with channel addresses — read the manual. Every single DMX light or model is different, but commonly there are buttons on the side or back of the fixture itself that can be used to set both the channel number and the controller or peripheral modes.

DMX CONNECTORS

DMX controllers originally used a five-pin XLR5 connector, but most entry-level DMX controllers use XLR5-size connectors that have only three pins. In most cases, DMX and three-pin XLR5 are interchangeable for hobby work or small installations. They're the same dimensions except two pins are missing, which is why they're able to mesh.

There are also mini XLR and mini DMX connectors, as well as a variety of intermediate sizes that are hard to assess without looking at the specs (Figure A). One thing I ran into is that there are a variety of three-pin XLR sockets on mid-range, weatherproof DMX lights that are in between mini and standard size!

Weatherproof or stage-quality DMX lights can be really expensive, so you'll find a wide range of qualities and with them a wide range of connectors. It's really worth double checking the specs on each connector while you're sourcing lights. I've been using DMX lights for outdoor installations, so it's important to me that they be IP65 rated which have specific connectors and are very well sealed. You can find affordable non-waterproof DMX lights on Amazon or AliExpress, but be aware they might be extremely not waterproof, with open enclosures! This is fine for indoor installations. I've seen people manufacture boxes to put them in with acrylic covers to keep them safe too.

In the end, it is just a three-pin connector that you need to plug into the standard three-pin DMX socket on your Arduino breakout, so there are a few simple solutions if your connectors don't match. I bought a few spare DMX cables and a variety of XLR sizes and created a series of adapters. These are pretty easy to make because there are only three wires. Cut your standard DMX cable and use a multimeter to find the corresponding wires and solder them. Make sure to use heat-shrink tubing over each connection and the whole wire, then you're ready for any budget DMX lamps.

If you are chaining together many lights, you only need to create one adapter to connect the Arduino breakout to the first fixture. Then, each subsequent fixture can use whichever style connector they came with.



CHOOSING AND USING FIXTURES

For fixtures, there are two main shapes I use: a **PAR can** or a **light bar**. A PAR can (Figure B) is what you might think of as a spotlight, although they do have a wide range of angles. Originally *parabolic aluminized reflector* lamps, there are now LED replacements in the same form factor. Some are tightly focused to light up a single narrow area, and others are more like floodlights that can light up a wide area. I used LED PAR cans in the *Earth Resist* sculpture on page 91 to light both the exterior and interior of the piece.

Light bars (Figure C) are wide, good for lighting up a wall or big area. We also used light bars to edge-light large acrylic panels (Figure D). They're great for covering a large area





with lights rather than highlighting a particular element.

You'll find that some of the lower-quality lights have individual red, green, and blue lamps on the fixture. While this might seem at first like you won't get the clean light colors you want to make, you'd be surprised at how well they do mix with a bit of distance. In Figure D, we used a cheaper lamp to create an edge-lit effect by putting it on the edge of a large piece of acrylic. At the base of the acrylic the three colors aren't mixed, but after a few inches they fully mix to create really nice colors. We covered the first few inches of the acrylic so that only the full effect was visible (Figure E). If you want lamps with RGB integrated together, it will cost you. But I really

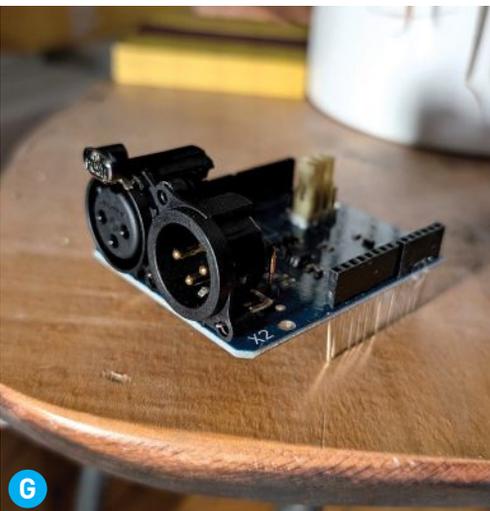


don't think it's a big deal!

If you give the lamps a lot of space, as in Figure F, the lights will mix in really interesting ways. I placed the lights at the bottom of these large metal sculptures and it created a great backlight effect while showing off the textures of the metal framing.

BREAKOUT BOARD FOR ARDUINO

For controllers, I have tried a variety of DMX breakout boards and settled on DFRobot's DMX Shield for Arduino (DFR0260) as my favorite (Figures G and H). It fits neatly onto an Arduino Uno, which is my preference for installation work, and it has a few jump switches that let you quickly program it and set different modes.





SET UP AN LED LIGHTING SYSTEM WITH DMX CONTROL

1. UNDERSTANDING THE CODE

After testing a range of DMX libraries, I have been using the Conceptinetics DMX library for Arduino (sourceforge.net/projects/dmxlibraryforar). It provides a series of really clear functions for using fixtures. You'll start by defining a series of parameters for your sketch. I've created an example:

```
#define channelSpacing 10 // This is the channels
```

Here we are setting the number of channels each fixture has. I chose to set **channelSpacing** at 10 because I was using a variety of different fixtures, some of which had four channels and some of which had eight. By setting it at 10, we are sure the channels won't run into each other. If you really want to fit in as many fixtures as possible, you'll need to fine-tune this.

```
#define DMX_MASTER_CHANNELS 512
```

This is the number of channels in one universe, but you may have bigger projects.

```
#define RXEN_PIN 2
```

This is connected via the DFRobot board. It is a jumper used to denote when you are in programming or running mode.

```
#define lamps 16
```

This is the total number of fixtures we will use. It's useful for running a loop in your sketch that iterates through each lamp. In this example, we want to loop through 10 channels for 16 lamps, sending data to each channel. So the loop will increase by the 10 (channel spacing number) each time, for the amount of lamps we have.

I also created a set of modifiers for each channel. For example, red is the first channel in each fixture, so it needs no adjustment. But white is the fourth, so for each loop of the lamp we need to increment by 4 to get to the white value.

```
typedef enum DmxCh {  
    RED    = 0,  
    GREEN  = 1,  
    BLUE   = 2,  
    WHITE  = 3  
} DmxCh;
```

2. PREPPING YOUR LIGHTS

You also need to prepare your fixtures. This will be different for each DMX fixture you have, so be sure to read the instructions.

You'll want to set your controller to its peripheral mode and manually input the starting channel address. You'll also need to make sure it connects to your Arduino, so that may involve either creating a new cable or purchasing an adapter.

It's worth making a map of all your lights and channels so you can fully understand how they work. You can also test each light by making the first light in the chain a controller, and seeing if the peripheral lights follow its sequence.

DMX and XLR cables do not carry power, so it's important to check how your power will be connected throughout your installation.

3. MAKING PATTERNS

There are a series of functions I've modified to help mimic how NeoPixel libraries are constructed, so that you can use your favorite NeoPixel patterns on big giant DMX lights.

First, let's look at how to set a DMX light manually:

```
void set_rgb_value(int lampNumber, int  
red, int green, int blue, int white) {  
    int channelNumber = lampNumber * chan  
nelSpacing;  
    // Set each of the colors  
    dmx_master.setChannelValue(channelNum  
ber + RED, red);  
    dmx_master.setChannelValue(channelNum  
ber + GREEN, green);
```

```

    dmx_master.setChannelValue(channelNumber + BLUE, blue);
    dmx_master.setChannelValue(channelNumber + WHITE, white);
}

```

This manually takes the lamp number and determines the channel spacing based on the offsets, then applies the RGB-W value you assigned. You can use something like this to manually set each pixel, but it's not ideal for mimicking your favorite patterns if you don't want to write new ones.

Fortunately there are two functions, `setPixel` and `showPixels`, that parallel those in the Adafruit NeoPixel library. NeoPixels use a buffer-style system that sets lights, then shows them. So I've had to replicate that with those two functions:

```

void setPixel(int index, int lampNumber,
int red, int green, int blue, int white) {
    pBuff[index].lamp = lampNumber;
    pBuff[index].red = red;
    pBuff[index].green = green;
    pBuff[index].blue = blue;
    pBuff[index].white = white;
}

```

The `setPixel` function uses a buffer to prepare the pixel at each index with a red, green, and blue value. `index` here indicates the position we are at in the array of lamps.

```

void showPixels() {
    for (int i = 0; i < index; i++) {
        int channelNumber = pBuff[i].lamp *
channelSpacing;
        // Set each of the colors
        dmx_master.setChannelValue(channelNumber + RED, pBuff[i].red);
        dmx_master.setChannelValue(channelNumber + GREEN, pBuff[i].green);
        dmx_master.setChannelValue(channelNumber + BLUE, pBuff[i].blue);
        dmx_master.setChannelValue(channelNumber + WHITE, pBuff[i].white);
    }
    clearBuffer();
}

```

The `showPixels` function is similar to `set_rgb_value` but it shows every single pixel available in the buffer. You can use this buffer style to immediately deploy your pre-existing patterns, like in this fade example:

```

for (int i = 0; i < 255; i++) {
    for (int j = 0; j < lamps; j++) {
        setPixel(j, i, 255, 0, 0);
    }
    showPixels();
    delay(5);
}
for (int i = 255; i > 0; i--) {
    for (int j = 0; j < lamps; j++) {
        setPixel(j, i, 255, 0, 0);
    }
    showPixels();
    delay(5);
}

```

Here the pixels are set after each loop, and then shown after the white strip has been set. These functions have the same name as standard NeoPixel functions, so you should be able to swap out your patterns easily. You can do this by replacing `strip.setPixelColor` with `setPixel`, and `strip.show()` with `showPixels()`, and `strip.numPixels()` with `lamps`. Keep in mind that in your code it may not be `strip` but whatever you have named your NeoPixel object.

DMX ME!

I'd love to see you take your projects to the next level and make something BIG! Share your DMX projects with me at hello@leecyb.org. 📧



Breathe Breezy

This DIY powered respirator keeps you safe and cool for a fraction of commercial prices

Written by Christina Cole and Victoria Jaqua



TIME REQUIRED:

2–3 Hours + 14 Hours Printing

DIFFICULTY: Moderate

COST: \$30–\$85

MATERIALS:

See the full BOM at makezine.com/go/FDSPAPR-BOM for part numbers and costs.

- » H13 HEPA filter
- » DuPont Tyvek 1443R fabric, 50cm×80cm
- » Non-woven polypropylene fabric, 10cm×10cm
- » Surgical face mask, Level 3
- » Face shield visor with lightweight foam band
- » Blower fan, 40mm×20mm, 5VDC
- » DC motor speed controller (PWM) with potentiometer switch
- » Step-up voltage regulator, 9V, U3V16F9 Pololu 4944
- » LEDs, 5mm, 3V: green (1) and red (1)
- » Capacitor, electrolytic, 16V 2200µF
- » Mini relay, 1A 5VDC, SPDT, PCB mount
- » Diode, 1000V 1A, 1N4007
- » Resistor, 62Ω
- » Piezo buzzer, 1.5–24V, chassis mount
- » USB power bank, 5V 10,000mAh
- » USB 2.0 Type A plug, 4-pin male connector
- » Two-conductor cable, 28 AWG, gray/black aka “figure 8” speaker cable
- » Stranded wire, 28 AWG in red and black
- » Heat-shrink tubing, 2mm diameter in red and black
- » 3D printer filament, ABS or PETG
- » White thread
- » Cord lock toggle
- » Elastic bands, white: 6mm and 20mm
- » Safety pin, tapes, and glues

TOOLS:

- » **3D printer** We used a Creality Ender 3 V2 Neo with 220mm×220mm×250mm print volume.
- » **Sewing machine**
- » **Soldering iron**
- » **Hot glue gun**

Need to do yard work during pollen season?

Clean up that dusty shop? Survive the next zombie outbreak? This might be the project for you!

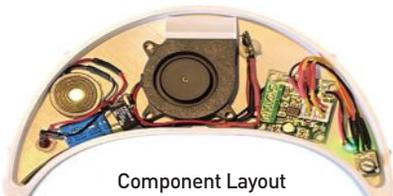
Open Source Medical Supplies (OSMS) is excited to present the Fernando & de Souza **powered air-purifying respirator** (FDSPAPR), an open source PAPR developed by the Australian team of Dr. Shiranth Fernando, a registered dentist, and Nathan de Souza, a mechatronic engineer. Innovated for healthcare worker protection during the COVID-19 pandemic, the FDSPAPR has emerged as a lighter, quieter, and more affordable alternative to traditional commercial PAPRs. It provides filtered air, face and eye protection, and skin surface cooling — without masking or requiring a large fan unit and air exchange tube, like most commercial PAPRs.

The FDSPAPR has undergone airflow velocity testing, and basic fogging and water resistance testing. While not yet tested for respiratory protection against airborne pathogens, the FDSPAPR does incorporate HEPA filtration, which can protect hobbyists from dust, pollen, and other large particulate matter. (Thus far we've been unable to test zombie splatter.) The wearer keeps cool and can speak and be heard easily while keeping the face and mouth visible, providing effective protection against particulates with

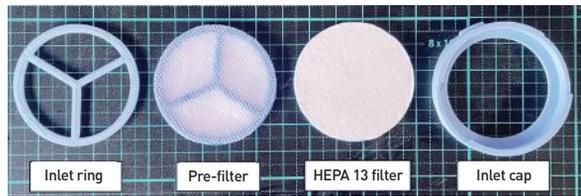
FDS PAPR	Commercial PAPR
Small USB power bank	Large, bulky 12V battery array
Weighs 220g	Weighs 1kg or more
Filtered exhaust outlet	No exhaust outlet
Fan noise level below 65dBA	Fan noise level 80dBA or more
No air tubing	Air tubing usually required
Less than \$100 DIY (\$30 or less in quantity)	Several hundred to \$1,000+

improved comfort and communication compared to traditional half-face masks.

Commercial PAPRs can cost hundreds of dollars, even \$1,000 or more. You can build an FDSPAPR in a weekend for less than \$100 by downloading the free 3D files and templates and following the instructions at Instructables ([instructables.com/DIY-Powered-Air-Purifying-Respirator-PAPR](https://www.instructables.com/DIY-Powered-Air-Purifying-Respirator-PAPR/)) and GitHub (github.com/nathanjds001/fdspapr). You'll need intermediate skills in 3D printing, sewing, and soldering. The rest is pretty much cutting fabric and hole punching.

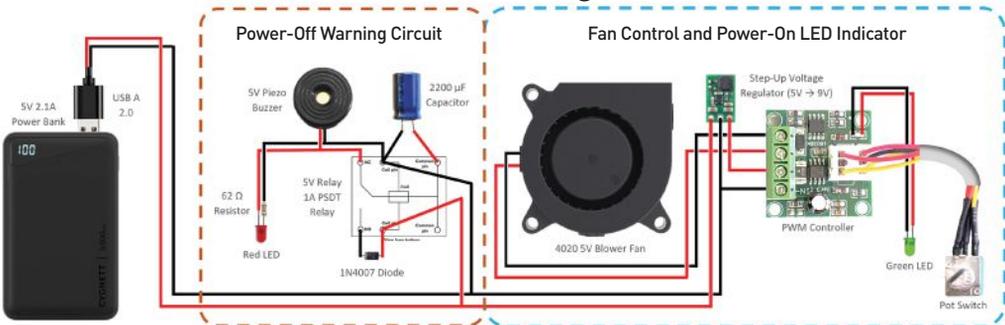


Component Layout



PAPR Circuit Diagram

Shiranth Fernando/Nathan de Souza



Powered Air-Purifying Respirator (PAPR)



FDSPAPR may be a versatile respiratory PPE solution for:

1. Users who need their face and mouth to be visible
2. Workers in hot, humid environments
3. Users with facial hair
4. Disabled and differently-abled users
5. Users sensitive to mask pressure on face and head
6. Users who struggle to find a mask that fits their face
7. Pediatric respiratory protection.

A PAPR FOR EVERYONE

Since 2020, OSMS has observed that standards and accessibility for personal protective equipment (PPE) and personal protective technology (PPT) struggle to match the broad needs of a new user market. No longer just for healthcare workers, individual interest in PPE/PPT has crossed into the public domain. Along with that interest has come confusion and challenges surrounding the current fit, operation, and accessibility of respiratory PPE.

In a 2023 report the National Academies of Sciences, Engineering, and Medicine found that U.S. standards for respirators and filtering masks fail to universally address fit and comfort across genders, ethnicities, and abilities (doi.org/10.17226/27094). This market gap could

potentially be addressed by FDSPAPR research — could an affordable, lightweight PAPR provide the same level of protection as a mask while improving comfort and usability for the general public?

REGULATORY RESEARCH AND TESTING

The FDSPAPR Design Report (makezine.com/go/FDSPAPR-design) thoroughly explores standards and regulatory compliance for international guidelines. These include the PAPR100-N and PAPR100-P classifications established by the U.S. National Institute for Occupational Safety and Health (NIOSH) to address the urgent need for safe and effective respiratory protection for healthcare workers during the pandemic.

To ensure the FDSPAPR's effectiveness and reliability, the design team followed several testing methodologies:

- **Filtration testing:** Using HEPA 13 filters, high efficiency of 99.95% of particles were removed.
- **Airflow measurements:** Exceeded the required 170 liters per minute, with measurements up to 300 LPM.
- **Decontamination procedures:** Comprehensive guidelines for cleaning and maintaining the device are provided.

NOISE COMPARISON IN DECIBELS (DBA)

3M Versaflo TR-300	<80 dBa
FDSPAPR	<65 dBa
Mortech MaxAir CAPR	<62 dBa

HELP DEVELOP IT

Since it's open source, you can help develop FDSPAPR to increase community value and adoption of the design. Be sure to document your research on the project GitHub repo! Some possible improvements:

- **Backup battery:** A backup system would allow the user to switch to an onboard battery if the power bank runs out, keeping the device on for longer while the user finds a new power bank. A rechargeable battery embedded into the visor could eliminate the need for a power bank.

