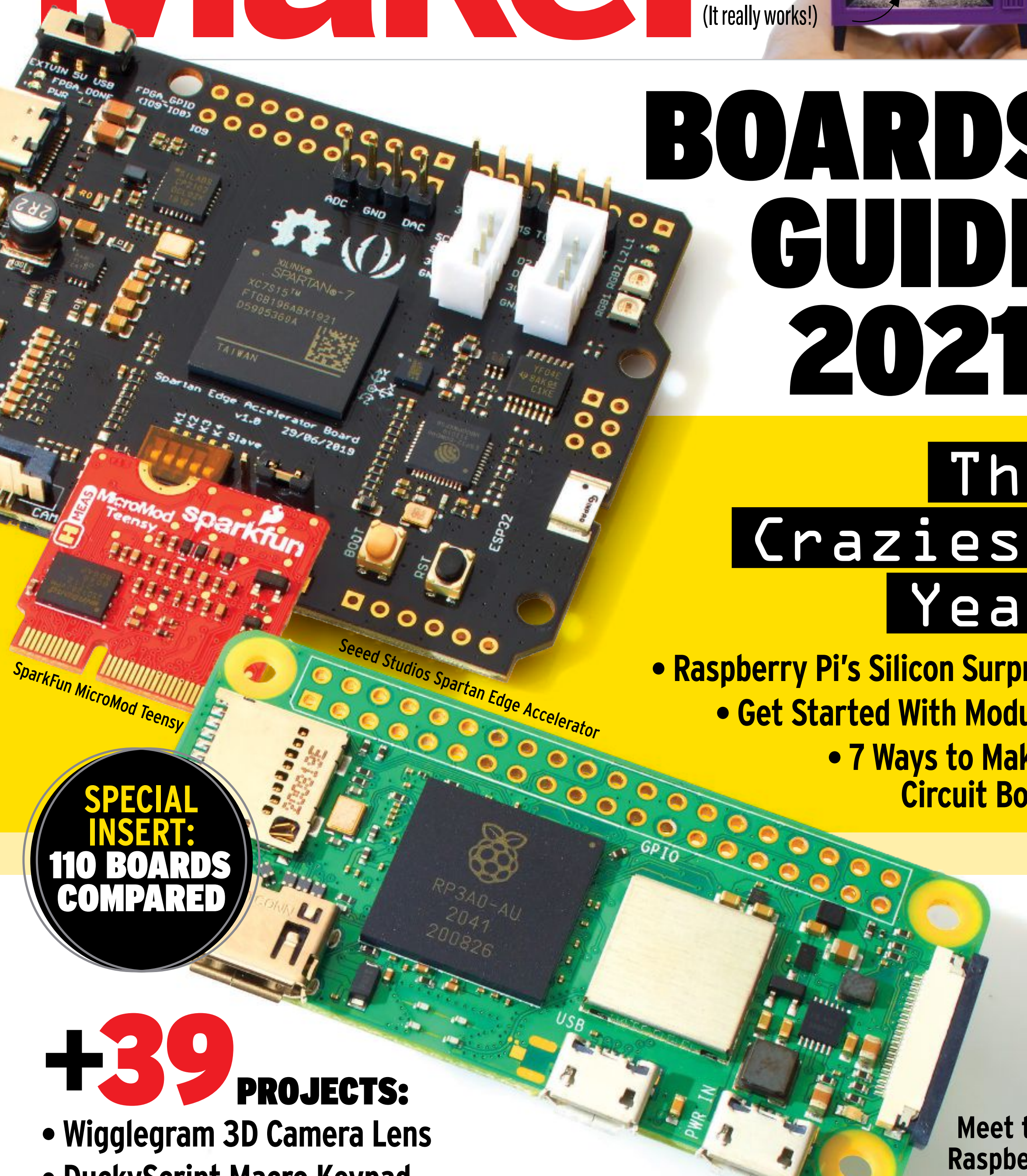
Make:

BUILD A TINY **SIMPSONS TV** (It really works!)

BOARDS GUIDE 2021:

The Craziest Year

- Raspberry Pi's Silicon Surprise
- Get Started With Modules
- 7 Ways to Make a Circuit Board



SparkFun MicroMod Teensy

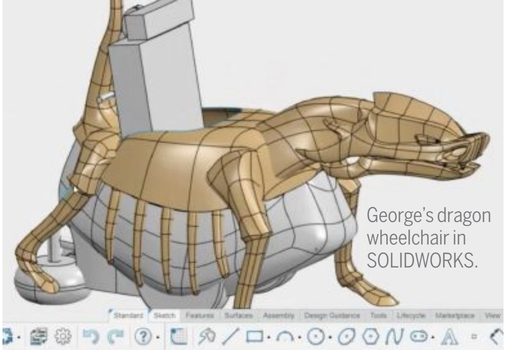
Seeed Studios Spartan Edge Accelerator

Meet the Raspberry Pi Zero 2 W!

SPECIAL INSERT: 110 BOARDS COMPARED

+39 PROJECTS:

- Wigglegram 3D Camera Lens
- DuckyScript Macro Keypad
- Cosmic Muon Telescope
- 3D Printed Wood Lathe ... and More!



George's dragon wheelchair in SOLIDWORKS.

Unveiling the finished ride.

SOLIDWORKS COMES TO MAKERS

The **new, affordable** offering brings the leading **CAD tool** to the masses

Kirby Downey (kirbydowney.com) spends his time designing and 3D printing props and other sci-fi and game-inspired creations. The South Africa-born designer taught himself SOLIDWORKS while studying product design at Pretoria's Tshwane University of Technology, and has since leveraged it as his design software of choice.

After university, Downey headed to London to work for MyMiniFactory, where he used SOLIDWORKS to create 3D-printed props for cosplay-focused YouTubers. His elaborate, oversized pieces from the game *Destiny* caught the attention of the SolidWorks team, who subsequently profiled him and his work — spotlighting a maker using their software.

SolidWorks has now released **3DEXPERIENCE SOLIDWORKS for Makers**, a low-cost suite of SOLIDWORKS cloud-based and cloud-connected design tools that puts the leading professional design tool into the hands of hobbyists and enthusiasts — anyone doing personal, non-commercial projects. In addition to this offer, the companion **Made in 3D** online community lets makers work with each other, and provides access to guidance and support.

Downey and his fellow SolidWorks Champions celebrated the launch of this new offer with a SolidWorks-sponsored collaboration on a Magic Wheelchair build for George, a UK youth with spinal muscular atrophy. Magic Wheelchair is a nonprofit organization that builds epic costumes for kiddos in wheelchairs, at no cost to their families. The group designed a Ninjago-inspired dragon that George can ride in, complete with articulating wings. "When he told us he wanted something Lego-themed, I was like 'Yes!'" says Downey. "George was blown away by it."

3DEXPERIENCE SOLIDWORKS for Makers is available now. For just US $99/year you'll get the same intuitive cloud-connected CAD modeling tools that the professionals use, along with:

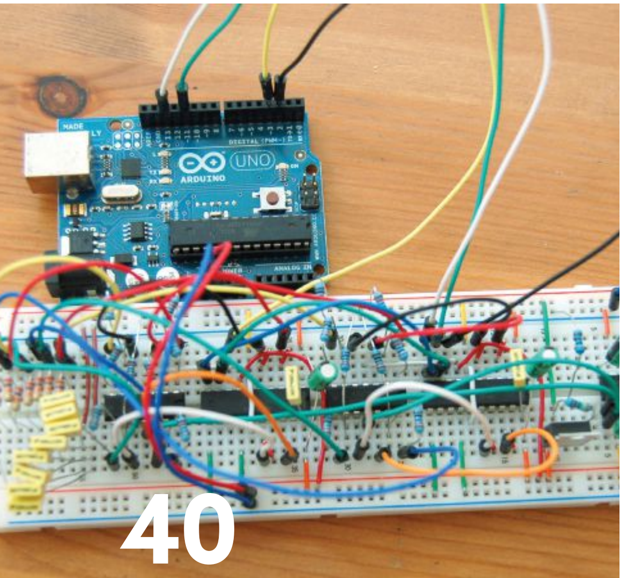
- **Fully online design solutions you can access from any web browser — no download required**
- **An online community that lets you connect with worldwide makers from fab labs, makerspaces, and influencers, all ready to share their designs, ideas, and expertise**
- **Access to an expanded professional ecosystem to rapid prototype your parts, or receive engineering services via the 3DEXPERIENCE Marketplace**
- **Support to help you get the most out of 3DEXPERIENCE SOLIDWORKS for Makers**

SOLIDWORKS

Discover more at:
discover.solidworks.com/makers

Make: 79

CONTENTS



ON THE COVER:
New FPGAs, microcontrollers, and single-board computers give makers a lot to get excited about this year.

Photos: Mark Madeo and Brandon Withrow

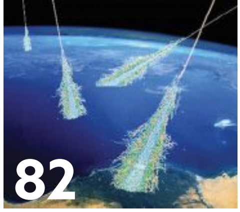
COLUMNS

FEATURES

B.O.A.R.D.S.

SPECIAL INSERT

PROJECTS



Mark Madeo, George_P._Macklin - wikipedia, Brandon Withrow, Allen Wong, Carol Liggett, Bob Knetzger, Katie Marchese

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: 10/1/2021. 4. Issue frequency: Quarterly. 5. Number of issues published annually:4. 6. The annual subscription price is 34.99. 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, Mike Senese, 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 2021. 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:56,563. Actual number of copies of single issue published nearest to filing date: 57,900. B. Paid circulation. 1. Mailed outside-county paid subscriptions. Average number of copies each issue during the preceding 12 months: 41,376. Actual number of copies of single issue published nearest to filing date: 43,244. 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: 4,310. Actual number of copies of single issue published nearest to filing date: 4,250. 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: 45,686. Actual number of copies of single issue published nearest to filing date;47,494. 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:16. Number of copies of single issue published nearest to filing date: 27. 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: 704. Number of copies of single issue published nearest to filing date: 639. E. Total free or nominal rate distribution. Average number of copies each issue during preceding 12 months: 720. Actual number of copies of single issue published nearest to filing date: 666. F. Total free distribution (sum of 15c and 15e). Average number of copies each issue during preceding 12 months: 46,406. Actual number of copies of single issue published nearest to filing date: 48,160. G. Copies not Distributed. Average number of copies each issue during preceding 12 months: 10,157. Actual number of copies of single issue published nearest to filing date: 9,740. H. Total (sum of 15f and 15g). Average number of copies each issue during preceding 12 months: 56,563. Actual number of copies of single issue published nearest to filing: 57,900. I. Percent paid. Average percent of copies paid for the preceding 12 months: 98.45%. Actual percent of copies paid for the preceding 12 months: 98.62% 16. Electronic Copy Circulation: A. Paid Electronic Copies. Average number of copies each issue during preceding 12 months: 3,661. Actual number of copies of single issue published nearest to filing date: 3,406. B. Total Paid Print Copies (Line 15c) + Paid Electronic Copies (Line 16a). Average number of copies each issue during preceding 12 months: 49,347. Actual number of copies of single issue published nearest to filing date: 50,900. C. Total Print Distribution (Line 15f) + Paid Electronic Copies (Line 16a). Average number of copies each issue during preceding 12 months: 50,067. Actual number of copies of single issue published nearest to filing date: 51,566. D. Percent Paid (Both Print & Electronic Copies) (16b divided by 16c x 100). Average percent of copies paid during preceding 12 months: 98.56%. Actual percentage of copies paid for single issue published nearest to filing date: 98.71%. 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 2021 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.

Make:

> "What I cannot create, I do not understand."
> —Richard Feynman

PRESIDENT
Dale Dougherty
dale@make.co

VP, PARTNERSHIPS
Todd Sotkiewicz
todd@make.co

EDITORIAL

EXECUTIVE EDITOR
Mike Senese
mike@make.co

SENIOR EDITORS
Keith Hammond
keith@make.co
Caleb Kraft
caleb@make.co

PRODUCTION MANAGER
Craig Couden

CONTRIBUTING EDITOR
William Gurstelle

CONTRIBUTING WRITERS
Ace Makerspace, Cabe Atwell, Roberto Baldwin, Joshua Bird, Chris Borge, Tim Deagan, Kelly Egan, Gareth Halfacree, Paul J. Henley, Daniel Hienzsch, Mel Ho, Brian Jepson, Whitney Knitter, Bob Knetzger, Julie Legault, Elliott Liggett, Mario Marchese, Forrest M. Mims III, Justin Pahara, Eirik Paye, Marshall Piros, Brandon Withrow, Allen Wong, Chris Yohe

CONTRIBUTING ARTIST
Mark Madeo, Rachel "McCrafty" Saad, Julie Legault, they Fai Kosciolek

MAKE.CO

ENGINEERING MANAGER
Alicia Williams

WEB APPLICATION DEVELOPER
Rio Roth-Barreiro

DESIGN

CREATIVE DIRECTOR
Juliann Brown

GLOBAL MAKER FAIRE

MANAGING DIRECTOR, GLOBAL MAKER FAIRE
Katie D. Kunde

MAKER RELATIONS
Siana Alcorn

GLOBAL LICENSING
Jennifer Blakeslee

MARKETING

DIRECTOR OF MARKETING
Gillian Mutti

COMMUNITY MANAGER
Dan Schneiderman

OPERATIONS

ADMINISTRATIVE MANAGER
Cathy Shanahan

ACCOUNTING MANAGER
Kelly Marshall

OPERATIONS MANAGER & MAKER SHED
Rob Bullington

PUBLISHED BY

MAKE COMMUNITY, LLC
Dale Dougherty

Printed in the USA by Schumann Printers, Inc.

Comments may be sent to:
editor@makezine.com

Visit us online:
make.co

Follow us:
@make @makerfaire @makershed
makemagazine
makemagazine
makemagazine
twitch.tv/make
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's your favorite DIY gift you've given or received for the holidays?

Mel Ho
San Francisco, CA
(Guide to Boards reviews)
A friend made me a VHS bootleg of the movie *Drive*, with faux-80s video-rental sleeve and lo-fi tape scanlines to give it that extra grindhouse feel.

Daniel Hienzsch
Las Vegas, NV
(Circuit Board (Re)Flow Chart)
My favorite DIY gift was an N-Scale model train kit I got at age 6. Electricity! Movement! Still have it 40 years later!

Joshua Bird
Cambridge, UK
(DIY Film Wigglegram Lens)
3D-printed lithophanes always make great presents when you're low on time!

Issue No. 79, Winter 2021. *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 or via phone at (866) 289-8847 (U.S. and Canada); all other countries call (818) 487-2037. 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.