- **Simplified construction:** A small circuit board for the electronic components would reduce wiring and soldering required in the head chamber. Eliminating the need for a sewing machine would make the device more accessible to a broader audience.
- **App control:** Create a phone app-controlled PAPR to select fan speed, monitor battery level, and monitor internal hood and external environment temperatures.
- **Customization:** Head sizes vary; the ability to 3D print different-sized PAPRs would make them accessible to more users. Adding an adjustable headlight would benefit users needing directed light sources at close range, such as dentists, artists, makers, and research professionals. Increased fan speed for face cooling would be advantageous for users in hot, humid climates.

RESEARCH POTENTIAL

As supporters of open source medical hardware, OSMS and the FDSPAPR team hope to see new open source filtering data and design iterations resulting from the current build files. Investigation avenues include:

- ISO PAPR 100-N testing
- Volatile organic compound (VOC) filtering
- Scaled manufacturing partners
- Community-focused PPE trialing
- Face-cooling air technology.

If you're a research or scaled manufacturing partner interested in collaboration, please email victoria@opensourcemedicalsupplies.org. 📧



CHRISTINA COLE (San Francisco)

is a founding member and head of documentation for Open Source Medical Supplies (OSMS) and serves on the steering committee for the Global Open Source Quality Assurance System (GOSQAS). She is also a co-founder and board treasurer at the REAP Center in Alameda, California.



VICTORIA JAQUA (Amarillo, Texas)

is the OSMS Medical Community Lead and curator for the OSMS Project Library. She serves on the GOSQAS steering committee and supports numerous open source medical projects toward deployment and publication.

Router Restarter

Cure the scourge of internet and Wi-Fi outages with this daily rebooter

Written and photographed by Theron Wierenga

TIME REQUIRED: A Weekend

DIFFICULTY: Moderate

COST: \$60-\$70

MATERIALS

- » Arduino Nano microcontroller board
- » IC socket for Arduino Nano
- » Dual relay board, 5V, Arduino compatible
- » LDC display, 20 character × 4 line, with I²C interface
- » Real-time clock (RTC) breakout board, DS3231
- » Coin cell battery, CR2032
- » Voltage regulator IC, LM7805, with mini heat sink
- » Capacitor, 0.1 μ F
- » Electrolytic capacitors: 1,000 μ F 6.3V (1) and 4,700 μ F 16V (1)
- » Duplex wall jack, rectangular, 120V
- » Power inlet socket, 3 pin, with mating computer power cord
- » Power jack, 3.5mm compatible with 9V 1A “wall wart” power supply
- » Printed circuit board or breadboard for point-to-point wiring. You can download my PCB design at makezine.com/go/router-restarter.
- » Project box Buy, build, or print it from the 3D files at makezine.com/go/router-restarter.
- » Miscellaneous: Hookup wire, spade connectors, header pins, 4-pin header jumpers (4" and 6"), #4×¼" standoffs, cables, screws and nuts

TOOLS

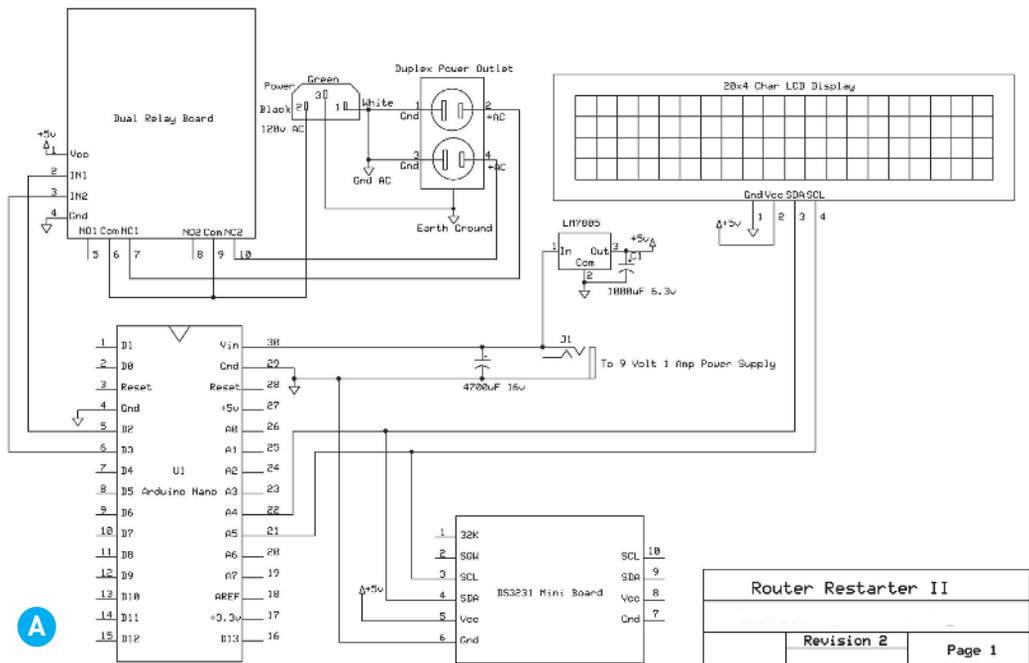
- » Phillips screwdriver
- » Soldering iron
- » 3D printer (optional) if you're printing the project box
- » Drill and drill bits
- » Thread tap, 4-40
- » Computer with Arduino IDE free from arduino.cc/downloads



Some years ago I built a Router Restarter for our condominium in Florida. We are “snowbirds” and only reside there in the winter months; from May to October we’re in Michigan. It’s important to have the air conditioning running in Florida during the summer to prevent black mold or other humidity problems. To make sure it’s operating correctly, I can check an internet-connected smart thermostat via my smartphone. But there’s one problem — sometimes the wireless router or cable modem in the condominium hangs up.



THON WIERENGA is a retired high school physics teacher, high school principal, and IT director. He has written for *Nuts & Volts*, *Servo*, and now *Make*.



A

Router Restarter II
Revision 2 Page 1

If you have Wi-Fi in your home you've probably experienced this, and the solution is to simply restart the cable modem and router by unplugging them from power for a few seconds and then plugging them back in. Unfortunately, it takes a person in the condominium in Florida to do this — while half the year we're in Michigan!

Enter the Router Restarter. This device simply restarts the cable modem and router by removing AC power to them for 1 minute in the middle of the night, every night. If there's ever a hangup, it gets corrected within a day.

RELAYS AND REAL TIME

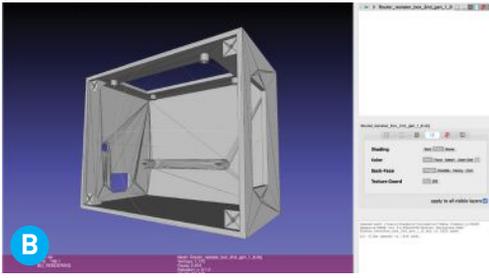
The microcontroller used for this project is the Arduino Nano, an easy-to-use workhorse for many purposes. My original Router Restarter used the DS3234 real-time clock (RTC) IC in a surface mount (SMT) package. This IC works nicely but requires Serial Peripheral Interface (SPI) programming and interfacing, and the tiny SMT package is a nuisance to solder onto a circuit board. Version II of the Router Restarter uses a DS32331 RTC breakout board that has an I²C interface and is simple to program and use. This breakout board also has a built-in coin cell holder for a CR2032 battery to maintain the current time

when power is lost.

The schematic for the Router Restarter II circuit is shown in Figure A. The Arduino Nano controls a 5-volt, dual-channel relay board. I chose relays instead of triacs for AC control, for their reliability and load capacity and because no heatsinks are needed. The normally closed (NC) contacts are used on the relays. This reduces current draw since the relay coils are activated only when the contacts are opened to cut power.

The Arduino Nano also drives a 4-line, 20-character LCD display for status. The circuit is powered by a 9-volt 1-amp wall wart type power supply. The 9 volts is connected to the Vin pin of the Arduino Nano and also powers a 7805 voltage regulator for the DS3231 RTC board, the LCD, and the relay board. Pins D2 and D3 of the Arduino Nano control the two relays, and pins A4 and A5 are the I²C interface for the LCD and the DS3231 (Figure A).

I designed a printed circuit board using free ExpressPCB software and ordered three of their mini boards using my design. With the many connections, this greatly simplifies construction. You can download my ExpressPCB design or the Gerber files from makezine.com/go/router-restarter.

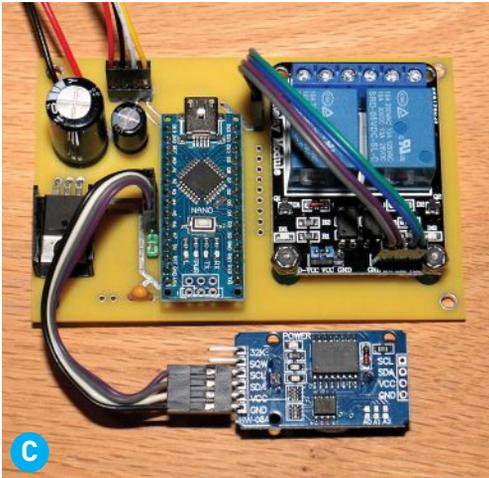


BUILD YOUR ROUTER RESTARTER

1. PREPARE THE PROJECT BOX

To 3D print the box, you can download my SketchUp designs and OBJ files (Figure B). This also makes for easy construction as the printed box has all the right-size cutouts in place.

If you buy or build your own project box, make sure it's large enough to accommodate everything since there are a number of wire connections needed inside the box besides the PCB.



2. MOUNT LCD AND POWER JACKS

In the walls of the box, mount the LCD display, rectangular duplex 120V wall jack (remember to remove the side tabs that connect the upper and lower sockets), 3-pin power inlet socket, and 3.5mm power jack.

3. SOLDER THE PCB

Solder the Arduino Nano IC socket, 5V dual relay board, LM7805 voltage regulator with mini heat sink, capacitors, and header pins onto the PCB. Then insert the Arduino Nano into its IC socket.

Figure C shows the PCB with the Arduino Nano, relay board, 7805, and filter capacitors mounted. A 4-pin header jumper connects the DS3231 RTC module at the bottom and another 4-pin header jumper connects the relay board to the PCB. A third 4-pin header jumper, visible at the upper left, connects to the LCD display.



4. MOUNT THE PCB

The inside of the box is shown in Figure D. The LCD module is at the top and the DS3231 RTC in the upper left. Power in connections are on the left, and the duplex 120V wall jack is on the right. The printed circuit board is centered left. If you're using the 3D-printed box, the mounting posts for the PCB and the tabs for the back cover should be threaded with a 4-40 tap.



5. WIRE THE 120V POWER

Keep the 120V wiring to one side of the project box. The relay module is shown in Figure E. The AC wires which connect to the duplex plug are connected to the normally closed second and third, and fifth and sixth, screw connectors at the left, numbered from the top. The jumper between

header pins JD-VCC and VCC needs to be in place as shown in the lower right. The 4-pin header in the upper right connects to the PCB for power and signals. A relay is activated when an input receives a low (ground) signal, and an LED on the module lights up.

When connecting the “hot” (black) side of the 120V AC to the duplex plug from the relays, remember to snap off the small connector between the screws on that side so that each plug will be controlled separately.

6. CONNECT THE JUMPERS

Connect the three 4-pin jumpers between the 4-pin headers on the PCB and the LCD, the DS3231 RTC board, and the dual relay board.

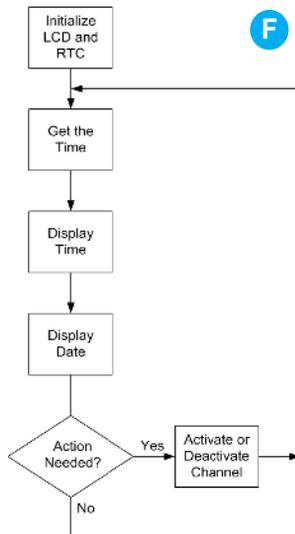
7. MOUNT THE RTC BOARD

Mount the DS3231 real-time clock board to the side of the box. I used a single #2-56 machine screw and a 1/4" standoff.

8. DOWNLOAD THE SOFTWARE

The software for the Arduino Nano is fairly simple. After initializing, it simply reads the time and date, displays this on the LCD along with the status of the two relay channels, checks to see if the relays need to activate or deactivate, and then loops. Figure **F** shows the program flowchart.

Two software libraries are included in the program, one for the DS3231 (github.com/NorthernWidget/DS3231) and the other for the 4x20 LCD display that I used (community.dfrobot.com/makelog-312225.html). There are a number of libraries for different types and sizes of LCD displays, so be sure you have the correct library that matches the display you use. Instructions



for installing libraries in the Arduino IDE can be found at docs.arduino.cc/software/ide-v1/tutorials/installing-libraries.

To program the Arduino Nano you must first uncomment the following lines in the program (delete the first // on each line), and assign the current time and date. For simplicity the 24-hour format was chosen. Install a CR2032 battery in the DS3231 breakout board. Compile and upload the program to the Nano and verify that it's running correctly.

```
// rtc.setClockMode(true); // 12
hour mode = true, 24 hour = false
// rtc.setSecond(0);
// rtc.setMinute(11);
// rtc.setHour(15);
// rtc.setDow(5); // 1-7
// rtc.setDate(1); // 1-31
// rtc.setMonth(12); // 1-12
// rtc.setYear(22);
```

Now comment out these lines again, and then compile and upload the program again. The battery backup on the DS3231 breakout board will now retain the correct time and date when power is removed, so powering up the Router Restarter will not change the running time and date.

9. CLOSE IT UP

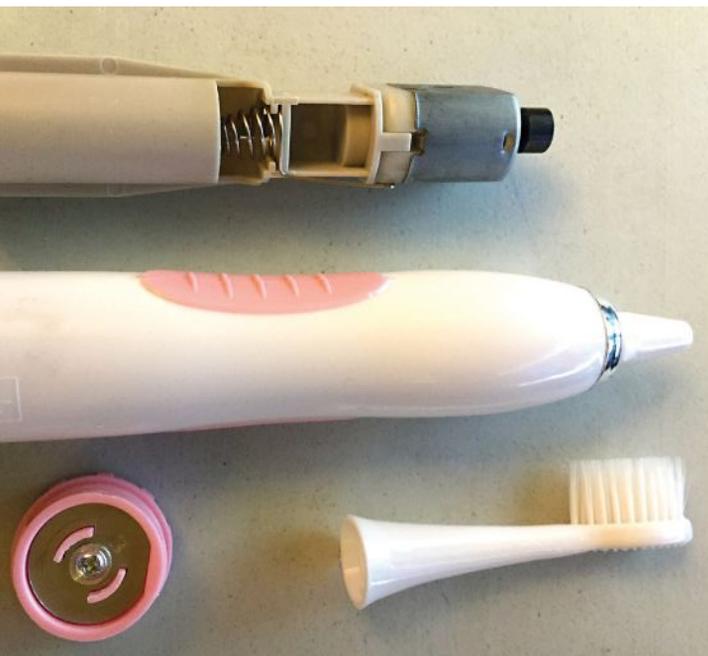
Attach the back cover with four #4-40 machine screws. If you like, you can super-glue the 3D-printed label to the front of the project box.

INTERNET ETERNAL

That's it! Plug your Router Restarter into the wall, plug your cable modem and Wi-Fi router into its duplex outlets, and they'll be rebooted daily whether you're home or away.

ADAPT TO AC OR DC

While the Router Restarter was specifically built to turn off and on a cable modem and router, with a few small changes in the program it can turn anything on or off that only requires a SPST switch, either normally closed or normally open. The circuit it controls can be either AC or DC, only limited by the relays' maximum rating of 10 amps at 240 volts AC, or 10 amps at 30 volts DC. ⚡



TIME REQUIRED: 20–30 Minutes

DIFFICULTY: Easy

COST: Free

MATERIALS:

- » Discarded electric toothbrush, battery powered
- » Cardboard
- » Plastic bottle caps
- » Paper clips
- » Adhesive foam tape, double-sided
- » Small box
- » Electrical tape and transparent tape
- » Black thread
- » Plastic spoon

TOOLS:

- » Pliers
- » Small screwdriver
- » Scissors

Hack Your Toothbrush

PART 2: Salvage the magnets from discarded electric toothbrushes and use them for fun projects!

Written by Cy Tymony

Hundreds of millions of electric toothbrushes are sadly discarded each year, yet most people don't realize that toothbrush parts are salvageable! Using the motors and magnets inside discarded electric toothbrushes (and toys and other small electronics) you can build all

kinds of practical projects.

In Part 1 of this article (Volume 87, makezine.com/go/hack-toothbrush-1) we built five projects using repurposed Micro 150 or 130 motors. Here in Part 2, we'll build five projects using the magnets found inside those motors.



CY TYMONY is a Los Angeles-based author and inventor who teaches super sneaky and simple ways to use everyday objects to make things. He's authored 10 volumes of his *Sneaky Uses* series, starting with the original, *Sneaky Uses for Everyday Things*, in 2003. sneakyuses.com

TOOTHBRUSH AND MOTOR DISASSEMBLY

These instructions are for the inexpensive electric toothbrush models that use replaceable AA batteries. Similar methods can be used for other varieties.

Remove the toothbrush end cap and battery. Twist the case apart. Pull the motor out, and pry the “wobble” weight off the shaft with a pair of pliers.

To open the motor case, pry the 2 side tabs upward with a small screwdriver. Remove the end cap and slide the rotor out of the case.

The two curved magnets in the case are held in place by a V-shaped metal clip which you can pry out with pliers and a small flat blade screwdriver (Figure A).



MAGNET PROJECTS

Reusing micro motors is lots of fun (see Part 1), but if you have a broken motor you can salvage the magnets inside to make these practical projects.