READER INPUT

Can't Quite Put My Nose on It...

Bryan Serinese

WHOEVER SMELT IT...

I've been a long-time subscriber and try to make at least one project per edition. This time I took the dive into machine learning after reading so much about it but never seeing how to apply it to my life. [Benjamin] Cabe's Artificial Nose ["Second Sense," Vol 77, page 32] filled a specific need in my house. Taco Tuesday is always an explosive affair around the dinner table. Plenty of refried beans and spicy salsa leads to unclaimed gas emissions as the tacos are devoured. Since our Golden Retriever JuneBug loves to sit under the table waiting for the inevitable scrap to fall, she is often blamed for the odors wafting around. She can't deny it, so we assume she supplied it. Not any more now that I built the Artificial Nose. I had no problem training the nose to detect the difference between a toot from JuneBug and myself as we both have a steady supply. I just couldn't get the rest of the family to help train my Nose so there will be a built-in bias until I can figure out a way to get more data points. *—Bryan Serinese, via email*

MAKER CAMP 2021

MAKING TOGETHER IS BETTER

As we wrap up our public library's summer reading program, I wanted to reach out to thank you for the phenomenal resources provided by Maker Camp this year. From the project ideas to the supplies, you enabled us to provide high quality, hands-on events for local families.

After a year of being unable to do in-person programming, families were absolutely delighted to do hands-on projects together.

Families worked together on varied projects and activities, and we were reminded of the joy, creativity, and importance of gathering together and making. We heard nothing but glowing reviews of the Maker Camp sessions, from both the public and the staff who assisted.

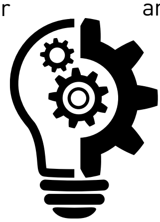
Thank you so, so much for the supplies, the projects, and the delight Maker Camp brought to our corner of the world. It is deeply appreciated! *—Kelly Durkin, Warren County Library, Blairstown, NJ*

The New BMOC

by Dale Dougherty, President of Make: Community

Some stand several stories tall in bright new buildings, with various labs for all the equipment and group workbenches. They are seen as incubators for student startups, design and innovation hubs, and gathering spaces. They have full-time staff supplemented by teams of student workers. Their operating budgets are the envy of a community makerspace.

Welcome to the **Big Makerspace on Campus**. Makerspaces are becoming important college features, designed to attract and encourage students to become innovators and creators. To recognize colleges and universities that offer makerspaces and maker programs, *Make:* in association with *Newsweek* recently released an unranked listing of Best Maker Schools that provide students with access to the tools and materials for designing and making things (see page 14).



The University of Maryland College Park provides a recognizable pattern of growth for makerspaces on campus. In 2012, an assistant professor at the Human-Computer Interaction Lab created the HCIL Hackerspace, which was intended to serve its own researchers. In 2014, the first publicly-accessible makerspace on the UMD campus was started in the basement of the Chemistry building by a student organization, Terrapin Hackers, called Collider Makerspace. Around this time, 3D printers first appeared in the McKeldin Library, which eventually became the John and Stella Graves Makerspace. Then at the Clark School of Engineering, Terrapin Works was founded to give access to advanced additive manufacturing tools and provide a mix of hands-off production and hands-on experiences.

In 2014 Bill Pugh, an emeritus computer science professor, pitched the idea of a large makerspace to Brendan Iribe, a former CS student and one of the co-founders of Oculus. Iribe was donating $30M for a new computer science building, to be called the Iribe Center. "Look at the early days of Oculus," said Pugh of the Oculus offices. "It was like a makerspace." He explained to Iribe that the makerspace was for students to be able to do self-directed projects without being supervised by faculty, to do whatever they want. "That very much appealed to him," Pugh says. Pugh opened a small Sandbox Makerspace in 2015 as a test. In April 2019, the Iribe Center opened, dedicating one floor as the Singh Family Sandbox Makerspace.

Elsewhere on campus, new makerspaces had opened such as Vortex (which took over the former Physics Welding Shop), the Clarice Scene Shop, the FabLab in the Architecture department, and Booklab in the Humanities department. While some expanded, others closed. Collider shut down when student leaders of Terrapin Hackers graduated. The HCIL Hackerspace closed after its founder left. By 2020, there were seven independent makerspaces on campus. The leaders of these spaces formed the University of Maryland Makerspace Initiative (makerspace.umd.edu) to align as part of an entire ecosystem.

"Typically each department has a makerspace or two and then you have a couple stray ones," said Rick Blanton, the Director of Technical Operations for Terrapin Works. "If you have wildly different requirements for gaining entry to a space or different rules or cost structures, it just makes it very confusing." The groups sought to improve access by standardizing safety training. They also wanted to promote the makerspaces to students and faculty, sharing the skills and knowledge that each one offered. They published the 2021 Makerspace Impact Report, which does an excellent job of showcasing the many makerspaces at the University of Maryland and their varied uses by students. Rick added that the group wanted to be "good advocates for the maker movement," which is thriving there and on campuses across the country.

Listen to our "Makerspaces on Campus" group discussion: makezine.com/go/makerspaces-on-campus

MADE ON EARTH

Backyard builds from around the globe

Found a project that would be perfect for Made on Earth? Let us know: *editor@makezine.com*

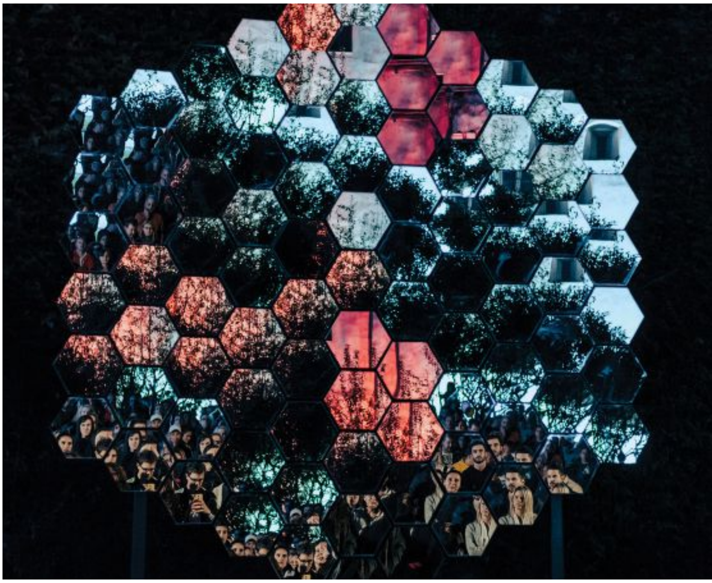
MIRROR, MIRROR...

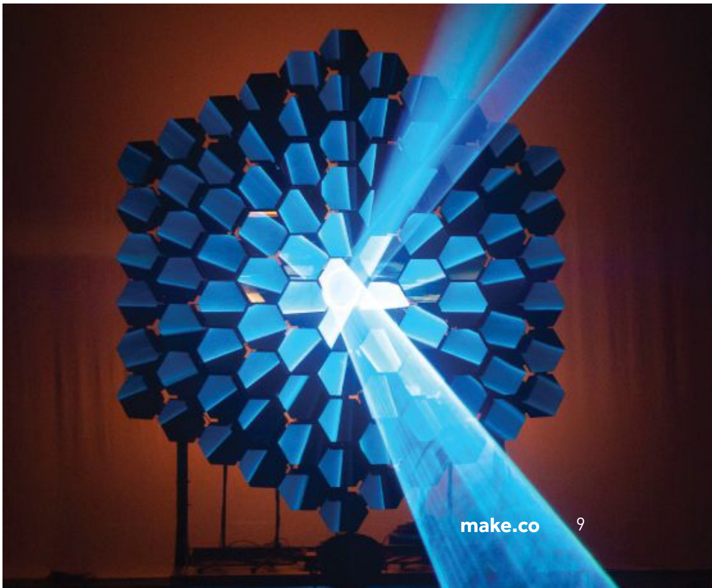
PRUSALAB.CZ/PROJEKTY/REFLEXE

When you're at a music festival, you'll always find many visual elements designed to inspire and dazzle. There are lasers, motorized lights, fog machines, pyrotechnics, and more, with producers unveiling new creations regularly to keep pushing the limit on ways to captivate a crowd.

Reflection is one of those festival installations that grabs your attention by showing you... yourself. The massive, articulated array of mirrors bobs and weaves to mesmerize the onlooking crowds, rippling in patterns and sending random spots of light throughout a space. This undulating light bender has 182 servos controlling 91 mirrors, which in turn are controlled by 18 Raspberry Pi modules. All of this comes together in perfect harmony, in a system that is designed to be deconstructed, transported, and reconstructed at multiple events.

Czech artists **Adam Cigler** and **Petr Vacek** designed and built the piece with support from the Prusa 3D printer company, making heavy use of the makerspace within its factory in Prague. Total timeframe to design and assemble everything was over six months. Their hard work paid off; videos show a constant crowd that couldn't look away when it was displayed at the Signal Festival. *—Caleb Kraft*





Dušan Vondra, Petr Vacek, Alex Dobrovodsky

Daniele Barresi

SCRUMPTIOUS SCULPTURES

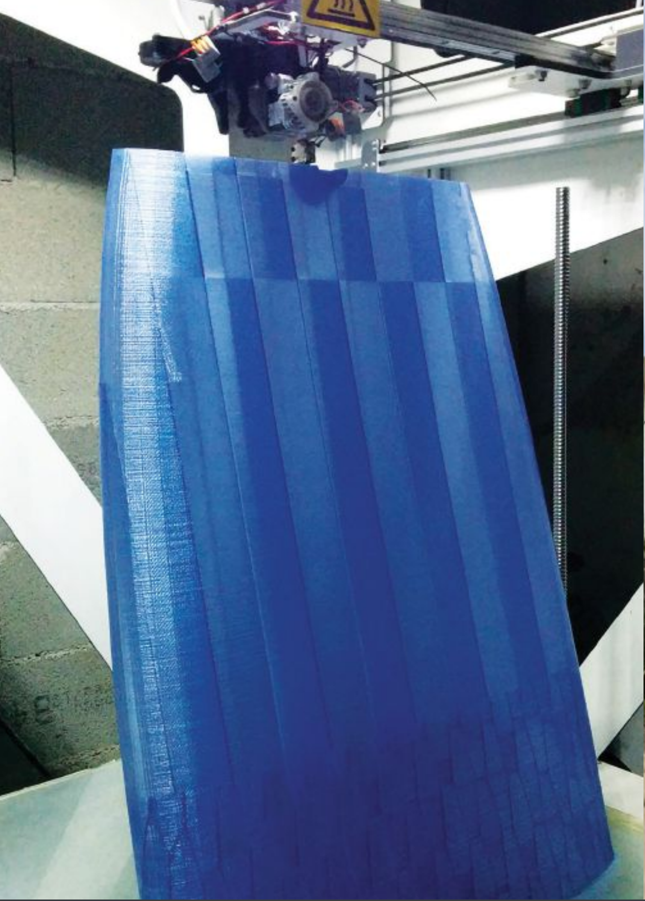
DANIELEBARRESI.NET

When most people get a craving for avocados, they'll likely think of guacamole or toast. But for **Daniele Barresi** (IG: @danielebarresi_artist), an avocado gives him a craving for carving. The Australian-based artist specializes in making impossibly intricate carvings out of an unorthodox set of materials, such as pumpkins, melons, cheese, foam, soap, or avocados.

Barresi has been channeling his inner chef ever since childhood, when he was the designated food preparer for his family's fishing business. "I discovered this love of cooking while working with my family and decided to go to cooking school. But I've always been an artistic person," he says. His creations, which have won him the World Association of Chefs' Societies championship in 2013 and 2014, were started during his culinary studies. "I saw this amazingly carved pumpkin at a wedding, and thought to myself 'I want to know how to do that.' So I started watching videos and tutorials on YouTube, and here we are today."

Although Barresi's materials of choice tend to be on the smaller side, he is no stranger to working on large carvings. He was commissioned to make a series of 18-inch soap sculptures by Hilton for their Global Handwashing Day exposition in 2019, and carved strawberries and watermelons for an International Women's Day event that members of the Australian government attended. Currently, he is managing his new pastry shop "Eden Pasticceria" in New South Wales and is working on some special Halloween-themed carvings.

Barresi tells beginners to be persistent, never give up, and start out with honeydew melons and zucchini. "Those are the easiest foods to carve because they're not too hard and not too soft." Although his works are feasts for the eyes meant for a king, Barresi loves to inspire any and all with his creations. —*Marshall Piros*



REDUCE, REUSE, PADDLE OUT

YUYO.SURF

Surfers are known for their love of the ocean and fierce dedication to keeping it clean. Unfortunately, surfboards are a toxic combination of materials that are anything but ecologically conscious. **Romain Paul**, founder of surfboard shaping company Yuyo, explains the paradox of surfing, "it's the contradiction between the eco-friendly mindset of most of us and the toxicity of the equipment we use."

After years in marketing and advertising, Paul transformed his surfboard-shaping hobby into a full time job in 2018 in order to create boards that keep true to his ecological ideology. The result is custom 3D-printed surfboards constructed from recycled and non-toxic components.

Made to order using a suite of software ("We use BoardCAD for board shaping, then SpaceClaim for 3D illustration and modifications, and finally Simplify3D for slicing and printings parameters"), Yuyo's boards take three weeks to produce. Production happens on a massive 60cm×60cm×120cm Tobeca printer, using filament made from recycled plastic trays used to sterilize surgery tools. Paul is looking to further that aspect: "We are running an R&D program aiming at producing our own filament made from locally collected plastic bottles," he says.

Changing the way surfboards are made isn't without its issues. The first small-scale prototype didn't float and even after they fined-tuned their system, and the bio-composite lamination they use takes far longer to cure than traditional resin.

With about 100 surfboards in use, the feedback has been positive. "Most surfers and people agree with what we're doing," Paul says. "The objective is to be more than the company that has innovated in a new way, greener way to ride waves. The goal now is for people to realize Yuyo is," as Paul says, "a real surfboard manufacturer people can rely on." *—Roberto Baldwin*

Yuyo

Shop Tour

Peek inside my pop-up backyard woodshop

Written by Eirik Paye

Eirik Paye, Jocelyn and Spencer Photography — hello@jocelynandspencer.com; IG: @jocelynandspencer

When we moved into a new home, I needed to create a woodshop with level space, good air flow, room for power tools, and most of all, protection from the elements. I determined it would need to be 200 square feet. Unfortunately, local zoning doesn't allow for a shed that large, or even a covered deck, but a "floating" temporary deck isn't an issue. And that's how I conceived my 10'×20' carport-tent workspace.