1. COUNTERFEIT CURRENCY DETECTOR

If you doubt the validity of your paper money, you can take advantage of the ferrofluid in U.S. currency inks to test for counterfeit bills.

Simply tape two or more motor magnets together. To make the magnets easier to hold, you can tape or glue them to a ring (Figure B), dowel, or toy wand.

Fold a dollar bill in half and hold it in the air. When you bring the magnets near the corner of the folded bill (Figure D), it will move toward the magnets!

2. SNEAKY CAR

You can use magnets to propel a small toy car. If you don't have a free-rolling car, you can quickly assemble a makeshift one using cardboard, bottle caps, and paper clips (Figure E).

The example shown here is made with two straightened jumbo paper clips and a 6"×5" piece of cardboard. Punch holes in the cardboard for the paper clip axles 1" from the top and bottom and 1/2" from the sides. Then fold the sides down 1" from the edge.

Bend the end of one paper clip at a right angle



PROJECTS: Toothbrush Motor Magnets

and then into a V shape (Figure F). Secure it to a bottle cap wheel using double-sided foam tape (Figure G). Stick the paper clip through the two holes, bend the other end the same way, and tape it to the bottle cap. Repeat for the second axle.

Tape a pair of motor magnets on the rear of the vehicle using adhesive foam tape, and you can move it with your ring or dowel. Make sure your vehicle magnets and ring magnets are pointing their North poles at each other so they will provide a repelling force.

3. LEVITATE PHOTOS

The same set of motor magnets allows you to make a floating photo display! Obtain a small gift box and tape the magnets inside, at the top.

Cut off a 2" piece of a small paper clip and bend it so the ends cross each other (Figure H). Then tie an 8" length of black thread to it. Tape a small piece of paper to the paperclip (Figure I).

Tape one end of the thread to the bottom of the box. Allow some extra thread to dangle so you can pull on it to alter the height of the picture.

Finally, tape a small photo to the piece of paper (Figure J).

When the photo is suspended at the proper distance from the magnets, it will appear to levitate (Figure K), since it's attracted to the magnets but the thread is preventing it from moving higher. To hide the magnets you can cover them with black electrical tape.

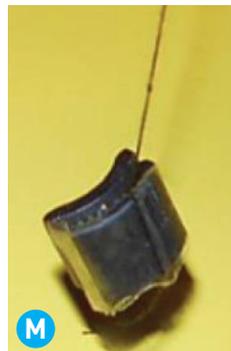
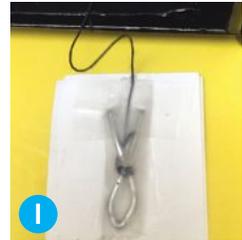
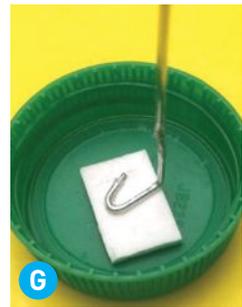
4. SNEAKY COMPASS

Direction finding is critical for hikers and campers. Magnets can be useful to locate the North/South bearing and possibly even save a life. Here are three ways to make a compass with your motor magnets:

Spoon Compass

Break the handle off a small plastic spoon and stick a small piece of adhesive foam tape to its center. Notice the bottom curve where the spoon balances, and stick the magnets there on the foam tape (Figure L). Ensure the spoon can spin freely. If not, remount the magnets until it does.

Spin the spoon several times and it should always stop in the same position, which indicates the North/South direction. Be sure to do this on



a flat, smooth surface without any other magnets nearby.

Thread Compass

Wrap or tape a piece of thread to two motor magnets. Suspend the magnets (Figure M), and they will eventually stop in the North/South direction.

Needle Compass

You can use a needle to make a sneaky compass — but instead we'll use the small V-shaped magnet clip from inside our motor!

Straighten the clip. To magnetize it, stroke it with a magnet 20 times in the same direction. Stick the clip through a piece of sponge or adhesive foam tape (Figure N) and gently float it on the surface of still water (Figure O). The clip will eventually settle and point in the North/South direction.

5. CIRCUIT ACTIVATOR

Toothbrush motor magnets can activate electrical devices at a distance and through objects. Here's how to make a magnetically sensitive switch.

Connect a 3V–9V battery supply to a device like an LED or buzzer, in a circuit with wire and a paper clip as shown in Figure P. Tape the wire loop over the paper clip (Figure Q) and ensure that they do not make contact. This acts as the circuit's switch. When the paper clip moves toward and contacts the wire loop, it will activate the device.

Now carefully hold your magnet over the wire loop and paper clip. The paper clip will rise, touch the wire, and turn on the LED!

Going Further: If you substitute the wire and paper clip with a **reed switch** (Figure R) — a thin, magnetically sensitive switch — you can place this simple circuit inside greeting cards, toys, and more, using a 3V watch battery, an LED, and the reed switch (Figure S).

MORE WITH MAGNETS

Small motor magnets can provide many more possibilities for science and craft projects. Here are a few suggestions:

- Motor magnets can be glued to small



ornaments and photos to be held on your refrigerator or other metal panels.

- Place magnets on desks or inside drawers to secure thumbtacks and paper clips.
- Add small magnets to purses, cabinets, and the end of sliding doors to keep them closed.
- Magnets can be used to detect studs in walls, and to find small parts, like screws, that have fallen on carpet and grass. 🍀



Barrel of Makeys!

Update a classic toy
with the *Make: mascot*,
Makey the robot

Written and photographed by Bob Knetzger



TIME REQUIRED: 3-4 Hours

DIFFICULTY: Easy

COST: \$5-\$10

MATERIALS

- » Acrylic sheet, 1/8" thick
- » Glue: cyanoacrylate "super glue" or acrylic solvent

TOOLS

- » Laser cutter

Here's a fun and easy update to the classic skill and action game **Barrel of Monkeys**. This version adds an easy skill level and other options for making and playing. And the monkeys are now long-armed versions of Makey, the robot mascot of *Make*: and Maker Faire.

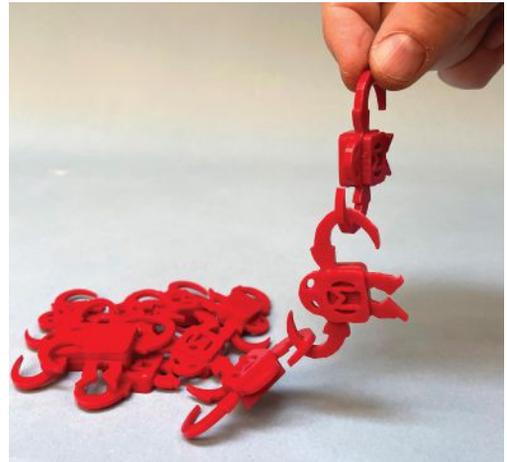
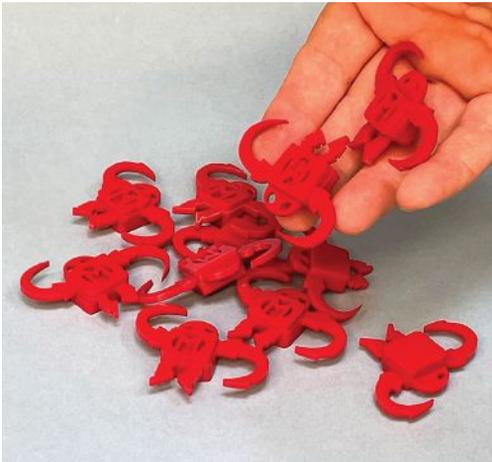
Simple to play: Dump the Makey bots in a pile. Then, holding one bot by the arm, link arms with another Makey bot and lift it up. Keep adding to your chain: How many bots can you link up before one falls? Whoever makes the longest chain of Makeys wins!

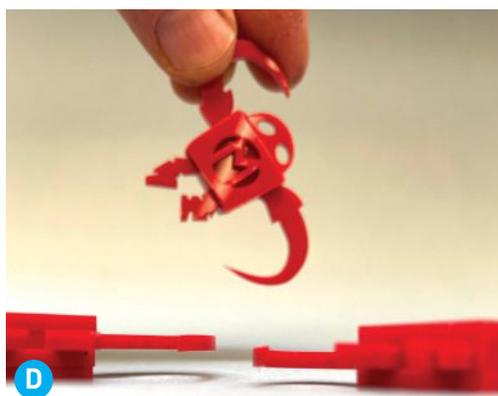
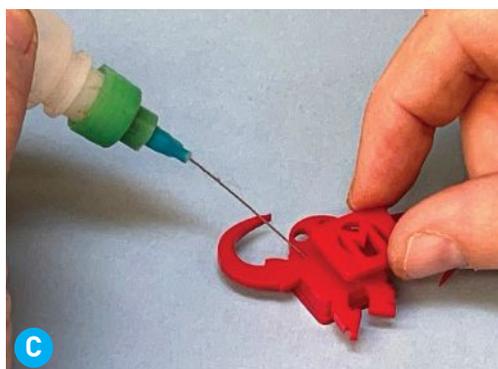
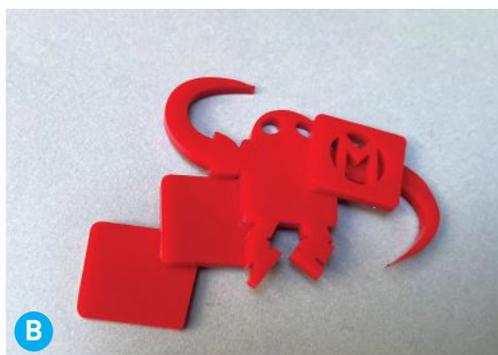
MEET THE MONKEY

Independent toy inventors Leonard Marks and Milton Dinhofer created the first version of this game, called "Chimp to Chimp." It sold modestly at first but it when it was licensed to Lakeside Toys in 1965 it became a big hit, thanks to the promotional power of a cute, monkey-filled television commercial (youtu.be/tNhcQEUbw0&t=164s).



BOB KNETZGER is a designer/inventor/musician whose award-winning toys have been featured on *The Tonight Show*, *Nightline*, and *Good Morning America*. He is the author of *Make: Fun!*, available at makershed.com and fine bookstores.





Barrel of Monkeys went on to become a classic game, sold by various companies up to the present day. It's been featured in all the Disney *Toy Story* movies (youtu.be/_fSAyvaTQvE). The TV show *Family Game Night* has a Barrel of Monkeys live game version, there's a video game version in arcades, and at Disneyland Paris you can shop inside a big, blue, Barrel-of-Monkeys-shaped store (youtube.com/shorts/jwY843zyDBU).

MAKE IT!

A laser cutter is perfect for making the game's pieces. Sure, you could cut the parts out by hand (and stack them to cut a batch at a time), but it's much faster and more fun than a barrel of Makeys to use an automated laser cutter to knock 'em out, fast!

Since there are no precision parts, I used an interesting feature of the Glowforge laser "printer" to make a quick batch. Instead of creating vector artwork to generate an SVG tool path, I used the Glowforge's "Trace" feature: Just place any line drawing on the cutting stage and the built-in camera automatically takes a picture and creates the sequence of cuts to make your part (Figure A). Anyone can do it! For other laser cutters, go to makezine.com/go/barrel-of-makeys to download the .svg file.

Cut the four-piece design out of $\frac{1}{8}$ " acrylic sheet (Figure B), then stack and glue the parts and together with cyanoacrylate or acrylic solvent as shown (Figure C). Make 12 Makeys in all, just like in the original game.

Unlike the original, this design has a new easy-play feature: the double-thick back raises the arms off the table surface, making it easier to hook up the Makey bots (Figure D). Even the youngest player can do it!

PLAY IT!

Dump the Makey bots in a pile. The person whose birthday comes next goes first. Choose any bot to hook another bot. Then add to your chain — but be careful! If a robot falls, your turn is over. Count the number of bots left in your chain. Then the person with the next birthday takes a turn. Whoever "makes" the longest chain of Makey bots is the winner!

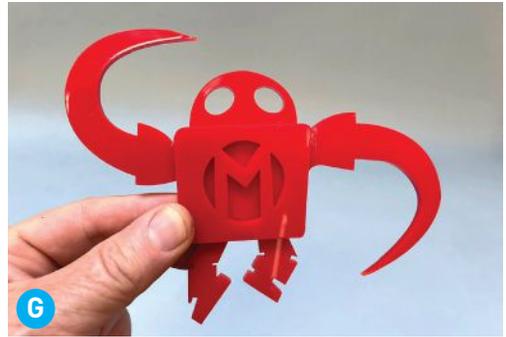


STRATEGIES

- Try to plan the sequence of Makeys on your chain: Leave the easier ones that are laying face-up for last, when your dangling chain is longer.
- Deferred gratification: Use your dangling chain to drag a hard-to-hook face-down Makey bot near the edge of the table first, then link to that easier, overhanging arm.

PLAY VARIATIONS

- Make a batch of Makey robots of various sizes. Make tiny ones for a tricky challenge that fits in your pocket (Figures **E** and **F**). Or make giant sized Makeys for massive fun! Use thicker material or double up on the body parts to be able to link with these bigger arm hooks (Figure **G**).
- Make a batch of Makeys in assorted colors (Figure **H**). Now instead of the letting the player choose which Makey to link next, the waiting players call out the color of the next bot to be linked. A much harder game!
- Makey Madness: A race against the clock! Time each player to see who can do it the quickest. If you drop a Makey — keep going! 🚫



HOW TO DETECT **SPY TECH**

Teardown: How the budget K18 Bug Detector works — and how to use it

Written and photographed by Tim Deagan



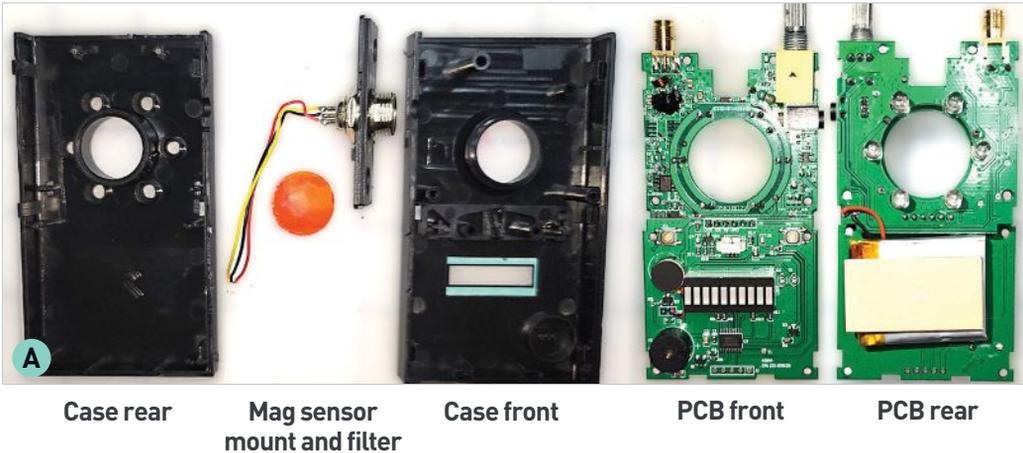
Can a budget bug detector really work? When you see a gadget selling for as low as \$18 (eBay) that claims to detect hidden cameras and wireless transmission devices between 1MHz and 6.5GHz, can it really do what it says? Surprisingly, with a few caveats, the answer is ... Yes!

Sold under a variety of names, this detector, which combines radio frequency (RF) detection, magnetic field detection, and hidden camera detection, is generally known as the **K18**. While clearly mass-produced with the cheapest possible parts, this device can successfully detect most of the commonly available wireless listening devices and hidden cameras, if used properly.

But using the K18 properly relies on understanding how the devices that it can detect work and how it tries to find them. In this article, we'll dig into principles of operation and methods of detection to gain that understanding.



TIM DEAGAN makes, breaks, and collects things in Austin, Texas. He loves the experience of using software to breathe life into hardware, using any microcontroller he can get his hands on.



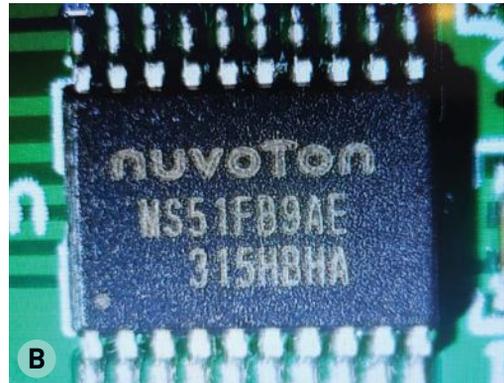
WHAT'S INSIDE THE K18?

The K18 is an exercise in min/max design. Maximum functionality from minimal, or at least minimal-cost, parts. Under the hood, there aren't any overly specialized components (Figure **A**). Some basic power management, a few resistors, diodes, capacitors, a coil, some buttons, a potentiometer, some LEDs, a Li-ion battery, a speaker, and a pleasant surprise!

Before doing a teardown, I imagined that some piece of custom silicon had been developed and widely shared to handle all the various sensor and output functions. Chinese manufacturers have a way of sharing IP among themselves that [Bunnie Huang](#) has termed *gongkai*, a similar but not identical concept to Open Source (if you're interested, read his writeup at [bunniestudios.com/blog/2014/from-gongkai-to-open-source](#)).

But, to my delight, the chip I found at the heart of the device took me back to my beginnings in microcontroller development in the late 1980s. Looking up at me from the circuit board was a modern revision of a 44-year-old design. My first digital love, the mighty 8051.

The MCS-51, commonly referred to as the 8051 ([wikipedia.org/wiki/MCS-51](#)), was introduced in 1980 by Intel and has been in continuous production (with enhancements) ever since. The version in the K18, the MS51FB9AE, is produced by Nuvoton (Figure **B**). While it sports new features like onboard I²C and SPI serial communications, PWM generation, and an analog



to digital converter (ADC), the code I wrote in 1989 would still run on this lovely little workhorse.