The tent has a pitched roof and flaps that go to the ground on all four sides. The deck is built from 2×8 ground-contact lumber with plywood sheeting, braced by 15 pier blocks on gravel. (Aside: Two blocks later shifted in the rain. Fixing the issue was a simple matter of pulling up one sheet of plywood, using a car jack to lift the deck and moving the blocks back into place.) For my router, chop, and table saw, I attached long shop surge strips to the tent's vertical supports, then placed DIY'd 10-gauge multi-outlet extension cords close to the tools.

A family of rabbits has now turned the space under the deck into a summer home, and the babies poke up and around the shop, sometimes under foot. Blue jays fly in and out and pester me for food on occasion. Working outdoors is great.

EIRIK PAYE is an animation industry art director, concept artist, husband, father, designer, maker, and semi-retired Viking. He's based in Santa Barbara. instagram.com/eirik_paye_studio

Best Maker Schools 2021

from *Make:* and *Newsweek*

Written by Jennifer Blakeslee and Keith Hammond, Make: Community

The Maker Movement is all about "learning by doing," supported by online communities and DIY publications like *Make:*, by 2,000 shared makerspaces around the world, and by in-person Maker Faire events held in 72 countries (and counting). But DIY doesn't mean learning alone — the Maker Movement has also been embraced by educators who seek to prepare students for careers in science, technology, engineering, and the arts.

Hands-on "maker education" — training in the tools and techniques of electronics and engineering, 3D design and digital fabrication, and traditional craft skills — has become a star attraction at institutions of higher learning. Maker education is cutting-edge, constantly evolving, and exciting — a pathway to brand-new careers literally creating the future.

This year, for the first time, *Make:* has teamed with *Newsweek* to seek out the Best Maker Schools in Higher Education. Nominated by our

unique global community of makers, educators, and Maker Faire leaders, we've found 200 great maker schools — powerhouse universities, lesser-known schools with amazing programs, and standout community colleges, vocational and trade schools. Large or small, these are the schools that have invested in innovative maker-focused programs and on-campus makerspaces, the schools that makers want to attend to deepen their skills and broaden their scope.

We hope this list will help guide students, parents, educators, and employers to recognize excellence in maker education around the world.

Methodology

This non-ranked list was generated by recommendation through a survey tool in collaboration with an international community of educators, administrators, students, and maker leaders. To be considered for the list, recommended schools had to demonstrate excellence or competency in the following areas:

- ***Integrated learning-through-doing orientation***
- ***Mentoring / guidance / coaching for making***
- ***Physical makerspaces / fab labs / workshops / studios***
- ***Accessible spaces and tools that support independent, collaborative projects***
- ***A diverse, active community of makers.***

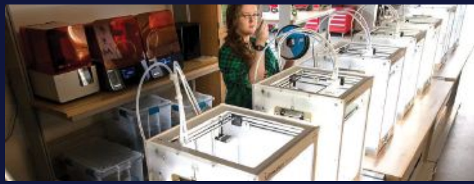
How was *Make:* able to gather the most comprehensive list of maker schools? We're ***non-biased*** — we are not an education institution looking to compete, nor an education-market publisher. We are a community of makers of all types and levels, from self-taught inventors and rank beginners to Ph.D. engineers and entrepreneurs.

We asked our ***unique network*** of makers and educators in the U.S. and worldwide that's been built over 15 years by Maker Faires, *Make:* magazine, and events like our Maker Education Forum. Many of the schools on this list have built a maker community over the years by hosting local and regional Maker Faires.

We strove for a ***balance of programs***: engineering but also arts and design, vocational/technical, mechanical/fabrication, and standout interdisciplinary programs. We also strove for ***geographic balance***, soliciting recommendations for all regions of the U.S. and the world, and considered ***accessibility*** (location and cost).

To learn more about schools on this list, watch the *Make:* blog (makezine.com) and *Make:*cast podcast (makezine.com/makecast) for interviews and profiles of excellent maker schools and programs. To nominate a school for next year's list, visit makezine.com/best-maker-schools-2021-from-make-and-newsweek.

Makings of a Maker School



In the Innovation Wyrkshop at the **University of Wyoming**, students, faculty, and the community can design, tinker, and make using a wide array of creative resources. In addition, this flagship center in Laramie is developing five satellite spaces to expand maker resources across the Front Range. wyrkshop.org

The **Universitat Politècnica de Catalunya** runs diverse, creative future-thinking programs and facilities supported by FabLab Barcelona (based in the Institute for Advanced Architecture, and the global coordinator of Fab Academy) and projects like the self-sufficient Green Fab Lab in the hills outside the city.

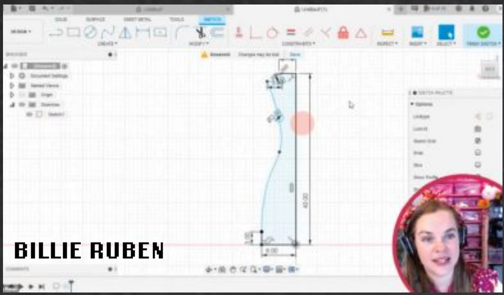
The IDEAstudio at **Houston Community College**'s West Houston Institute provides a community space to explore and create using the latest technologies and traditional tools while encouraging the diverse student body to develop their maker mindset and skills.