In the end, this makes tremendous sense: a commodity 8-bit microcontroller using a few discrete parts and some software instead of expensive custom silicon to read an internal ADC, a converter for radio frequency (RF) power levels, and a simple magnetic field detector (more on that below), then light up an LED bar graph and beep a piezo speaker. Simple, robust, effective, and cost effective.

This circuit design also allows the MS51 ADC, which has a max sample rate of 500ksp/s (kilosamples per second) to measure the power of signals operating at 6GHz, far faster than it could convert directly. By sampling the voltage level produced by the diode circuit rather than the signal itself, the speed of the microcontroller is not a limitation.

K18 CAPABILITIES	Microphones	Cameras (may have microphones)	GPS Trackers
K18 Can Detect	Transmits using radio between 1MHz and 6.5GHz	Transmits using radio between 1MHz and 6.5GHz or is retroreflective	Transmits using radio between 1MHz and 6.5GHz or has a magnetic mount
K18 Cannot Detect	Wired to remote recording or listening station	Wired to a remote recording or viewing station and is not retroreflective	Stores GPS data internally for later retrieval and does not use a magnetic mount
	Stores audio internally for later retrieval	Stores audio/video internally for later retrieval and is not retroreflective	Stores GPS data internally for later retrieval and does not use a magnetic mount

C

WHICH BUGS CAN (AND CAN'T) YOU DETECT?

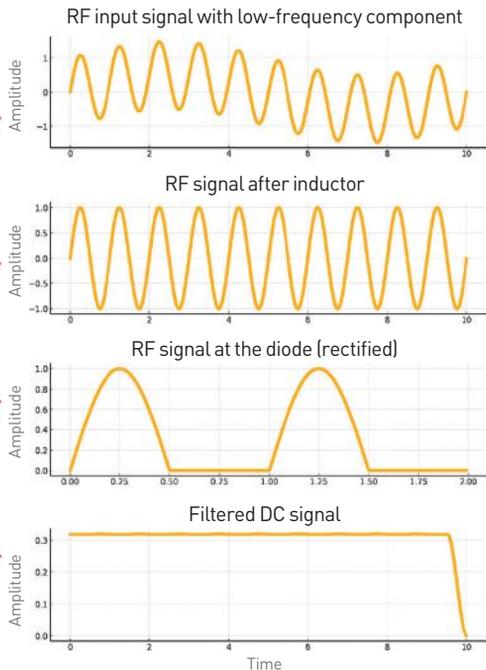
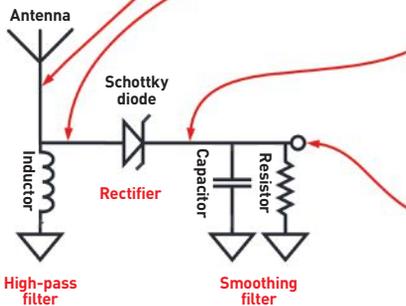
Unless someone is spending big dollars to use custom or expensive high-end surveillance gear, the kinds of spy tech that most people would encounter are **GPS trackers, listening devices, and hidden cameras.** Subsets of these are what the K18 is designed to find. If the device is attached with a magnet (such as a GPS tracker hidden in a car's wheel well) the magnetic field sensor could be used to find it. If the device is broadcasting to a remote location, the RF detector is useful. If the device uses a

retroreflective CCD sensor or lens, the hidden camera tool comes into play. However, wired devices, or devices that store audio (or video) internally without broadcasting, do not fall within the detection capabilities of the K18 (Figure C).

DETECTING RADIO FREQUENCY TRANSMISSIONS

The K18 responds to RF emissions between 1MHz and 6.5GHz. This is a huge range of frequencies that covers everything from the top of the AM radio band through much of the 5G cellular frequencies. While the K18 can detect

D Changes in signal characteristics in a diode-based RF power detector



RF energy within its range, it can't distinguish between different frequencies, protocols, or modulation types. In other words, it can't tell a ham radio transceiver from a Bluetooth headset. It only determines the overall level of RF energy in much the same way that an audio decibel meter tells you the sound level, but doesn't tell you anything about which frequencies, such as bass or treble, are producing it.

Unless you're in a Faraday cage, the world around us is constantly awash with RF signals from everything from Wi-Fi to FM radio stations to satellite transmissions, so the K18 set to its highest sensitivity will always show RF activity. Dialing the sensitivity down, or **attenuating**, allows it to find peaks of RF energy. The more RF activity there is in the environment, the more attenuation is required. This results in a need to be closer to the emitting device, so that the energy level of the signal is stronger due to the inverse-square law (see sidebar), to find it.

The technique used by the K18 is **RF peak power detection**, using a diode to convert the power level of the RF signal into a voltage level that the MS51 reads with its ADC (Figure D). Any undesirable low-frequency components of the signal, e.g. 60Hz from power lines, are filtered out with a low-pass inductor filter. That output is sent through a Schottky diode that cuts out all the negative parts of the wave, aka **rectification**. A resistor-capacitor (RC) smoothing filter then converts that set of resulting positive pulses into a clean DC voltage level by charging the capacitor during the positive peaks so that it discharges during the null sections and evens out the signal.

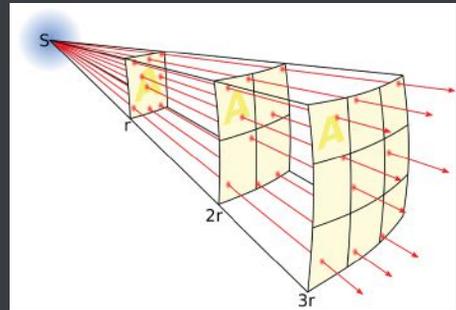
Determining what kind of device is emitting RF requires much more time and more complex (and expensive) equipment, such as a spectrum analyzer (see my article "Getting Started with Software Defined Radio" in *Make*: Volume 84, makezine.com/go/gsw-sdr). Even then it's a challenge, as Wi-Fi and Bluetooth devices are mixing their traffic with other devices, and many devices utilize frequency hopping, various digital encoding schemes, and even encryption. In this regard, the minimalist approach taken by the K18 is a reasonable compromise.

So what can the K18 detect using its RF sensor? Surprisingly, it covers the most common

INVERSE-SQUARE LAW

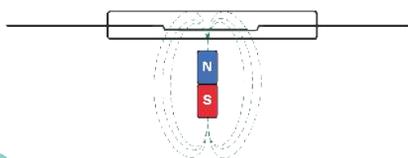
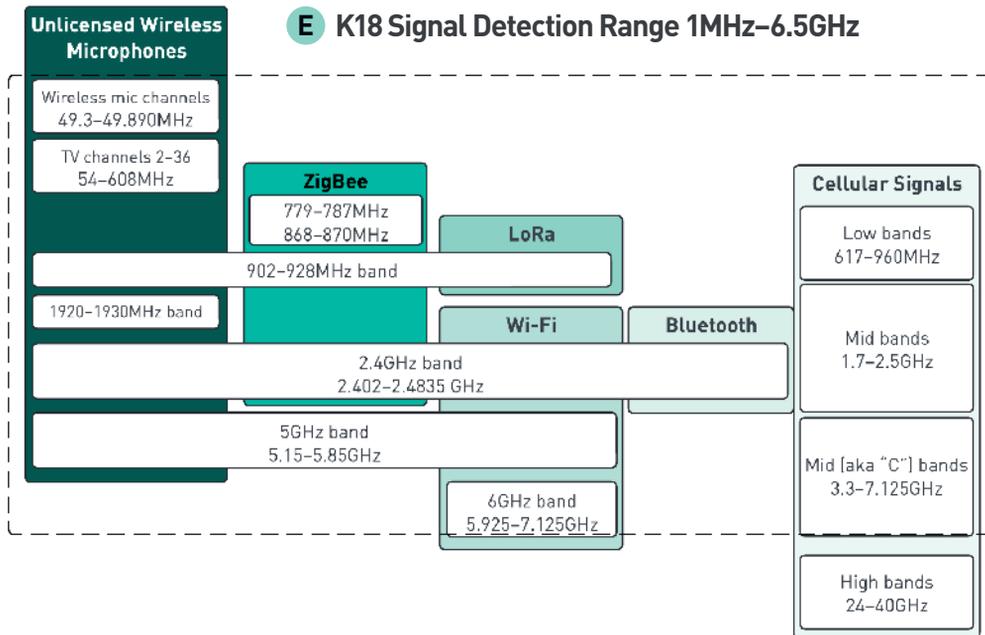
When devices such as radio transmitters or light bulbs emit energy, the intensity decreases as you get farther away. This decrease happens in a very predictable way known as the *inverse-square law*, which states that the intensity is inversely proportional to the square of the distance. "Inversely proportional" means that the bigger the distance, the lower the intensity. The "square of the distance" means the farther away you are from the source, the more area you have to cover. This can be written as:

$$\text{intensity} = 1/\text{distance}^2$$



Imagine you're chewing bubble gum and blowing a bubble. The bigger the bubble, the thinner the wall becomes as it stretches to cover more area. Similarly, radio signals expand as they radiate from an antenna. That's why the signal gets weaker as you get farther away — and stronger as you get closer to the source.

E K18 Signal Detection Range 1MHz–6.5GHz



F Magnetic field present, switch closed

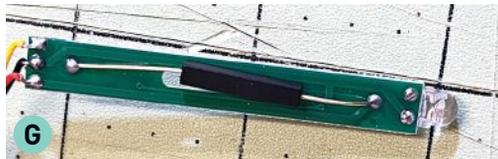
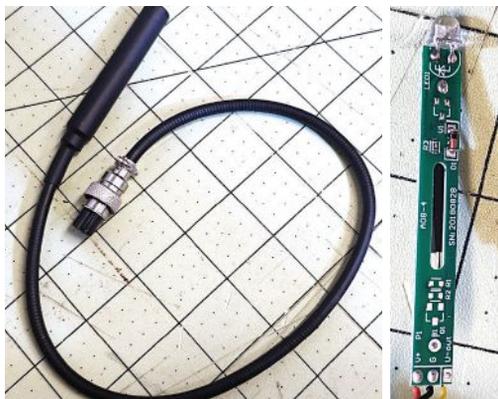
parts of the radio spectrum used by commercial wireless surveillance devices (Figure **E**).

DETECTING MAGNETIC FIELDS

At first glance, the magnetic sensor on the K18 may appear pretty useless. But it actually serves an interesting role. Many devices, such as GPS trackers, don't broadcast RF, they only receive signals from satellites. They serve as data loggers, storing the information on-device, and are physically retrieved rather than remotely accessed. An RF detector is useless for finding them. These devices are often hidden on the chassis of vehicles to track their drivers and include a magnetic base for easy mounting and removal.

There are three generally used methods for sensing a magnetic field. **Hall effect** sensors change voltage in the presence of a magnetic field. **Magnetoresistive** sensors change resistance. But the simplest and cheapest method is a **reed switch** (Figure **F**).

When looking at low-cost commodity gadgets like the K18, it's generally a safe bet that they're using the cheapest tech available. For magnetic field detection, this is definitely the case. The sensor at the end of the K18 gooseneck is a reed switch (Figure **G**) — two very closely positioned



thin metal leaves that bend and touch each other when a magnetic field is very close.

I found that the sensor had to be within 1 inch of a 10cm neodymium magnet to detect it. There is no sensitivity adjustment, the leaves of the switch are either touching or they aren't. A small LED on the end of the gooseneck lights when a field is detected. As an extremely scientific test to verify that the sensor is a reed switch, I whacked the end of the gooseneck soundly (on the theory that a brief mechanical insult would bounce the leaves of the switch together) and saw a brief flash of the LED.

DETECTING RETROREFLECTION

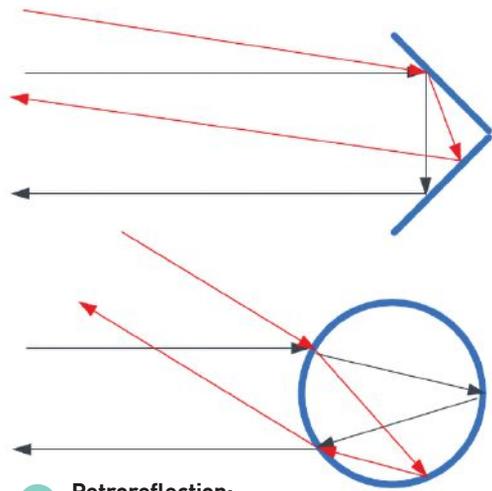
A **retroreflector** is a device that reflects radiation directly back at the source, regardless of the angle at which the radiation, such as light, hits it (Figure H). Even folks with no interest in optical geometry have generally seen a cat's eye shine with an eerie glow when reflecting car headlights or a flashlight. This is because the back of the spherical feline eye has a (retro)reflective layer of tissue called the tapetum lucidum. The world's greatest amateur scientist, Forrest M. Mims III (forrestmims.org), wrote an excellent article in *Make*: Volume 35 (makezine.com/projects/how-to-make-and-use-retroreflectors) on making your own retroreflectors.

The K18 has a ring of red LEDs surrounding a viewport with a red light filter (Figure I). Any red light viewed through the port has a higher contrast than normal, making it easier to spot. By observing through the ring of LEDs it's easier to stay in line with the light reflected directly back at the LEDs, helping you see retroreflectors — such as common digital camera sensors.

USING THE K18 ANTI-SPY DETECTOR RF DETECTION

To begin using the K18, attach the straight wideband "rubber duck" antenna to the SMA connector. This is the gold-colored threaded extension labeled ANT. The unit may require charging; if so, use the provided charger to top it off before going further.

Power up the K18 by turning the top knob, starting it up in RF detection mode. Full clockwise provides maximum sensitivity, lighting up the



H Retroreflection: Regardless of the angle the light enters, it is reflected back in the same direction.



I

entire bar graph and eliciting a beep. This doesn't mean you've detected a bug, it just means there are RF emissions around you. As noted above, unless you're in an extremely remote place or a Faraday cage, this will almost always be the case.

Slowly turn the sensitivity knob counterclockwise until it stops beeping. Now begin the hunt! Slowly walk around the space



you're investigating, gently waving the K18 near bookshelves, vents, appliances, or any other place a bug might be hiding. If you hear a beep, something is emitting nearby (Figure 1). You can often determine which direction it's in by slowly moving the K18 about.

The provided antenna is not particularly directional; it more or less receives well from all directions. However, it does receive better from the sides of the antenna than from the top or bottom. You'll typically need to go through a repeated sequence of slightly reducing the sensitivity and narrowing your search pattern until you zero in on the device's location.

HIDDEN CAMERA DETECTION

When hunting for hidden cameras, press the right-hand button on the front of the case labeled "GS." This lights up the ring of LEDs on the back of the case. That's really all the K18 itself will do

for this task. The rest is up to the Mark I human eyeball.

Place the K18 up to your face so that one eye is looking through the central viewport. This operation works best in a dark room so that there are no other sources of reflected light. Slowly scan the area for a brighter-than-average spot of light (Figure J). It's likely to be very small given the size of modern digital camera sensors.

Cheaply available hidden cameras are small enough to be concealed in air vents, clocks, stuffed animals, fake electrical outlets, garden decorations, smoke detectors, scent dispensers, plants, picture frames, or almost anything else in an office, hotel room, or house. Not every reflection will be a camera, and it can take some practice to spot well-hidden devices.

Like all modes of the K18, slow, steady, and close are the key techniques to use it successfully. The broader the field of view of a hidden camera, the wider the area its sensor will retroreflect back at you. Field of view varies, so careful movement to spot the bright spot reflection (usually less than 5mm across) is needed. Cameras may also be positioned behind a plastic front, as in the clock-based camera shown here. Moving around so that the reflection of the plastic front doesn't overwhelm the CCD reflection is helpful.

A bit of thought about what the person who hid the camera is trying to see will help narrow down the places to look. Starting your search from the most likely targets — doors, beds, etc. — is a good strategy.

MAGNETIC FIELD DETECTION

Devices like the K18 are frequently represented as effective for detecting GPS trackers. This isn't necessarily untrue, but it's definitely a bit of marketing hype. Granted, if the GPS tracker is broadcasting its data via cell signal, Wi-Fi, or Bluetooth, the RF detector has the potential to find it. However, many of these devices go into sleep mode when not in motion to preserve battery life. Some act only as data loggers and don't broadcast at all.

One strategy for keeping tabs on someone's location with a GPS tracker is to hide it in their car. Many models of GPS trackers have a built-in magnet, or offer a case with a magnet, to



make discreet placement on a vehicle (and later retrieval) easy. If that's the case, the magnetic field detector on the K18 offers a way to search for them.

After powering up the K18, press and hold the left-hand button labeled "MS." This puts the device into magnetic field detection mode. If the gooseneck sensor isn't attached, plug it into the three-pin aircraft connector on top and secure it with the threaded coupling. The sensor is in the middle of the bulge at the top of the gooseneck (see Figure G above).

Since the device is a Yes/No type of detection, there is no sensitivity adjustment. Using the magnet sensor requires getting the sensor fairly close to the target. Even my 500lb-pull neodymium fishing magnet required me to be less than 3 inches away to detect it. Smaller magnets required 1 inch or less proximity. The

gooseneck does make it easier to probe inside wheel wells, bumpers, etc., but slow, careful passage as close as possible is the ticket for successfully locating a magnetically mounted device with the K18.

BUDGET SPY BUSTER

So, is the K18 worth the price for detecting surveillance devices? Within its constraints, the answer is yes! It's a bit of a blunt instrument, but careful usage can make up for that. There is a plethora of similarly priced devices on the market that are likely pretty close to identical under the hood, and Amazon has numerous devices at four times the price of the K18 that are unlikely to do very much more. High-end equipment in the thousands of dollars can do considerably more, but for small dollars, the potential for a bit more peace of mind is a good investment. 🛡️



Written and photographed by Larry Bank

CHEAP YELLOW DISPLAYS

Get started with new **Arduino libraries** and example projects for these **inexpensive ESP32+LCD boards**



LARRY BANK is a software developer and maker with more than 30 years of experience in low-level coding and optimization. He grew up in New York during the rise of video games and 8-bit micros. After college, he moved around to different states and countries and is still going.