Newsweek, Innovation Wyrkshop, IAAC, IDEAstudio

2021 Best Maker Schools in Higher Education

- **University of New South Wales**
  Australia, NSW, Sydney
- **University of Wollongong**
  Australia, NSW, Wollongong
- **University of Queensland**
  Australia, QLD, Brisbane
- **Monash University**
  Australia, VIC, Melbourne
- **TU Graz**
  Austria, Graz
- **Instituto Federal do Rio Grande do Sul - Campus Porto Alegre**
  Brazil, Porto Alegre
- **Universidade de São Paulo (USP)**
  Brazil, São Paulo
- **Emily Carr University of Art+Design**
  Canada, BC, Vancouver
- **Simon Fraser University**
  Canada, BC, Vancouver
- **Wilson School of Design**
  Canada, BC, Richmond
- **University of Toronto**
  Canada, Ontario, Toronto
- **Ryerson University**
  Canada, Ontario, Toronto
- **Concordia University**
  Canada, Quebec, Montreal
- **Universidad Viña del Mar**
  Chile, Viña del Mar
- **The Hong Kong Polytechnic University School of Design (PolyU Design)**
  China, Hong Kong
- **Tongji University**
  China, Shanghai
- **CUHK-Shenzhen**
  China, Shenzhen
- **Tsinghua University**
  China, Beijing
- **University of Zagreb**
  Croatia, Zagreb
- **Czech Technical University (CTU Prague)**
  Czech Republic, Prague
- **UCL Jelling**
  Denmark, Jelling
- **The American University Cairo**
  Egypt, Cairo
- **Talinn University of Technology (TalTech)**
  Estonia, Talinn
- **Aalto University**
  Finland, Espoo
- **ENSCI Les Ateliers**
  France, Paris
- **Paris College of Art**
  France, Paris
- **TU Munich**
  Germany, Munich
- **Hochschule Rhein Waal**
  Germany, Kamp-Lintfort
- **University of Europe for Applied Sciences**
  Germany, Berlin
- **University of Ghana**
  Ghana, Accra
- **Universidad Del Valle de Guatemala**
  Guatemala, Guatemala City
- **CMR University**
  India, Bangalore
- **Technion (Israel Institute of Technology)**
  Israel, Haifa
- **Politécnico di Milano (Polimi)**
  Italy, Milan
- **Università degli Studi di Torino**
  Italy, Torino
- **La Sapienza di Roma**
  Italy, Rome
- **Institute of Advanced Media Arts and Sciences**
  Japan, Ogaki
- **Tokyo Metropolitan University**
  Japan, Tokyo
- **Korea University**
  Korea, Seoul
- **Malawi University of Business and Applied Sciences**
  Malawi, Blantyre
- **Tech de Monterrey Mexico**
  Mexico, Monterrey
- **Universidad Nacional Autónoma de México**
  Mexico, Mexico City (D.F.)
- **La Universidad Politécnicade Baja California**
  Mexico, Tijuana
- **Design Academy Eindhoven**
  Netherlands, Eindhoven
- **TU Delft**
  Netherlands, Delft
- **Amsterdam University of the Arts**
  Netherlands, Amsterdam
- **University of Auckland**
  New Zealand, Auckland
- **Massey University**
  New Zealand, Palmerston North
- **University of Lagos**
  Nigeria, Lagos
- **University of Oslo (UIO)**
  Norway, Oslo
- **ITU Punjab**
  Pakistan, Lahore
- **Universidad de Lima**
  Peru, Lima
- **Politécnico de Lisboa**
  Portugal, Lisbon
- **National University of Science and Technology MISIS**
  Russia, Moscow
- **King Abdullah University of Science and Technology**
  Saudi Arabia, Thuwal
- **University of Ljubljana**
  Slovenia, Ljubljana
- **University of Pretoria**
  South Africa, Hatfield
- **Polytechnic University of Catalonia (UPC)**
  Spain, Barcelona
- **Madrid CEU University**
  Spain, Madrid
- **Umeå University**
  Sweden, Umeå
- **Uppsala University**
  Sweden, Uppsala
- **University of Geneva**
  Switzerland, Geneva
- **National Kaohsiung First University of Science and Technology (NKFUST)**
  Taiwan, Kaohsiung City
- **Lunghwa University of Science and Technology LOHOC**
  Taiwan, Taoyuan City
- **UAE University**
  UAE, Abu Dhabi
- **University of Sheffield**
  UK, Sheffield
- **Cambridge University**
  UK, Cambridge
- **University College London**
  UK, London
- **University of Brighton**
  UK, Brighton
- **University of Manchester**
  UK, Manchester
- **London College of Communication**
  UK, London
- **University of Surrey**
  UK, Guildford
- **Royal College of Art**
  UK, London
- **Glasgow School of Art**
  UK, Scotland, Glasgow
- **University of Alaska Fairbanks**
  USA, AK, Fairbanks
- **Auburn University**
  USA, AL, Auburn
- **University of Arkansas**
  USA, AR, Fayetteville
- **Arizona State University**
  USA, AZ, Tempe
- **University of Arizona**
  USA, AZ, Tucson
- **GateWay Community College**
  USA, AZ, Phoenix
- **Stanford University**
  USA, CA, Palo Alto
- **University of California, Berkeley**
  USA, CA, Berkeley
- **University of Southern California**
  USA, CA, Los Angeles
- **Sierra College**
  USA, CA, Rocklin
- **Cabrillo College**
  USA, CA, Aptos
- **California College of the Arts**
  USA, CA, Oakland
- **University of California, Davis**
  USA, CA, Davis
- **University of California, San Diego (UCSD)**
  USA, CA, San Diego
- **University of California, Los Angeles (UCLA)**
  USA, CA, Los Angeles
- **California Institute of Technology**
  USA, CA, Pasadena
- **CalPoly SLO**
  USA, CA, San Luis Obispo
- **SciArc**
  USA, CA, Los Angeles
- **Harvey Mudd College**
  USA, CA, Claremont
- **Sacramento City College**
  USA, CA, Sacramento
- **University of Colorado, Boulder**
  USA, CO, Boulder
- **Yale University**
  USA, CT, New Haven

- **Wesleyan College**
  USA, CT, Middletown
- **Georgetown University**
  USA, DC, Washington
- **Florida Polytechnic University**
  USA, FL, Lakeland
- **University of West Florida**
  USA, FL, Pensacola
- **University of Florida**
  USA, FL, Gainesville
- **Georgia Institute of Technology**
  USA, GA, Atlanta
- **Morehouse College**
  USA, GA, Atlanta
- **Spelman College**
  USA, GA, Atlanta
- **Savannah College of Art and Design (SCAD)**
  USA, GA, Savannah
- **University of Georgia**
  USA, GA, Athens
- **Emory University**
  USA, GA, Atlanta
- **University of Hawaii at Manoa**
  USA, HI, Honolulu
- **Iowa State University**
  USA, IA, Ames
- **Boise State University**
  USA, ID, Boise
- **University of Illinois, Urbana-Champaign**
  USA, IL, Urbana
- **DePaul University**
  USA, IL, Chicago
- **Northwestern University**
  USA, IL, Evanston
- **University of Chicago**
  USA, IL, Chicago
- **School of the Art Institute of Chicago**
  USA, IL, Chicago
- **Purdue University**
  USA, IN, Lafayette
- **Fort Hays State University**
  USA, KS, Hays
- **Hazard Community and Technical College**
  USA, KY, Hazard
- **Tulane University**
  USA, LA, New Orleans
- **Wheaton College**
  USA, MA, Norton
- **Massachusetts Institute of Technology**
  USA, MA, Boston
- **University of Massachusetts Amherst**
  USA, MA, Amherst
- **Olin College**
  USA, MA, Needham
- **Boston University**
  USA, MA, Boston
- **Harvard University**
  USA, MA, Boston
- **Hampshire College**
  USA, MA, Amherst
- **Mount Holyoke College**
  USA, MA, South Hadley
- **Smith College**
  USA, MA, Northampton
- **University of Maryland**
  USA, MD, College Park
- **Johns Hopkins University**
  USA, MD, Baltimore
- **University of New England**
  USA, ME, Biddeford
- **University of Michigan**
  USA, MI, Ann Arbor
- **University of Minnesota**
  USA, MN, Minneapolis - St. Paul
- **Mississippi State University**
  USA, MS, Mississippi State
- **University of Montana**
  USA, MT, Missoula
- **Montana State University**
  USA, MT, Bozeman
- **University of Nebraska-Lincoln**
  USA, NB, Lincoln
- **Duke University**
  USA, NC, Durham
- **Penland School of Craft**
  USA, NC, Penland
- **University of North Carolina at Chapel Hill**
  USA, NC, Chapel Hill
- **North Dakota State University**
  USA, ND, Fargo
- **New Jersey Institute of Technology**
  USA, NJ, Newark
- **Princeton University**
  USA, NJ, Princeton
- **Stevens Institute of Technology**
  USA, NJ, Hoboken
- **University of New Mexico**
  USA, NM, Albuquerque
- **Central New Mexico Community College**
  USA, NM, multiple locations
- **Navajo Technical University**
  USA, NM, Crownpoint
- **University of Nevada, Reno**
  USA, NV, Reno
- **Columbia University**
  USA, NY, New York
- **New York University**
  USA, NY, New York
- **Cooper Union**
  USA, NY, New York
- **Alfred University**
  USA, NY, Alfred
- **Cornell University**
  USA, NY, Ithaca
- **SUNY Polytechnic Institute**
  USA, NY, Utica
- **Pratt Institute**
  USA, NY, New York
- **Union College**
  USA, NY, Schenectady
- **Rochester Institute of Technology**
  USA, NY, Rochester
- **Parsons School of Design**
  USA, NY, New York
- **Case Western Reserve University**
  USA, OH Cleveland
- **Kent State University**
  USA, OH, Kent
- **University of Cincinnati**
  USA, OH, Cincinnati
- **University of Akron, Wayne College**
  USA, OH, Orville
- **Ohio University**
  USA, OH, Athens
- **Otterbein University**
  USA, OH, Westerville
- **University of Oklahoma**
  USA, OK, Norman
- **University of Oregon**
  USA, OR, Eugene
- **Oregon State University**
  USA, OR, Corvallis
- **Carnegie-Mellon University**
  USA, PA, Pittsburgh
- **Bucknell University**
  USA, PA, Lewisburg
- **University of Pennsylvania**
  USA, PA, Philadelphia
- **Pennsylvania State University (Penn State)**
  USA, PA, University Park
- **Lafayette College**
  USA, PA, Easton
- **Lehigh University**
  USA, PA, Bethlehem
- **Drexel University**
  USA, PA, Philadelphia
- **University of Puerto Rico**
  USA, PR, San Juan
- **Rhode Island School of Design**
  USA, RI, Providence
- **IYRS School of Technology and Trades**
  USA, RI, Newport
- **Brown University**
  USA, RI, Providence
- **South Dakota School of Mines & Technology**
  USA, SD, Rapid City
- **Vanderbilt University**
  USA, TN, Nashville
- **Middle Tennessee State University**
  USA, TN, Murfreesboro
- **Tennessee Tech**
  USA, TN, Cookeville
- **Rice University**
  USA, TX, Houston
- **University of Texas at Austin**
  USA, TX, Austin
- **Houston Community College**
  USA, TX, Houston
- **University of Texas at Arlington (UTA)**
  USA, TX, Arlington
- **Texas A&M**
  USA, TX, College Station
- **Abilene Christian University**
  USA, TX, Abilene
- **University of Utah**
  USA, UT, Salt Lake City
- **Virginia Commonwealth University**
  USA, VA, Richmond
- **George Mason University**
  USA, VA, Fairfax
- **James Madison University**
  USA, VA, Harrisonburg
- **University of Vermont**
  USA, VT, Burlington
- **University of Washington**
  USA, WA, Seattle
- **Washington State University**
  USA, WA, Pullman
- **Milwaukee College of Science and Engineering**
  USA, WI, Milwaukee
- **University of Wisconsin, Madison**
  USA, WI, Madison
- **West Virginia University**
  USA, WV, Morgantown
- **University of Wyoming**
  USA, WY, Laramie