For 10 years I've been working with a wide variety of IoT hardware, mostly sourced from China, and I've been on a sort of quest to get to know the capabilities, price, and availability of every low-cost sensor, display, MCU, and camera. This has been useful for my work as a consultant and also for my personal projects. Since the purchases are usually out of pocket, I tend to focus on the lowest-priced hardware. After acquiring the devices, I try my best to write easy-to-use software to control them. This knowledge of the market, combined with the software I've written, allows me to quickly put together prototypes and proof-of-concept projects for clients or personal use.

For the past few years, Chinese companies have been selling hobbyist products that combine an Espressif Systems ESP32 microcontroller with a display, and maybe some sensors, buttons, a battery, etc. The list is growing, but three of the bigger players are M5Stack, Seeed Studio, and LilyGo. Of the three, M5Stack's offerings are more "product-like"; they come in injection-molded plastic cases and have a team of people working on software to support them.

But there's a new source on my radar now.

I WANT TO LOVE YOU (C.Y.D.)

About a year ago, I became aware of a new vendor selling ESP32+LCD products at a much lower price point. These so-called **Cheap Yellow Displays**, or **CYDs** for short, are gaining momentum in the market with their increasingly impressive hardware and low prices. (It was Brian Lough who coined the term, because of the yellow solder mask used on most of the boards.) The manufacturer seems to use two trade names for them: Sunton and Guition. They're sold primarily as **human-machine interfaces (HMIs)** and not targeted for individual makers or hobbyists. Apparently companies use them for control panels in other value-added products and lean heavily on the free LVGL graphics library to do it (github.com/lvgl).

There's one problem — when hobbyists buy CYDs, they only realize too late that there isn't really any software support for them. But if you can look past the lack of support from the manufacturer — and I'll help you do that in this article — there is some impressive hardware available.

Here's a table containing all the common variants I've seen (and own!):

PRODUCT NAME	MY NAME	LCD	TOUCH	MCU	LiPo
ESP32-2432S028	DISPLAY_CYD	2.8" ILI3941 240×320 SPI	R	ESP32	No
ESP32-2432S028 (with 2 USBs)	DISPLAY_CYD_2USB	2.8" ST7789 240×320 SPI	R	ESP32	No
ESP32-2432S024	DISPLAY_CYD_24R	2.4" ST7789 240×320 SPI	R + C	ESP32	Yes
ESP32-2432S022	DISPLAY_CYD_22C	2.2" ST7789 240×320 8-bit parallel	R + C	ESP32	Yes
JC4827W543	DISPLAY_CYD_543	4.3" NV3041A 480×272 QSPI	R + C	ESP32-S3 w/PSRAM	Yes
JC3636W518	DISPLAY_CYD_518	1.8" round 360×360 QSPI	C	ESP32-S3 w/PSRAM	No
ESP32-2424S012	DISPLAY_CYD_128	1.28" round 240×240 SPI	C	ESP32-C3	Yes
ESP32-3248S035	DISPLAY_CYD_35	3.5" ILI9488 320×480 SPI	C	ESP32	No
ESP32-8048S043	DISPLAY_CYD_8048	4.3" ST7701 800×480 RGB Panel 16-bit	C	ESP32-S3 w/PSRAM	No
ESP32-8048S050	DISPLAY_CYD_8048	5" ST7701 800×480 RGB Panel 16-bit	C	ESP32-S3 w/PSRAM	No

There are other variants of these PCBs (e.g. without any touchscreen controller), but these are the most commonly available models. The first one in the list is the “original” CYD; the second has the same product name, but is a very different board! An outlier is the 2.2” version — it has a fast, 8-bit parallel interface for the LCD. The QSPI displays can update the LCD the fastest.

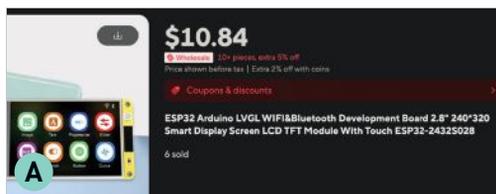
The “RGB Panel” displays have a very different interface to the MCU. They use the ESP32-S3’s PSRAM as a framebuffer. The pixels are continuously streamed to the display by the ESP32-S3’s LCD hardware component. This is easier to manage from software; all you have to do is write pixels directly into memory. The downside is that PSRAM limits the refresh speed to about 40FPS. Here’s a video comparing the same code running on a QSPI versus an RGB Panel display: youtu.be/jP287_Qq-Bo. The processor and GIF file are identical, but the QSPI display is able to redraw the GIF frames faster.

WHAT’S IN THE BOX?

CYD boards are usually shipped in a plastic snap-shut case to protect them from damage; I’m sure this is what saved a few of mine from destruction when the postal carrier squeezed them into my mailbox slot.

Also included in the case are a plastic stylus (for the resistive touch versions), a USB cable, and a 4-pin 1.25mm connector cable to fit one of the expansion connectors on the board.

The software and documentation are more difficult to find. The manufacturer has created the absolute minimum documentation and software necessary for expert users to use these products. It comes in the form of a large zip file which is downloaded from a Chinese file server. The link for this file may or may not be shared on the sales page you buy it from. This file contains example code, info about the GPIO connections and, for some PCBs, a complete schematic drawing.



WARNING: People who are new to embedded software or not familiar with the ESP32 will find the CYD manufacturer documentation cryptic and mostly useless. You may even feel that you’ve just purchased an expensive paperweight.

WHERE DO I GET IT?

New CYD devices normally are available first from Chinese vendors on the Alibaba and AliExpress marketplaces. These are a lot like eBay or Amazon in that small vendors can have a storefront and sell their goods. After a few weeks or months, the same products can be seen for sale on Amazon, eBay, and other local markets, but at a substantial markup. Here’s a good example (Figures **A** and **B**): From AliExpress, you can find it for \$10.84 with free shipping, while from Amazon, it’s nearly twice as expensive.

It’s the exact same product, but it probably passed through another middleman to get to that market. I’ve been buying electronic components and IoT products from AliExpress for several years now and they’ve improved their service quite a bit in that time. They now have a huge Cainiao customs clearing warehouse in Belgium for European customers, and for everywhere else their shipping has been streamlined too. Of course it’s up to you where to buy your gadgets; I’m simply showing you the options.

LESS FRUSTRATION, MORE FUN

As I described above, for my own work and projects it’s been useful to create software to support the myriad of devices I use. I’ve seen a lot of people buy these devices and immediately hit problems trying to use them. My latest efforts have been focused on making the various CYD products less frustrating by writing code libraries with consistent APIs across all the permutations of processors and LCD types. Feedback from the people using my CYD code has helped fix bugs



and add missing features, too. There are two libraries, both on GitHub and the Arduino library manager, and one collection of example projects.

The first library is **bb_spi_lcd** (github.com/bitbank2/bb_spi_lcd). It lets you talk to all the different CYD displays by using a simple constant name (e.g., **DISPLAY_CYD**) which contains the correct configuration (LCD type and connection pins) for your board. It also includes the resistive touch library with this same configuration. Here's an example of how simple this makes it to use on your CYD. No other software solution is as simple.

```
#include <bb_spi_lcd.h>
BB_SPI_LCD lcd;
setup() {
  lcd.begin(DISPLAY_CYD); // initialize the LCD controller
  lcd.rtiInit(); // initialize resistive touch
  lcd.fillScreen(TFT_BLACK);
  lcd.println("Hello World");
}
```

For the CYD boards with capacitive touch controllers, I wrote the **bb_captouch** library (github.com/bitbank2/bb_captouch). It auto-detects the capacitive touch controller if you just give it the GPIO numbers it uses for SDA/SCL/INT/RESET.

And finally, there's a collection of example Arduino projects (github.com/bitbank2/CYD_Projects) to show you how to use these libraries (and others I've written, e.g. my JPEG decoder and GIF players). Projects include a basic clock that also shows temperature, humidity, and air quality (Figure C), animating with RAM "sprites" (Figure D), an internet weather display (Figure E), and more. With these libraries and examples, you should be able to get your project working with a lot less frustration.

SOLUTIONS LOOKING FOR A PROBLEM

Based on the features of the CYD boards, I assume there is some specific target audience that needs those displays and connectors. Maybe they're just throwing darts at a wall, or there exists some kind of graphical control panel



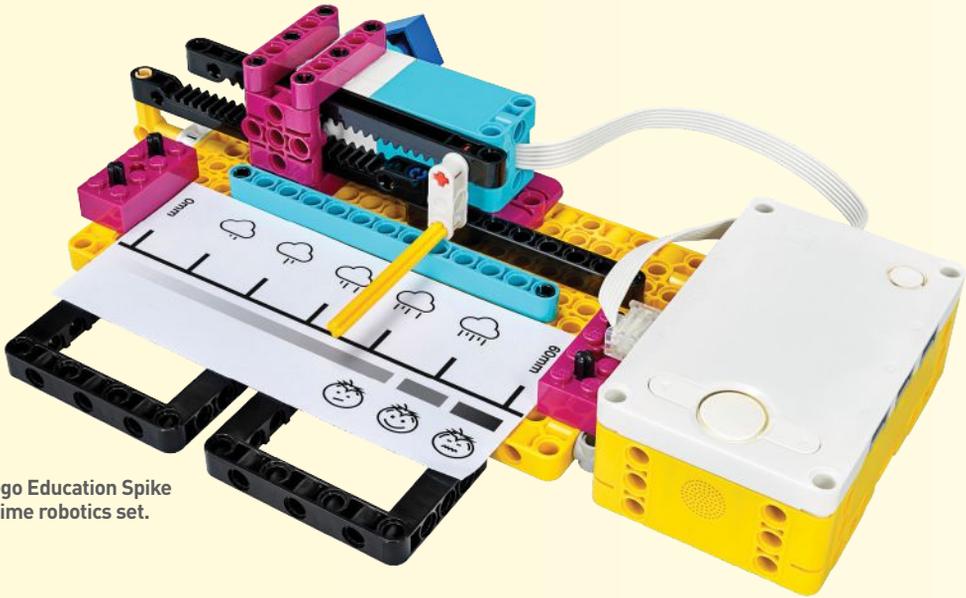
market with 4-wire serial port cabling. In any case, we're lucky that these particular boards happen to use the ESP32 and other common components. A fast RF-capable processor, a variety of display options, audio amplifier, microSD card slot, and sometimes lithium battery management — those features sound like the starting point for a lot of projects! 🎯



Try Larry's example Arduino projects for Cheap Yellow Displays, including JPEG display (from flash memory), IoT weather station (JSON parsing), Clock with temp, humidity, and CO₂ level (I²C sensors), GIF playback (from MicroSD card), and more. github.com/bitbank2/CYD_Projects

How to Choose Robot Kits for Kids

Making sense of billions of bots Written by Sam Freeman



Lego Education Spike Prime robotics set.

Are you looking to get your kid into STEM? Want to reward a child in your life with something fun and exciting that will surely guarantee a successful career in the engineering arts?

Robots are the most exciting STEM toys out there, but that's not exactly narrowing it down.

Robotics is a healthy blend of electronics, mechanical engineering, and programming. Kits are everywhere, but with 1,000+ to choose from on robotshop.com alone, how do you get the right one for the kid in your life? Here are a few ways to slice the offerings to help narrow down your selection.

Types of Experiments

Do you want to encourage playing with programming, or mechanical systems, or electronic circuits? The magic of robotics is the synergy of mixing all three disciplines together! Stirring these tastes into one bot soup is my

favorite thing about the medium, but finding one kit that teaches all three equally well is a tough order.

A lot of kits feature one subject well, and for younger kids that's a fine place to start. Some kits combine two subjects well, namely Lego or Jimu. These add complexity but open up more room for creativity. For instance, if your bot keeps knocking into things, you might try to build a longer whisker to activate a bumper sensor. Or you could try adjusting the speed or sensitivity in software. With multiple ways to tackle hurdles, there's room for nonlinear thinking and problem-solving.

I haven't found a perfect trifecta in one pre-packaged kit yet, so I suggest thinking about which subjects to focus on.

MECHANICAL

Lego Spike robotics kits are great at giving lots of easy mechanical experimentation and marrying it

with software possibilities. They probably take the gold here, but don't leave out **UBTech Jimu** sets if you're on a budget (see review, page 124). Getting the most out of either means picking up multiple sensors, motors, and a ton of construction blocks. But even a small starter kit can be put together in a bunch of ways, so creative kids will still find room to play with a smaller set. They might also beg for extra parts. (I am in no way trying to influence anyone to buy their kids all the things I wished for in my youth. That would be terrible.)

ELECTRONICS

The purest circuit-teaching robots are **BEAM bots** (see *Make*: Volumes 02, 06, 08, and 90). Simple and functional, these elegantly use the fewest components, sometimes with basic ICs, to get minimal functionality. On the plus side, they're relatively cheap, but they may seem more artsy and less thrilling for younger kids.

Arduino Alvik and similar kits get into the electronics play details — satisfy that electronics itch — but also offer multiple avenues into the world of programming.

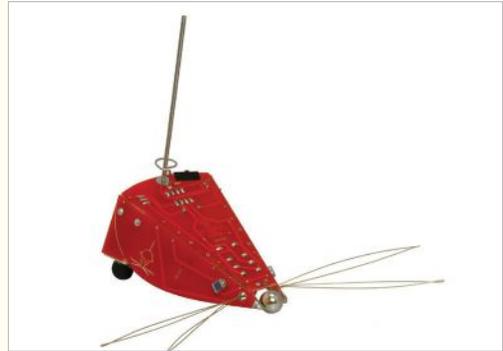
PROGRAMMING

Lots of kits come completely (or nearly) pre-assembled and serve as a platform for learning programming, from the cute **MakeBlock mTiny Smart Panda** for little kids to the powerful **XGO-Mini 2 Quadruped Robot Dog** for older students. This makes sense. It's easier to play with bits than atoms, and once you give somebody the keys to open the door in software, the possibilities explode compared to what you can do in hardware. If you want to try out a bunch of different arm lengths for a robot, you need to supply a kit with a bunch of hardware pieces for a bunch of different arm lengths. And that's just for one appendage. But if you want to try a thousand variables in software, just type in as many numbers as you can think of.

Personally, I think coding a robot might be the most engaging way to enter the world of programming. Seeing hardware move after a few simple lines of code is so much more rewarding than spitting out print statements on a screen.



Jimu Astrobot kit.



Herbie the Mousebot V2 kit from Solarbotics, available from makershed.com.



Arduino Alvik robot kit.



MakeBlock mTiny Panda robot.



XGO-Mini 2 quadruped robot dog.

I have yet to find an off-the-shelf package that teaches electronic experimentation, mechanical experimentation, and programming in perfect harmony. So if you need all three, maybe your best bet is getting multiple kits, or finding some DIY plans online, like **Otto** (ottodiy.com), featured in *Make*: Volume 61.

Types of Motion

WHEELED

The simplest mechanism to move something around is the wheel. It's been around a while. Wheeled robots are the easiest to start with, and pound for pound the fastest. If your kid's excited about FIRST Robotics, race cars, or BattleBots, check out the wheeled kits.

LEGGED

Trading speed for charm, legged form factors (including bipedal, quadrupedal, and hexapedal) are an interactive way to explore the field of robotics. The servos for each joint add up to a higher-cost kit than simple wheeled motors, and that cost goes up for each extra pair of legs. But if you've sensed an interest in building a robot Fido, they're worth a look. Names are mandatory on these bio-inspired bot buddies.

HUMANOID

I put humanoid robots in the third category because they add complicated things like arms and faces, and I like sets of things in three. Humanoid kits are probably the most impressive; who ultimately doesn't dream of building a robot person? But human bodies aren't the easiest designs to copy, and these kits are best enjoyed by those who really enjoy geeking out on technical stuff. It's a spendy way to get into the hobby but once you're there, there are whole clubs and games dedicated to things like wrestling, dancing, and soccer.

What Makes a Great Kit?

It should be fun — it won't be used for a minute if it isn't — and for the longest life it should be expandable. STEM is slapped on so many things that it's lost all meaning. Beyond the buzzword, what should you be looking for?

If you want to train a child to follow orders, get them a linear construction kit. If you want to help them solve problems and invent new things, get them something like building blocks or Snap Circuits. And if you want them to train themselves, get them something fun.

Learning happens through reckless



Otto DIY bipedal robot.



Peto's Bittle dog robot: open source, 8DOF quadruped platform (descended from Peto's Nybble cat robot in *Make: Volume 70*).



SmallKat cat robot: open source, 3D printed, 16DOF quadruped platform (descended from Worcester Polytechnic Institute).

experimentation. There's a saying about dialing in the failure ratio just right so that people are excited to play, but not guaranteeing success so much as to be pointless. It's a trick that makers of slot machines and successful mobile games know. And it's something to shoot for in a good kit. Does it have multiple ways to put it together? Does it have multiple bad ways to put it together? Does it have ways that sort of work and are sort of interesting that no adult would ever think of? Lego again is great at this, but so are MakeBlock, UBTech, and others that can be reconfigured and expanded.

Do You Have to Break the Bank?

What's the difference between a \$30 kit, a \$300 kit, and a \$1,000 kit? Generally, good kits cost \$100–\$400; below that it's harder to find kits with great features and quality. Closer to the \$1,000 range, you're looking at something that may be achievable only for a classroom, but it's going to be able to teach some advanced concepts.