COOL  SCHOOL

Meet the expert instructors that will help you level up your maker skills through our **Maker Campus** courses

Written by Dan Schneiderman

DAN SCHNEIDERMAN is a maker, space enthusiast, co-chair of Maker Faire Rochester, and Community Manager at Make:.

Maker Campus is officially one year old! Implemented by the Make: staff to encourage learning and to support our Maker Faire community while in-person events were not possible, Maker Campus offers folks an opportunity to learn directly from creators from all over the world, live and with immediate guidance. Topics have included puppetry, ham radio, creating chess pieces in Fusion 360, video production for makers, and more! Just about any skill you can learn at a Maker Faire or in *Make:* magazine, you can learn live on Maker Campus.

Here are some of the makers that have taught on Maker Campus:

BILLIE RUBEN

make.co/people/billie-ruben

Billie Ruben has been creating stuff all her life, initially costumes and fashion as a couturier, alongside other hand crafts. But in the last few years Ruben has fallen head-over-heels in love with 3D printing, and since then has helped moderate the largest 3D printing communities on Reddit and Discord (among many others).

In 2020, Ruben started her own YouTube channel where she not only covered her own projects, but also a weekly podcast with Geeky Faye called "Meet a Maker." In their podcast, they take deep dives with makers from all over the world, including Sophy Wong, Jorvon Moss, Hannah Makes, and more.

Ruben was an early Maker Campus instructor and helped find the initial footing for the initiative. Her first workshop covered how to design 3D-printed Christmas ornaments in Fusion 360 as a method to learning the basics in the program. Ruben gave an encore workshop that replaced the ornament project with chess pieces.

GEEKY FAYE (ALLIE KATZ)

make.co/people/Geeky-Faye

Allie Katz, known as Geeky Faye, is a powerhouse maker exploring the intersections of art, technology, and creative experimentation. Whether it's building cosplay costumes from scratch, designing and 3D printing brand new framing methods, or trying electronics live on stream for the first time, Katz makes, documents, and shares the experience on YouTube and across social media to show that anyone can make and be creative.

No matter if you're an aspiring YouTuber, or a maker trying to share your project with the world, being able to create a strong, compelling leadoff image is the first step to giving that project the attention it deserves. This summer, Katz's "Tips & Tricks for Better YouTube Thumbnails" workshop taught folks that good presentation skills go hand in hand with making (and so much more). Katz walked us through the design process, from how to come up with a strong idea for your thumbnail to properly communicating your message. Katz even threw in a few of their own photography and Photoshop tips and tricks.

JEN FOX

make.co/people/jen-fox

After dabbling in dark matter (she has a Bachelor's in physics from Occidental College), Jen Fox settled into engineering and inventing (getting a Master's in mechanical engineering from UCLA) to tackle climate change and social justice. Combining her varied interests and passion for learning, Fox founded FoxBot Industries in mid-2015 to provide an arts-based approach to STEM education. She's also a project manager at Microsoft for maker-related efforts.

This past July, Fox taught three workshops covering whimsical wearables, including creating custom controllers and using an Adafruit Circuit Playground Express with MakeCode to make gesture-controlled lights.

MITCHELL MALPARTIDA

make.co/people/mitchell-malpartida

In 2018, Mitchell Malpartida founded Masterful Creations STEAM Academy to help promote education and the maker mindset. When developing projects, Malpartida is always looking for ways to create experiences that inspire kids (especially his own) as well as adults.

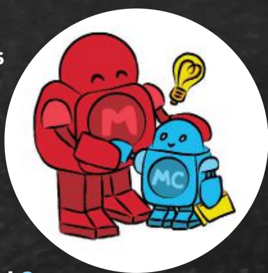
Malpartida originally caught our attention when he submitted his *Moana*-based Chicken Robot Voice Assistant (it only responds in chicken-speak; see page 105) to our Raspberry Pi contest. He has also created interactive gingerbread houses, *Star Wars* cosplays for his kids, and *Minecraft* candy dispensers.

This past September, Malpartida taught a weekend workshop where attendees learned to build their own "talking" animatronic sidekicks similar to his Chicken Robot. All attendees needed for the workshop was a Raspberry Pi, a bit of cardboard, a handful of servo motors, and piano wire.

Find all the upcoming listings at makercampus.com

Want to teach a course? Send a note to makercampus@make.co

Sign up and have fun learning!