For example, the **Bittle** by **Peto** is a charming pet robot with room for personality and expansion (as their recent OpenCat Robot Around the World Contest shows). It gets around cost constraints by being small and offering expandability through add-ons, but it's an affordable way to get started. Compare that to **SmallKat** which, with greater degrees of motion, can help someone experiment with coding closed-loop walking movements from

the ground up. It's great for someone who wants to geek out and put their personal touch into control routines, but it's priced so high that I don't expect to see it much outside a classroom.

What to Watch Out For

- **Solar energy** is important and exciting and makes for a dull robot kit. It's tempting to try and teach every lesson at once, and renewable energy is worth learning about. But size and cost constraints usually result in a kit that includes undersized panels. They take a long time to charge, limit the power available, and can take the fun out of the robot build.
- **Linear, step-by-step assembly** kits are satisfying, but not always educational. Personally, I love unwinding with a good laser-cut kit, and they're a great gift for a child who's interested in STEM topics. But I've seen the label slapped on piece-by-piece building kits promoting their "educational" value when they actually required less creative thought than a video game.

Parting Words

Ultimately, you know the child in your life better than I do. But whether they are artsy or mechanical, feel like getting their hands dirty or prefer looking at a screen, these are some ways you can try to reach out and engage them. 🍀



Jimu FireBot Robot Kit

\$66–\$70 amazon.com/dp/B07Q7QJ7L8

Jimu is a series of robot kits from the reputable brand UBTech; their FireBot looked like the best mix of entertainment value and low cost. With 606 pieces including three servomotors, these parts look and feel a lot like Lego, but are slightly odd-sized, so you can't swap one-to-one. The kit requires an app, and the animated instructions are helpful and clear, except for one servo connector that's backward (careful builders will be OK).

Finishing a build this size is satisfying, no matter your age. The assembly numbers are molded right into the parts, the power brick connects easily with JST-type cables, and the peg insertion tool is handy for connecting parts.

I put FireBot on the floor, paired Bluetooth, and scared the wits out of my dog. It's surprisingly zippy and its built-in actions are charming. Making my own action from scratch was tougher — my first attempts resulted in a scary chattering from a servo pushed too far.

The Blockly programming language is simple to grasp and includes variables and trig functions. Absolute beginners can follow the tutorials and write their own scripts.

Overall the Jimu FireBot is a great kit that's easily worth the cost. It suffers slightly from a weird UI and fragile-feeling servo cables, and includes only one touch sensor (other sensors are available if you combine kits). But it's a good build with room to grow, the presets are fun, and the programming mode leaves much to explore. —*Sam Freeman*



Zivko the Robot Kit

\$36–\$55 amazon.com/dp/B07NVDSJZ7

Elenco is an established brand with quality stuff like Snap Circuits. Their Zivko hexapod was the cheapest decent-looking robot kit out there, and I wanted to discover what the low end is.

Zivko's parts snap out of plastic frames, model airplane style. The gears are all plastic, but at this price I'm not surprised. In fact, I'm always impressed by the ingenuity of toy designers: Zivko moves six legs with just two motors.

The printed instructions are clear, and aside from one major confusion, assembly went smoothly. Less than 10 seconds after I turned it on, it stopped walking and made a grinding noise. Was it repairable? Thankfully yes. After hearing back from Elenco, it turns out I'd failed to tighten some screws in the walking assembly all the way.

The product page mentioned AI, but really there's just an IR sensor and two preset play modes: "Explore" to avoid obstacles, and "Follow" mode which was honestly pretty neat.

In the end, Zivko is a decent model kit that nets an amusing toy. Would I buy it again? Maybe if it's on sale. It was pretty simple to build and looks adorable. It's not a platform to grow and expand on, but while it doesn't provide much education, it's still entertaining enough that I think the nephews will get a kick out of playing with it. —*Sam Freeman*



Carvera Air cuts aluminum and other soft metals.



Example projects like this nautical plaque make it easy to learn the Air's controls.

Carvera Air Desktop CNC Mill

\$2,099 pre-sale makera.com/pages/carvera-air

The Carvera Air may not be the biggest or most powerful CNC, but if you put one on your desktop, you just might find it's one of your go-to tools.

This fully enclosed little CNC is only slightly larger than a desktop 3D printer but is capable of milling wood, plastics, and even metal. The optional laser module pushes the possibilities even further.

The original Carvera made waves when it brought an automatic tool changer (ATC) to the world of desktop CNC. The Air lowers the price by replacing the ATC with a manual quick-change lever that still makes multi-tool jobs a breeze.

Desktop CNCs often feel underpowered and not rigid enough for serious jobs. Carvera Air's die-cast metal frame and 20mm linear rods make for a strong, stiff machine. The 200-watt spindle cuts aluminum with ease. The front lid is a bit flimsy, which can interfere with the dust seals.

Makera makes it easy to get started by including example projects in various materials that help you use every aspect of the Air. I especially like the LED Light project that guides you through making your first PCB.

To control the Carvera Air, there's software for Windows, Mac, and Android. It could be more user friendly — some operations aren't entirely clear at first — but if you work through the example projects, you'll be up to speed. The Air uses a GRBL-based controller, so numerous CAM packages will work. Makera has their own CAM solution in beta, but also has configurations for Fusion 360 and VCarve. With its new features, VCarve 12 and the Air will be a winning pair.

Many desktop CNCs rely on just an enclosure to deal with dust, but the Air also includes built-in dust extraction (BYO vacuum). On my pre-production unit, most of the dust collection parts are 3D printed. With any luck, Makera will release the STL files, allowing users to print replacement parts or their own modifications.

The Air also has a probe to measure materials, a red dot laser to preview job extents, a 4th axis for near-3D rotary jobs, and optional 5W laser module for engraving. If you're in the market for a desktop CNC I can, with no reservations, recommend the Carvera Air. —*Matthew Stultz*

Matthew Stultz

UNLEASH YOUR INNER MAKER!

Whether you're 8 or 80, the Maker Shed offers an incredible array of **kits for learning about electronics, engineering, design, and coding**. So what are you waiting for? Dive in and discover a whole new universe of possibilities!

Make:

ARDUINO ELECTRONICS STARTER PACK

Learn How to Make Electronics That Interact with the Physical World

- Skill level: Beginner to Advanced
- Some soldering required
- Computer required

Books to Boost Your Experience:
Getting Started with Arduino
Learn Electronics with Arduino

Maker Shed

Make:
Getting Started with **ARDUINO**
4th Edition

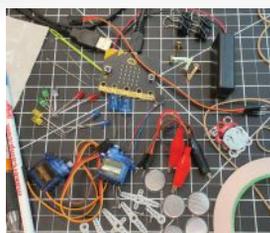
The Open Source Electronics Prototyping Platform
by Massimo Banzi co-founder of Arduino and Michael Shloh

The image shows a collection of electronic components including an Arduino Uno board, a breadboard, jumper wires, a red robot figure, a digital display, and various sensors. To the right is the book cover for 'Getting Started with Arduino' 4th Edition, which features a technical diagram of the Arduino board.

Make: Arduino Electronics Starter Pack & Getting Started with Arduino, 4th Edition book



Makey bit: Mobile Kit & Getting Started with the micro:bit book



Making Simple Robots Starter Pack with kit & Making Simple Robots book



FREE SHIPPING on orders \$25 & up with code **MAKEYKLAUS** at makershed.com/kits

Maker Shed

GIVE THE GIFT OF STEM EXPLORATION!

Quick and easy-to-build kits with everything you need to launch in one convenient box!



This year, ignite a passion for science, technology, engineering, and math (STEM) with Estes Rockets Mini Launch Sets! With these hands-on model rocket kits, kids can build and launch their own mini rockets, learn about physics and aerodynamics, and fuel their curiosity for science and engineering.

Order your Mini Launch Set at EstesRockets.com

Looking to buy wholesale?
Visit B2B.EstesRockets.com



PicoScope® 3000E

Next-Level, Mid-Level Oscilloscopes

- Bandwidth to 500 MHz
- Sampling rate to 5 GS/s
- Capture Memory to 2 GS
- 4 Analog channels
- 16 Digital inputs (MSO models)
- Hardware resolution to 10 bit (14 bit enhanced)



www.picotech.com

Unconventional art education for everyone



Build an electronic ensemble

DIY SYNTHS

DIY RHYTHM WIDGETS

EAR RE-TRAINING

SOUND FORAGER'S GUIDE

CASSETTE HACKING

MUSICAL AUTOMATOPHONES

PEDAL MACHINE MUSIC

All materials included

No experience needed

Financial aid available



Apply **MAKE20** for 20% off
Sign up at dogbot.com

OVER THE TOP



A KINDER ROBOT

In August, *Make*: Editor-in-Chief Keith Hammond and I went down to Silicon Valley and met with Christoph Kohstall, founder of the humanoid robotics startup Kind Humanoid. I came away from the meeting with the idea that we are entering a new wave of robotics, you might say, robots with AI for brains.

Kohstall had brought their first prototype, called Mona, to Maker Faire Bay Area last year, and it was a spectacular sight: a 6-foot tall robot without skin protected in a glass case. Mona was all wires and motors wrapped around orange 3D-printed body parts. I almost didn't care if it worked; it was a work of art!

Kind Humanoid says the robot is "combining the power to reason with the ability to act." It moves AI into a physical body that occupies the same space that we do. In their workshop, the physical components of a robot and the processes for making them are familiar and within reach of any maker — racks of 3D printers printing out body parts for the skeleton, an electronics workbench, and stacks of specialized servomotors from China sitting in boxes waiting to be connected and tested. It looks a lot like a garage makerspace.

Kohstall and his team are working on a new version called Olympia. Olympia is sleek, with the technical insides enclosed by a 3D-printed hard body. For more updates, follow Kohstall at kindhumanoid.com or x.com/CKohstall.

—Dale Dougherty

Kind Humanoid



WOODCRAFT®

MIX. POUR. CREATE.
Your source for resin casting supplies.
COUNT ON US FROM START TO FINISH®

Scan
to be
inspired.



We carry
brands
makers love.
woodcraft.com



XTOOL

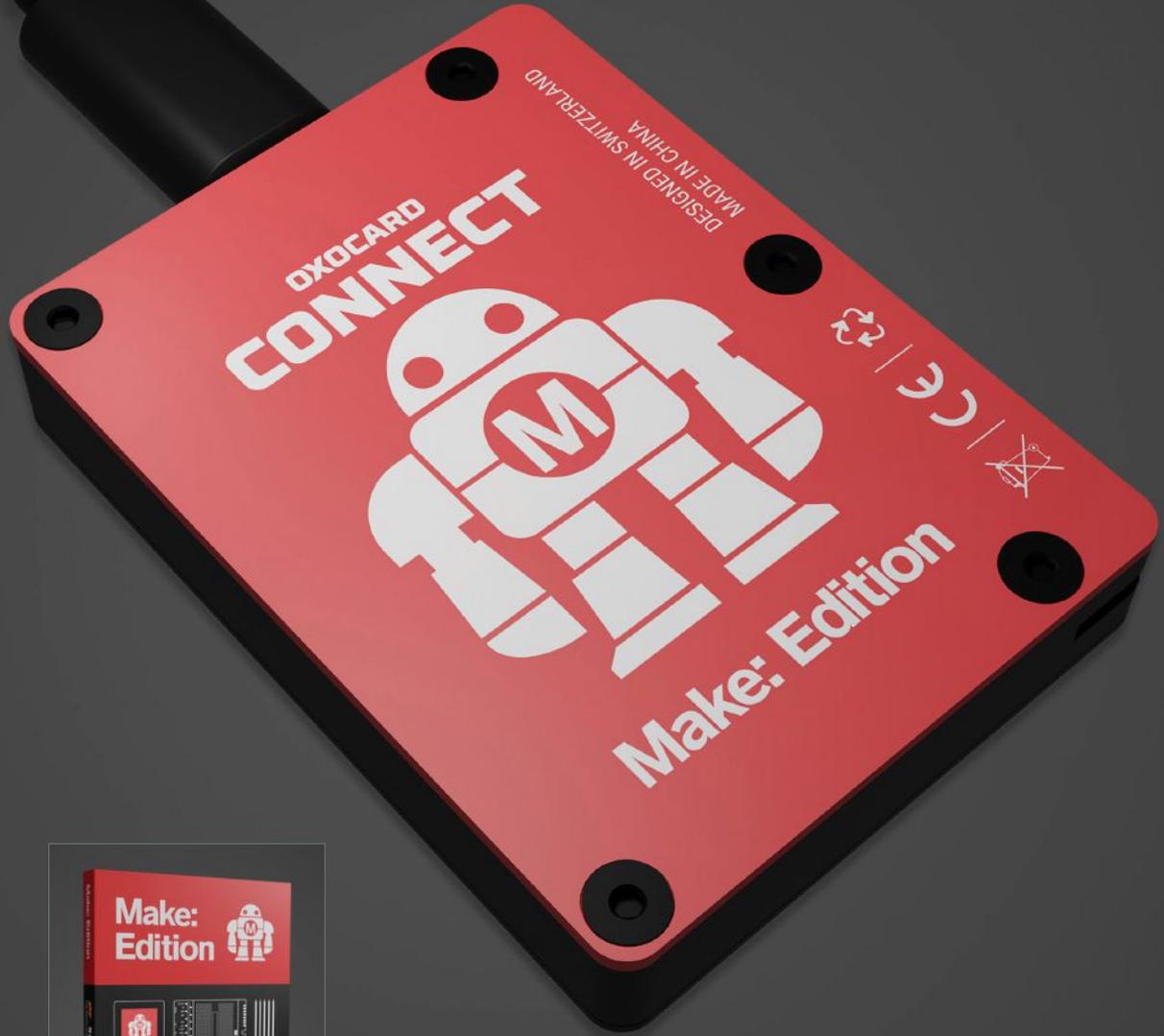


SHAPE

FESTOOL



INNOVATION IS WITHIN YOUR GRASP!



LEARN MORE



Make: OXOCARD

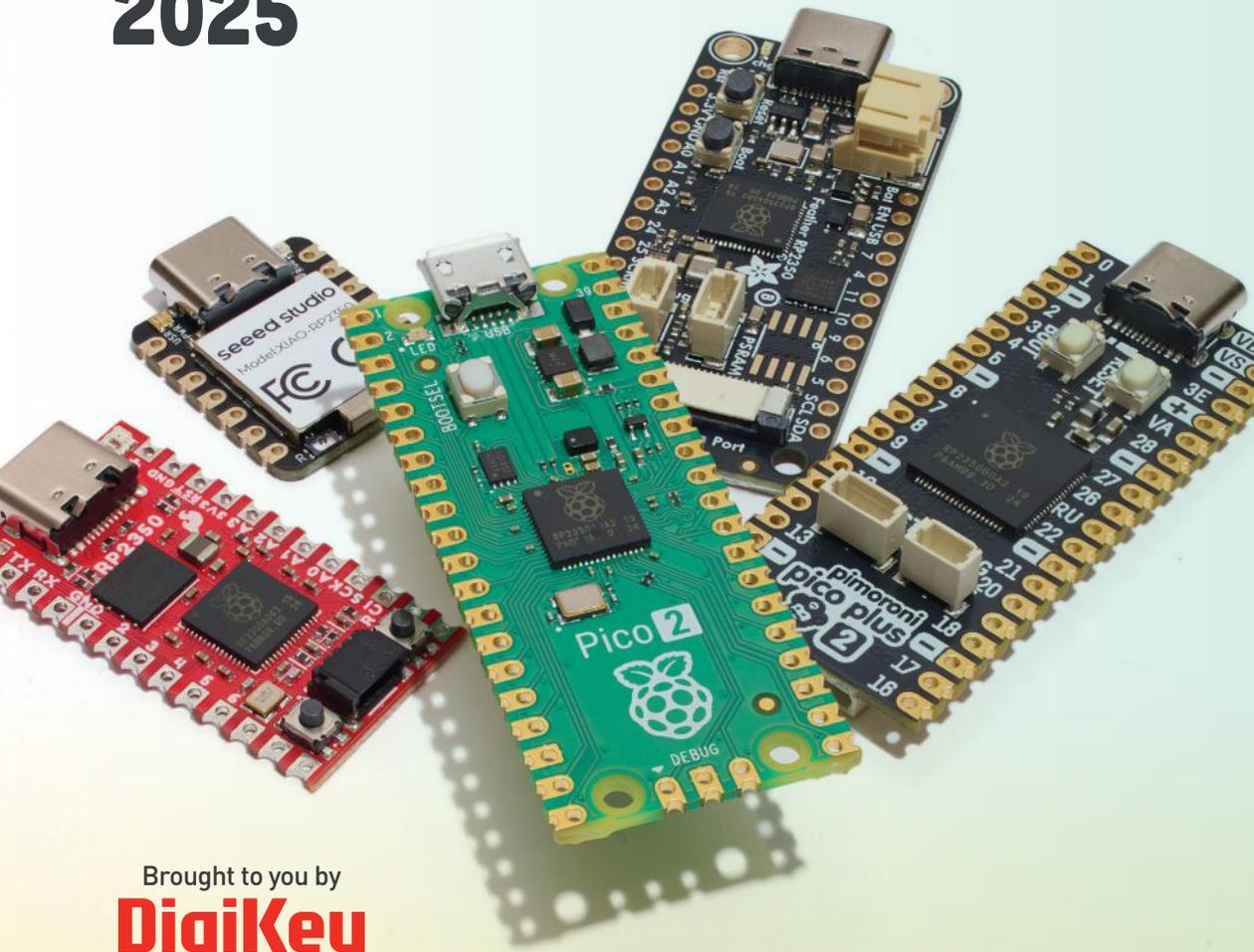
make.co/innovate

Make:

THE ORIGINAL

GUIDE TO BOARDS

2025



Brought to you by

DigiKey

BOARD ALMIGHTY!

2023 was a great year for boards, with both Raspberry Pi and Arduino launching new flagships in the form of the Pi 5 and Uno R4 respectively. Supply chain issues had largely been solved and we found ourselves in a Cambrian explosion of microcontrollers and single-board computers, thanks to abundant resources including the RP2040 and ESP32, and the sustained availability of basic integrated circuits and passive components whose scarcity had been stunting the growth of the domain.

2024 saw the continuation of that evolution, with manufacturers developing highly specialized species of boards for particular AI, LED animation, IoT, datalogging, and other tasks. Raspberry Pi's next

generation of silicon, the RP2350 microcontroller, finds itself on our cover, in the form of the Pico 2 and simultaneously-launched partner boards, but beyond this exciting launch, there are no blockbuster flagships like a Pi 6 or Uno R5. As such, this year's guide is an opportunity to feature some of the more exotic and unique offerings that we've seen this year, as well as boards that are launching soon. —David Groom



Explore these and countless other boards with DigiKey's **Innovation Hub Boards Explorer** immersive web experience. [innovationhub.digikey.com/?target=boards-explorer](https://www.digikey.com/?target=boards-explorer)

THIS YEAR'S MOST PRAISE-WORTHY BOARDS

1 BEAGLEBOARD BEAGLEY-AI

One of the first exciting new releases of 2024 was the open-source AI-focused BeagleY-AI from BeagleBoard. A departure from their typical “cape” headers, the BeagleY adopts a Pi-like form factor, with quad 1.4GHz 64-bit Arm Cortex-A53 cores, a powerful GPU supporting three simultaneous displays, and a built-in 4 TOPS AI co-processor.

2 SEED STUDIO SENSECAP WATCHER

As the artificial intelligence ecosystem continues to mature, more application-specific hardware is beginning to emerge. Perhaps the most striking example of this trend is Seeed's open-source physical AI agent, Watcher. This device combines local AI processing via its Himax WiseEye2 HX6538 chip with cloud or on-premise assistance for more complicated tasks, allowing you to, for example, request notifications upon the detected presence of people wearing delivery uniforms by the built-in camera.

3 USEFUL SENSORS PERSON SENSOR

Another AI-in-a-box solution is the Person Sensor from Useful Sensors (not to be confused with their

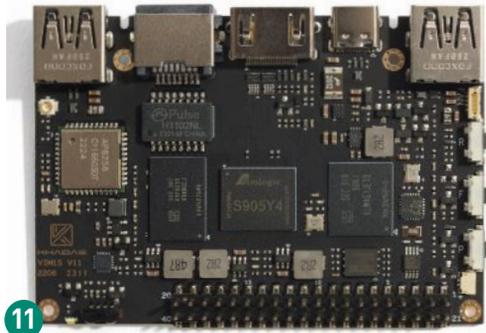
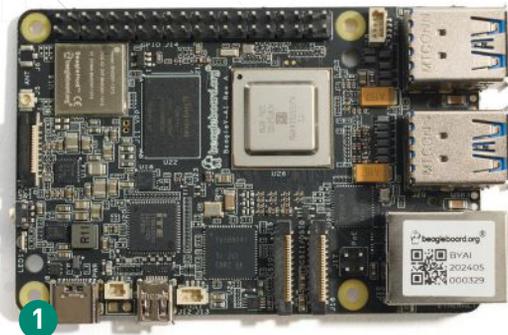
product literally named “AI in a Box”). This thumbnail-sized board uses its built-in camera and smarts to detect and identify faces, without requiring users to obtain a degree in machine learning. In addition to making it simple to use, the abstracted black-box functionality also helps ensure privacy.

4 PIMORONI PLASMA 2350

This issue's cover focuses on RP2350-based boards, and Pimoroni are a leading adopter of RPi's new silicon. Their Plasma board is a USB-C powered controller for addressable RGB LED strips, with screw terminals for easy attachment, a Qwiic connector for convenient expansion, and 3A of juice to power fun glowies like their optional 10-meter LED star strip.

5 OPENMV CAM RT1062

OpenMV are the “Arduino of Machine Vision” and their latest-gen smart camera board features the same pioneering MicroPython-based implementation with an updated 600Mhz i.MX RT1060 Arm Cortex-M7 and built-in Wi-Fi. Compatibility with over a dozen existing shields and a modular camera system makes theirs the ideal board for a range of AI projects.



Mark Madeo

6 ARDUCAM PINSIGHT

Another leader in embedded vision, Arducam have been pumping out a multitude of industry-ready yet maker-accessible products since 2012. The PiNSIGHT fuses a 4K camera plus hardware-based encoding and AI acceleration with your existing Raspberry Pi 5 to create an “intelligent vision hub” with extensive available pre-trained computer vision models supported by great documentation and examples.

7 SEED STUDIO GROVE VISION AI MODULE V2

On the smaller side, Seeed’s Grove Vision AI v2 Kit bundles an AI-optimized Arm Cortex-M55/Ethos-U55-based board with an optional OV5647 Camera Module and Xiao ESP32-C3 to create an approachable machine vision solution leveraging no-code SenseCraft pre-trained models. An extensive wiki and examples help makers take things to the next level with their own custom AI solutions.

8 CROWPANEL ESP32 1.28" ROUND DISPLAY

Elecrow have been cranking out a series of ESP32 + display combos under the CrowPanel moniker, and it was hard to pick just one, but in the end this stunning round capacitive touch model won out. Ideal for wearables, home automation, and IoT projects, it also supports the fantastic LVGL graphics library.

9 M5STACK DIAL

Another spin on the ESP32 + 1.28" round touchscreen formula, this time from the incredibly innovative M5Stack. What makes this one stand out is the rotary dial that surrounds the screen, providing a unique and charming user input method, which combined with an RFID reader and integrated M5StampS3 makes it great for home automation projects where temperature or light levels require convenient adjustment, as well as user identification and authorization.

SPECS: FEATURED MICROCONTROLLERS AND SINGLE BOARD COMPUTERS

Board Name	Price	Dimensions	Software	Clock Speed	Processor
BeagleBoard BeagleY-AI	\$72	85mm×56mm×20mm	Debian 12.5	1.4GHz	64-bit quad-core Arm Cortex-A53
Seeed Studio SenseCAP Watcher	\$99	64mm×68mm×19.5mm	SenseCraft	240MHz	32-bit ESP32-S3 (dual-core Tensilica LX7)
Useful Sensors Person Sensor	\$10	19mm×16mm	Arduino IDE, Python, CircuitPython, C/C++ via I ² C Qwiic connector	Not specified	Not specified
Pimoroni Plasma 2350	\$13	60mm×22mm	MicroPython, CircuitPython, C/C++	150MHz	Dual Arm Cortex M33
OpenMV Cam RT1062	\$120	45mm×36mm×29mm	MicroPython	600MHz	Cortex-M7 CPU w/ Double Precision FPU
Arducam PiNSIGHT	\$100	88.5mm×58mm×10mm	OpenVINO, OpenCV, Caffe, TensorFlow, MXNet	700MHz	Intel Movidius Myriad X
Seeed Studio Grove Vision AI Module V2	\$29	40mm×20mm×10mm	Arduino IDE, TensorFlow, PyTorch	400MHz	Dual-core Arm Cortex-M55 and Arm Ethos-U55
CrowPanel ESP32 1.28" Round Display	\$14	42mm×42mm×9.8mm (round)	Arduino IDE, ESP-IDF, Lua RTOS, MicroPython, LVGL	160MHz	32-bit ESP32-C3 (single-core RISC-V)
M5Stack Dial	\$35	45mm×45mm×32.3mm (round)	Arduino, UIFlow, ESP-IDF	240MHz	32-bit ESP32-S3 (dual-core Tensilica LX7)
DFRobot LattePanda Mu	\$189	69.6mm×60mm (module only)	Windows 11, Ubuntu 22.04, Debian 12.5	3.4GHz	64-bit quad-core Intel N100
Khadas VIM1S	\$59	82mm×58mm×13mm	Ubuntu 22.04, Armbian, CoreELEC, Android 11	2.0GHz	64-bit quad-core Cortex-A35
Particle Tachyon	\$249	85mm×56mm×13.5mm	Ubuntu 24.04, Yocto	2.7GHz	64-bit octo-core Qualcomm Kryo CPU

10 DFRobot LattePanda Mu

The LattePanda Mu is DFRobot's take on the compute modules offered by the likes of Raspberry Pi. Unlike the Arm-powered Pi CM4, however, it is based on the quad-core 3.4GHz Intel N100, with claimed performance of more than double the latest Pi 5, as well as 8GB of LPDDR5 RAM and 64GB of integrated eMMC 5.1 storage. An open-source carrier implementation turns it into a tiny PC motherboard, complete with PCIe slots, perfect for your next Windows or Linux-based project.

11 Khadas VIM1S

We featured the VIM4 in an earlier Guide, and now we're excited to see Khadas grace these pages again with the VIM1S. Its combination of compact size, yet full-size ports and built-in storage (saving you a bundle on specialty cables and accessories), plus great graphical performance and Oowow OS delivery service

make it stand out in the single board computer (SBC) space.

12 Particle Tachyon

A surprise new entry in the field, Particle's Tachyon is the 5G AI IoT SBC nobody saw coming. With a Pi-like form factor, an octa-core Qualcomm Kryo CPU, 12 TOPs of AI acceleration, and the Particle IoT platform behind it, this Ubuntu-based cellular-connected do-everything board will be ready to tackle just about anything when it launches in early 2025.



DAVID J. GROOM loves writing code you can touch. If he's not hacking on wearables, he's building a companion bot, growing his extensive collection of dev boards, or hacking on 90s DOS-based palmtops. Find him on Mastodon at @ishotjr@chaos.social

Memory	Radio	Video	Ethernet on Board	Input Voltage	Battery Connection	Operating Voltage	Type
4GB LPDDR4	Wi-Fi 6, Bluetooth 5.4 with Bluetooth Low Energy	microHDMI, OLDI (LVDS), MIPI-DSI	✓	5V, 3A	—	5V (3.3V I/O)	SBC
8MB flash, 520kB SRAM, 8MB PSRAM	Wi-Fi, Bluetooth Low Energy	1.45" round 412x412 display, 2952x1944 (5MP) camera	—	5V	—	3.3V	SBC
Not specified	—	Not specified	—	3.3V	—	3.3V	MCU
520KB SRAM, 4MB QSPI flash	—	—	—	3V–5.5V	—	3.3V	MCU
1MB of SRAM, 32MB SDRAM, 16MB flash	—	2952x1944 (5MP) camera	—	3.6V–5V	✓	3.3V	MCU
16MB flash	—	4056x3040 (12MP) camera	—	5V, 1A	—	3.3V	SBC
2432KB RAM	2.4GHz Wi-Fi, Bluetooth Low Energy 5.0 (via attached XIAO ESP32C3)	2952x1944 (5MP) camera	—	3.3V	—	3.3V	MCU
400kB SRAM, 8kB SRAM (RTC)	2.4 GHz Wi-Fi (802.11b/g/n) and Bluetooth 5	1.28" round 240x240 display	—	5V	—	3.3V	MCU
8MB flash, 520kB SRAM, 8MB PSRAM	Wi-Fi, Bluetooth Low Energy	1.28" round 240x240 display	—	6V–36V	✓	5V	MCU
8GB LPDDR5, 64GB eMMC 5.1 storage	—	HDMI 2.0 4k60, eDP 1.4b	✓ (carrier)	9V–20V	—	3.3V	MCU
2GB LPDDR4, 16GB eMMC 5.0, 32MB SPI flash	Wi-Fi 5, Bluetooth 5.0	HDMI 2.1 4k60	✓	5V	—	3.3V	SBC
4GB LPDDR4X, 64GB Flash	Sub-GHz 5G — CAT-18 cellular, GPS / Galileo / GLONASS / BeiDou, Wi-Fi 6, BT 5.2	USB-C 4k60, MIPI-DSI	—	5V	✓	3.3V	SBC

AI AT THE EDGE

Written by Shawn Hymel

1 RASPBERRY PI 5

CPU: Broadcom BCM2712 quad-core 64-bit Arm Cortex-A76 @ 2.4GHz

GPU: VideoCore VII @ 1GHz

RAM: 2, 4, or 8GB

Flash: Your own SD card or SSD

An accessible, low-cost single-board computer ideal for edge AI applications. Its powerful CPU can run full machine learning frameworks like PyTorch and TensorFlow, and supports GPU acceleration for inference (not training). Object detection (YOLOv8n) for images with 640×640 resolution runs about 5 FPS, and small LLMs (Llama8:3B) can generate responses at about 2 tokens per second. USB and PCIe slots allow for add-on AI acceleration hardware, which can dramatically increase speeds

2 NVIDIA JETSON ORIN NANO

CPU: 6-core 64-bit Arm Cortex-A78AE @ 1.5GHz

GPU: 512-core (16 tensor core) or 1024-core (32 tensor core) Nvidia Ampere @ 625MHz

RAM: 4 or 8GB

Flash: Your own SD card or SSD

Very powerful SBC with integrated Nvidia GPU for fast inference. Object detection (YOLOv8n) for 640×640 images runs about 30 FPS, and small LLMs (Llama8:3B) can respond at about 4 tokens per second. Training is also feasible, but slow. Much more expensive (\$500+) than the Pi, and the Nvidia-specific frameworks present a longer learning curve. Also has USB and PCIe slots for acceleration hardware.

3 SEED STUDIO XIAO ESP32S3 SENSE

CPU: Xtensa dual-core 32-bit LX7 @ 240MHz

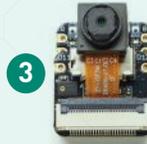
RAM: 8MB

Flash: 8MB

Hardware optimizations: SIMD, DMA, FPU

Software optimizations: ESP-DL, ESP-DSP

Compact, powerful stack of boards featuring a camera, mic, Wi-Fi/BLE radio, and AI acceleration for small tasks such as audio, vibration, and image classification. It can't handle full object detection (e.g. YOLOv8n), but can use Edge Impulse's FOMO model for constrained object detection at about 8 FPS, 96×96 resolution.



4 RASPBERRY PI PICO 2

CPU: Arm dual-core 32-bit Cortex-M33 @ 150MHz

RAM: 520kB

Flash: 4MB

Hardware optimizations: SIMD, DMA, FPU

Software optimizations: CMSIS-DSP, CMSIS-NN

The new RP2350 microcontroller is a big upgrade from the RP2040. Its dual-core Cortex-M33 offers special hardware optimizations like SIMD, DMA, and an FPU, for processing time-series data and inference tasks such as audio, vibration, and image classification. While it lacks the ESP32-S3's connectivity, the Pico 2 is cheaper and should perform similarly well.

OFF-THE-SHELF EDGE AI BOARDS

These work as intelligent, stand-alone sensors that can be connected to a microcontroller or SBC, with pre-trained machine learning models for special tasks.

Useful Sensors **Person Sensor** (see page 2) looks for faces and sends a notification via I2C. **Seed Studio SenseCAP Watcher** (page 2) looks for a pre-defined object, keyword, or gesture, then connects to an internet LLM, like a tiny, hackable Amazon Echo Show. **Seed Studio SenseCAP A1101** is an industrial camera with image classification and object detection, in an IP66 waterproof case. 🗨️

DAVE'S FAVES

Shawn's picks match my AI go-to list precisely! Here are some rarer gems:

- AI-in-a-box:* DFRobot's **HuskyLens** vision sensor and **Gravity** offline speech recognition module, Seed's **Grove Smart IR Gesture Sensor** and new **ReSpeaker Lite**, and Arducam's **Pivstation 5** and **KingKong** for Pi 5 and CM4 machine vision.
- Pi power:* **Raspberry Pi AI Kit** straps 13 TOPS of neural-net acceleration to your Pi 5; new **Raspberry Pi AI Camera** with Sony IMX500 sensor frees your CPU for other tasks. —DG

EDUCATION

Written by Joan Horvath and Rich Cameron

Need boards for classroom use? In the past you'd choose an **Arduino** or a **Raspberry Pi**, and your students' (in)ability to solder would constrain your choice of projects. Those are still good options — but now there are boards that pack a lot more functionality with a lot less wiring required.

Two of the most common boards in K-12 classrooms are the **Adafruit Circuit Playground Express**, aka "CPX" (learn.adafruit.com/adafruit-circuit-playground-express), and the **BBC micro:bit v2** (microbit.org), both available in \$30 starter kits including the board, USB cable, battery pack, and batteries. Let's compare.

CODING

Both boards are programmed via web browser in **Microsoft MakeCode**, a block-based environment similar to Scratch. **Python** and **JavaScript** are options, with drag-and-drop code blocks to ease the transition to text coding. In micro:bit's version (makecode.microbit.org) code is saved in the browser, and can be downloaded to laptop but not edited locally. (The **Micro:bit Classroom** (classroom.microbit.org) lets you share code with students and capture theirs, with no accounts required.) In Adafruit's version (makecode.adafruit.com) the code is stored on the CPX but can also be uploaded to computer for editing.

For text coding, CPX is compatible with the **Arduino IDE** and **CircuitPython**; micro:bit can also use Arduino with some setup shenanigans. Code.org has also created JavaScript exercises called "CS Discoveries" for the CPX, which require overwriting its firmware.

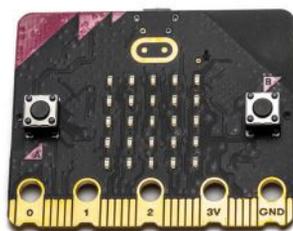
FEATURES

Both boards carry similar **sensors** — sound (mic), temperature, light, touch, and accelerometer — plus a mini speaker. Micro:bit adds a compass (magnetometer) handy for absolute orientation; CPX adds IR proximity sensing. The major differences are:

Available pins — The CPX wins by far: 14 pins with eight GPIO pins. The micro:bit exposes just three GPIOs plus 3V and ground; there are 17 more GPIOs but to access them you'll need an expansion board. If you hope to add more than one extra sensor, lights, etc., look to the CPX. Or look into **expansion kits** that

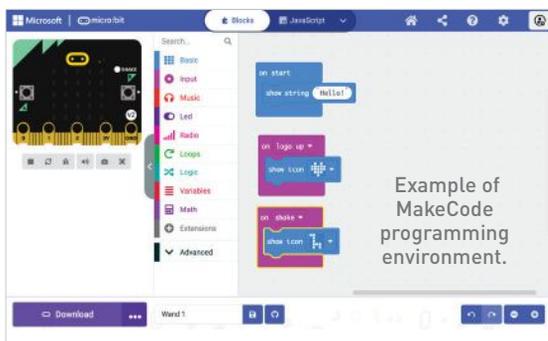


Adafruit Circuit Playground Express



Micro:bit v2

Rich Cameron



allow more connectors, like the **Adafruit Crickit** or the **Makey:bit Adventure Board** for micro:bit.

LED displays — The micro:bit's 5x5 red LED matrix and easy pattern interface make it best for displaying simple information. The CPX has a ring of ten RGB NeoPixels that can be coded to any color, for more sophisticated fashion or art applications.

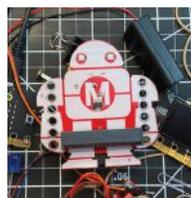
Board-to-board communication — The micro:bit has a Bluetooth radio, with block code for sending text or triggering commands on other micro:bits. This opens up all kinds of games and more serious pursuits. The CPX has a much more limited IR interface (though students love it for "laser tag").

BOTTOM LINE

The two boards are reasonably equal unless you require one of their unique features. The micro:bit feels more friendly, but more limited for add-ons. The CPX is better for LED projects, but with more of an Arduino feel and setup, and a bit longer learning curve. Both have big communities and many projects and tutorials online to help your students succeed. 🏆



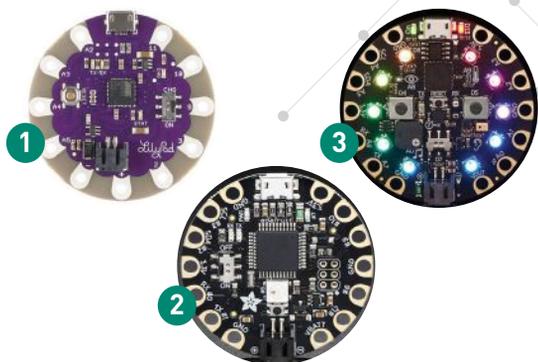
Adafruit Crickit



Makey:bit

WEARABLES

Written by Kate Hartman



1 LILYPAD ARDUINO USB

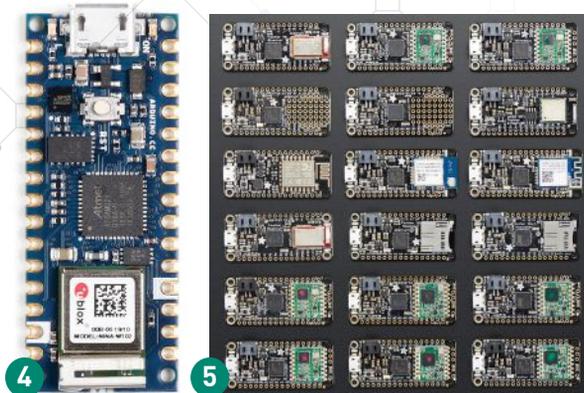
LilyPad Arduino is the original flavor of sewable circuit boards, developed by Dr. Leah Buechley and SparkFun Electronics. Since 2008, the original LilyPad design has inspired many other circular, hand-sewable boards. LilyPad Arduino USB is the latest version, featuring an ATmega32U4 microcontroller chip with built-in USB support, an on/off switch, and a JST connector for LiPo batteries.

2 ADAFRUIT FLORA

Flora V3 is Adafruit's dedicated wearable electronics board, and the smallest full-featured one around. Flora has the same ATmega32U4 chip with USB support, plus beginner-proof power management that accepts a versatile range of 3.5V to 16V DC. A full-color NeoPixel LED is perfect for providing visual feedback when you're debugging your wearable on the go.

3 ADAFRUIT CIRCUIT PLAYGROUND EXPRESS

CPX is the PCB equivalent of a Mary Poppins bag: Just when you think you've explored all the features on this board, you discover there are more! While not intended for wearables, it's currently by far my favorite all-in-one board for wearables and e-textiles. Packed with on-board sensors and actuators, its powerful ATSAMD21 Arm Cortex-M0 processor can be programmed in (at least) three different ways (see page 7). Connections aren't limited to conductive thread; other methods demonstrated in the Adafruit Learning System include a bolt-on kit that doesn't require any soldering!



4 ARDUINO NANO FAMILY

New Arduino Nano boards are released almost every year, and the newer ones have become favorites for classroom kits involving wireless communication. The Arduino Nano 33 IoT has been a go-to in recent years, featuring the ATSAMD21 processor, both Wi-Fi and BLE, and an on-board accelerometer/IMU for motion tracking. For wearables, the headerless version works well in small spaces and it can be powered by a power bank via USB.

5 ADAFRUIT FEATHER FAMILY

With their extensive variety of both main boards and accessory "wing" boards, working with Feathers feels like building a circuit with a PCB card deck of superpowers. Processor choices include ATmega32u4, M0, M4, ESP32, RP2040, and the new RP2350; radios include Wi-Fi, Bluetooth, GSM cellular, LoRa, and more. Short stacking headers, doubler/tripler prototyping boards, and Stemma QT connectors are all options for packing a lot of functionality into a small, robust package, perfect for fitting in a case or pocket of a wearable. 🦋

DAVE'S FAVES

Adafruit's **Circuit Playground Express (CPX)** is featured as a go-to for both education *and* wearables, so I'll point you to her similarly SAMD21-based quarter-dollar-sized sister, **Gemma M0**.

Along with the Seeed **Xiao** and Adafruit **QT Py** range, DFRobot's diminutive **Beetle ESP32 C3** is a great choice when size counts, and comes with a fantastic little expansion board for petite prototyping. —DG

LEDs

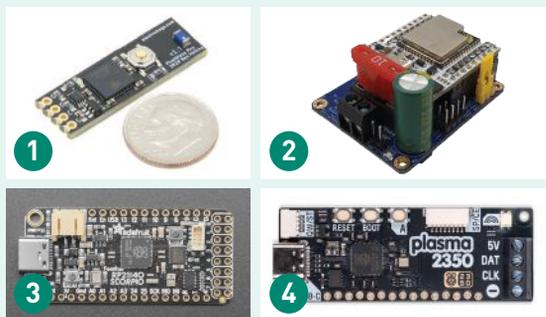
Written by Debra Ansell

A specialized control board can really expedite your addressable LED project. Features like signal level shifting and reverse polarity protection reduce your component count, while pattern-managing software makes animations easy. Which board's best? Depends on how many LEDs you're driving, their power requirements, and sensor integration. These boards offer many features to streamline your illuminated marvel. For more detail and a comparison matrix, go to makezine.com/go/boards-leds.

PRE-PROGRAMMED BOARDS

1 BEYOND THE MATRIX: The ESP32-based **Pixelblaze V3 Standard** (\$39) and smaller **V3 Pico** (\$35) run proprietary browser-based software with a JavaScript pattern editor that updates LED animations in real time. Patterns from a library can be arranged into playlists, and because pixel-mapping is separate from pattern code, the same animations display consistently on any arbitrary arrangement of pixels in 3D space. The Pico has a 6-axis IMU, while the Standard takes an add-on board with IMU, microphone, and light sensor. For big jobs, an output expander (sold separately) for the Standard supports multiple strips in parallel, and a simple software setting synchronizes multiple Pixelblaze controllers. These boards work with many addressable LED types, beyond the typical NeoPixel/DotStar variants, for a flexible, expandable, and powerful LED control system.

2 PERFECTLY POWERED: ESP32-based **QuinLED Dig2Go** (\$27) and **Dig-Uno** (\$33) provide simple solutions for managing a project's patterns and power. Both come pre-flashed with free, open-source **WLED** software containing mix-and-match color palettes and LED animations that can be arranged into playlists via WLED's browser-based interface. Both provide substantial regulated 5V power directly through the board (3A for Dig2Go, 15A for Dig-Uno), and can cut power to the LEDs to save energy when patterns are not displayed. The Dig-Uno connects easily to 3- and 4-pin LED strips via screw terminals. The Dig2Go has a mic for sound reactivity and a sturdy enclosure with 3-pin JST LED output and USB-C power input



terminals, making it a great choice for NeoPixel wearables.

DIY CODEABLE BOARDS

3 PLENTY OF PIXELS IN PARALLEL: The **Adafruit Feather RP2040 Scorpio** board provides eight pairs of level-shifted signal pins with ground pins to easily control eight LED strips at once. Combined with Adafruit's code library (C++ or CircuitPython), the RP2040's PIO processors can quickly push LED signal data to eight NeoPixel strips simultaneously while freeing the main processor for other tasks. As part of Adafruit's Feather ecosystem, the Scorpio is compatible with dozens of add-on boards, and while you must code your own LED patterns, helpful examples on Adafruit's website will get your project running quickly.

4 PERFECT FOR PROTOTYPING: **Pimoroni Plasma** boards come in RP2040, RP2040W, and RP2350 varieties (see page 2) and feature screw terminals with two level-shifted data pins for quickly reusable connections to 3-pin and 4-pin LED strips. The USB-C connector provides up to 3A of current at 5V to the LEDs. Two user buttons and lots of I/O pins make it easy to connect peripherals, and the speedy RP2040 and RP2350 processors have plenty of power to run intensive computations (C/C++ or MicroPython) while still animating the LEDs. 🔧

IoT

Written by David J. Groom

I wasn't sure if there would be an IoT category this year. Connectivity is so ubiquitous since the rise in popularity of system-on-chips like the ESP32 (and the ESP8266 SoC before it) that Wi-Fi and Bluetooth Low Energy (BLE) almost feel like table stakes. But thanks to the continued innovation of industry leaders like Particle, Blues, and SparkFun, we have a lot to talk about!

One of the first "IoT" companies I remember, **Particle** (or Spark as they were known when I backed their previous Kickstarter campaign) is also one of the ones I'm most excited about in 2024. I'm going to skip right past their current crowdfunding campaign for the Particle Tachyon (see page 5), and instead focus on their **Muon** dev board and **M-SoM** module. Particle's fourth-gen system-on-module offers Wi-Fi and cellular connectivity as with prior offerings, plus low-power LoRaWAN, but what has me really excited is the Non-Terrestrial Networks (NTN) satellite variant, which brings connectivity that was out of reach to makers just a few years ago. Coupled with the Muon, which transforms the module into a full-featured dev board, featuring a Qwiic connector and Raspberry Pi-compatible headers for easy expansion, I've been having a blast with this new hardware on a familiar platform.

I am a huge fan of **Blues'** approach to IoT: their developer-first APIs and dev tools enable anyone with a loose command of JSON to dive into connected devices, with the only required hardware skills typically being plugging in a USB cable. And while Particle have announced their NTN M-SoM for release in late 2024, Blues have already shipped their **Starnote** satellite failover solution, which mimics the form factor of their cellular and Wi-Fi modules but is actually an accessory to them, requiring a special **Notecarrier** dev board in order to accommodate a **Notecard** buddy as well. My favorite feature of Blues' products is that bandwidth is included, so for \$49 you get a fixed chunk of satellite data for life with no subscriptions or monthly minimums.

IoT, datalogging, and tracking often go hand-in-hand, and nowhere is this better exemplified than in **SparkFun's Artemis Global Tracker**. They put an



Particle Muon dev board and M-SoM satellite module



Blues Starnote Non-Terrestrial Networks module



SparkFun Artemis Global Tracker



Adafruit ESP32-S3 Reverse TFT Feather



DFRobot FireBeetle 2 ESP32-C6

Artemis low-power SoM, Iridium 9603N Short Burst Data modem, u-blox ZOE-M8Q GNSS receiver, and TE MS8607 pressure, humidity, and temperature (PHT) sensor into one bonkers board, so that you can transmit data via satellite while you obtain location data via ... a different satellite, even if you're in the middle of a polar expedition. Looking to roll your own? SparkFun's IoT-optimized **Thing Plus** ESP32-C6 and -S3 boards combine their nominal ESP SoCs with USB-C, Qwiic and JST battery connectors.

Another datalogger I just think is neat is **Adafruit's** adorably-named microSD-equipped **Feather RP2040 Adalogger**, and I'm also enjoying their ESP32-S3 **Reverse TFT Feather**, which sticks a really great little screen on the "back" of the board for IoT dashboards and data display. Both are compatible with the Arduino IDE and Adafruit's beginner-friendly CircuitPython programming language.

DFRobot has some interesting, inexpensive ESP32-based boards as well, in the form of their **FireBeetle 2** series. The ESP32-C6-based version in particular features Wi-Fi 6, Bluetooth 5, Zigbee 3.0, Thread 1.3, and Matter, making it ideal for home automation projects as well — all for under \$6!

Other boards based on the RISC-V C6 include the tiny **M5Stack NanoC6 Dev Kit** and **Seeed Xiao ESP32C6**, both under \$6 and as at home in home automation as IoT projects. I'm also loving **DPTTechnics' Walter** 5G tracker, but I'm running out of room in this segment, so check out [makethings.make.co](https://www.makethings.make.co) for my review of this and tons of other boards! 🚀

BOARD HOARD

Written by David J. Groom

We hope you've enjoyed this year's guide; it was really difficult to confine ourselves to just AI, Education, Wearables, LEDs, and IoT. For example, I'd hoped to include categories like Home (**Arduino's** sole dev board release this year, the **Nano Matter**), Music (**Adafruit Music Maker FeatherWing**), Cosplay (**Adafruit RP2040 Prop-Maker Feather**), and Robotics (**Pimoroni Yukon**), maybe Displays for the new crop of microcontroller + display combos (I'm obsessed with **Soldered's Inkplate 6 Motion**), and even a look at boards suitable for Digital Fabrication (Adafruit's Arduino Mega-sized **Grand Central M4 Express** featuring the SAMD51). As a last resort, I've wrestled this last page back as a way to jam in more of the boards and dev kits that impressed me this year.

In lieu of a full music category, allow me to highlight the **BeagleBone**-based **Bela** platform. I'm having a blast with their **PocketBeagle**-powered **Mini Multichannel Expander Starter Kit**, which gives you eight channels of interactive audio to empower practically any new instrument or sound installation you might imagine. Coupled with an impressive range of accessories, such as their Gliss Eurorack touch sensor module, an incredible built-in web-based programming environment, and fantastic documentation and tutorials, Bela is ready to make your wildest sonic fantasies a reality. Looking for something simpler? Be sure to check out the **Teensy Audio Library** and **Audio System Design Tool** plus associated boards.

Some LED boards I've been enjoying include the **Adafruit Matrix Portal S3**, which makes it easy to create your own CircuitPython-powered LED signs and passive information displays, and the **Pimoroni Tiny FX LED Effects Controller Starter Kit**, which helps you shove tiny LED dots into models, dioramas, and best of all: Lego!

In the things-with-their-own-built-in-displays department, I'm enamored with the **M5Stack Cardputer**, which crams a 56-key qwerty keyboard and display into a credit-card-sized, magnetic, Lego-mountable form factor. I'm on a perennial search for the next best thing to my beloved Chumby, so the **ThingPulse ESP32 WiFi Color Display Kit Grande**



Bela Mini Multichannel Expander



Pimoroni Tiny FX LED Effects Controller Starter Kit



Adafruit Matrix Portal S3



M5Stack Cardputer

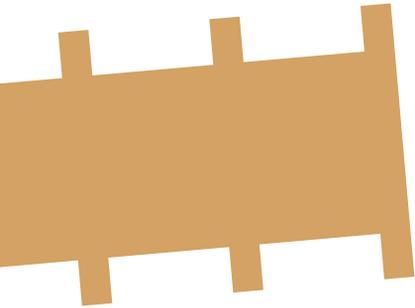
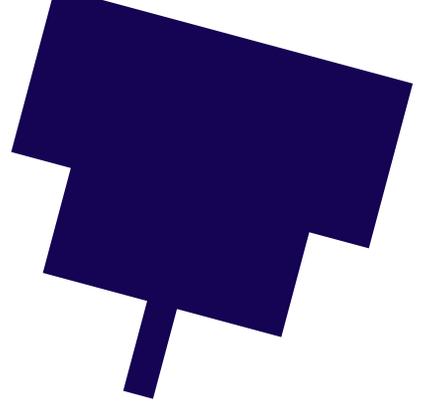


Raspberry Pi Bumper silicone base

caught my eye, as it pairs their **ePulse Feather ESP32** development board with a high-precision color capacitive touch display, making it ideal for tiny desktop dashboards. And **Adafruit's Funhouse** puts the home in home automation — or vice versa? — with its novel haunted house design and plethora of built-in sensors.

Finally, October sees the release of some great official ways to pimp your Pi, including the new portrait-oriented **Raspberry Pi Touch Display 2**, storage-boosting **SSD Kits**, and minimalist **Bumper** case.

That's it for another thrilling year of dev boards and dev-board-adjacent goodness! We hope you dig the new approach — be sure to let us know what you think, and we'll see you again soon with another comprehensive guide to all the new hardware released over the next 12 months! 🍷



Innovation demands selection

Let us help you engineer the future with millions of products from over 2,300 name-brand, industry-leading suppliers.

Find what you need at [digikey.com](https://www.digikey.com) or call 1.800.344.4539

DigiKey

we get technical

DigiKey is an authorized distributor for all supplier partners. New products added daily. DigiKey and DigiKey Electronics are registered trademarks of DigiKey Electronics in the U.S. and other countries. © 2024 DigiKey Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA

 **ECIA MEMBER**
Supporting The Authorized Channel