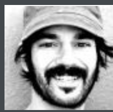


they Fai Kosciolek, Adobe Stock-khwanchai

FROM NO CHIPS TO NEW CHIPS

Written by Mike Senese

2021 HAS BEEN THE CRAZIEST YEAR

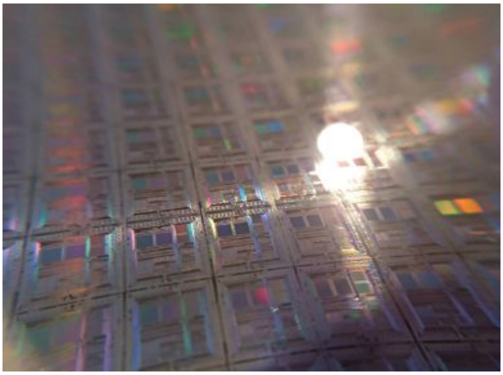


MIKE SENESE is the executive editor of *Make:*. He lives in the San Francisco Bay Area. @msenese

When we look back at the start of the 2020s, there's no question that it will represent an unprecedented period in our lives. The terrifying global pandemic, with its ensuing (and necessary) shutdowns of nations and economies worldwide to try to combat a deadly virus, has resulted in repercussions that no analyst could have forecast, and has exposed the precarious balance of our manufacturing logistics and commercial supply chains.

One highly visible ripple effect can be seen in the chip and component shortages that reared up in a big way this year. Anyone who's been shopping for cars lately is aware of the elevated prices of new and used vehicles alike. A big part of the cause? Manufacturers can't get microcontroller chips needed to complete auto assembly, so unfinished new cars have been stockpiling at factories, their completion stalled until the manufacturers can get those outstanding parts. Early estimates say this will cost the industry $110 billion in 2021.

This same situation has, to differing degrees, affected gaming consoles, televisions, and other consumer electronics that use silicon chips — like the dev boards that we makers love to program and use in our projects. Many popular boards have been impossible to buy for much of the year. Good luck finding a micro:bit or various flavors of Raspberry Pi from most resellers right now. Certain Nvidia Jetson dev kits are still unavailable as well. The scarcity continues through much of the boards lineup; when you can




JAY CARLSON
@jaydcarlson

In the distant future, alien archeologists are going to stumble upon troves of STM32 dev boards with the MCUs removed and wonder what the hell happened on Earth in 2021

1:49 PM · Jul 5, 2021 · Twitter Web App

22 Retweets 1 Quote Tweet 170 Likes

find something specific, you may also find that the selling price has been boosted, sometimes by four times above what it's listed for. (We even made the decision to list current market prices in our included "Guide to Boards" insert this year, as the MSRP is so far off in many cases.)

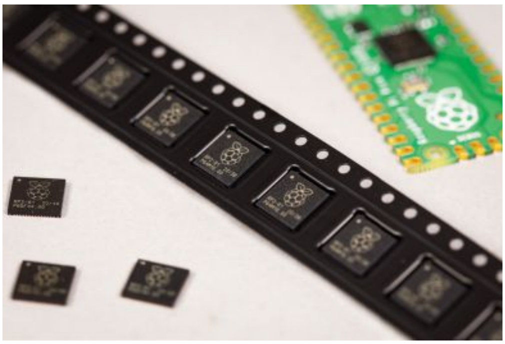
Some chip makers are struggling with this more than others, and some of their supply chain troubles have been compounded by unrelated and unexpected factory shutdowns caused by Covid,

Mark Madeo, timo vn - flickr CC BY 2.0, Adobe Stock-ahmadwahyu27

STMicro and NXP chips, two of the many you might not be able to find this year.

fires, or adverse weather. NXP's manufacturing plant in Austin closed down during the unusual Texas freeze early this year; one maker-focused electronics company says that they currently can't manufacture any NXP chip-based boards. ST Micro's STM32 is another popular chip that's just not available anywhere right now.

The root of these shortages is multi-faceted. Experts say that much of it started with the brief pause in consumer demand caused by the 2020 lockdown, followed by demand suddenly ballooning as a home-bound global community sought entertainment and activity while keeping indoors. Manufacturers, initially trying to avoid an expensive buildup of components, closed their factories that spring; once the demand came back, many struggled to reopen due to Covid. Along with that, the pandemic also ruptured most parts of the global shipping pipeline. Containers are in short supply, often unable to make return trips to active ports. Container ships themselves have backed up at busy ports at record levels. Labor shortages have made it hard to find truck drivers. Not only is it slow and hard for consumers to get their finished goods, it's just as hard for the manufacturers to get the raw materials needed to make those goods.



Raspberry Pi's surprise new RP2040 chip is one component that you *can* get in 2021.

By this point, much has been written about the chip situation (we recommend Nilay Patel's interview with Harvard professor Willy Shih for insights into not just the shortages, but a fascinating explanation of how silicon chips themselves are manufactured: makezine.com/go/chip-shortage). But what are makers themselves doing about it? Well, as the community is known for, they're being resourceful. They're redesigning their projects to use chips they can access. They're digging into storage closets to find their unused and forgotten chips from past projects. They're joking (or are they?) on Twitter about yanking the harder-to find chips from PCBs to use in new endeavors. One company, OKdo, has even built a recycling program for Raspberry Pi computers (okdo.com/raspberry-pi-renew), renewing previously used boards and putting them back into circulation.

Analysts say the chip shortages may last until 2023, so buckle up for the ride.

OTHER NEWS

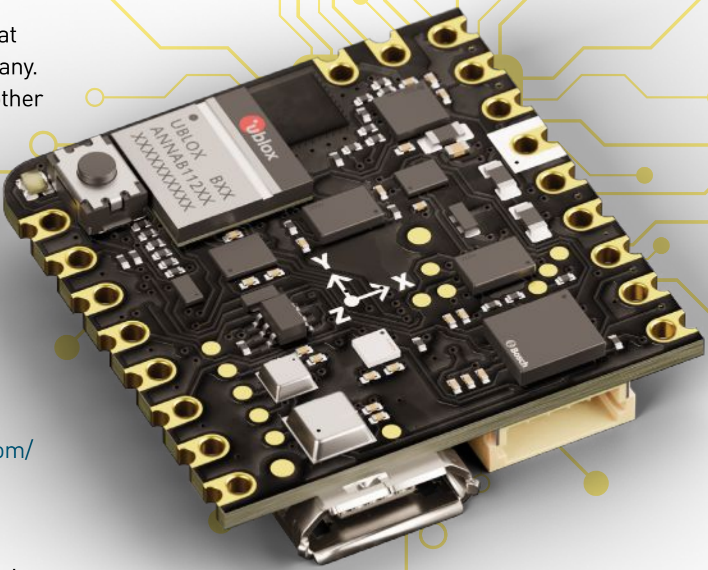
In the face of the chip shortages, the biggest board news in 2021 has been the surprise release of a new microcontroller chip designed by Raspberry Pi, the RP2040. This turned heads everywhere; be sure to read our interview with Eben Upton (page 24) about how it came to be, along with our primer on using its PIO pins (page 30) and a look at the sudden and expansive number of boards that are powered by the RP2040 (Guide to Boards insert, page 3).

We're also seeing the continuation of board makers moving into professional and industrial-grade solutions, covered in part in our "Mighty Modules" article (page 34). This fall, Arduino continued their own foray into this space, adding a few new products to its Arduino Pro line, including the high-power Portenta H7 Lite and the diminutive Nicla Sense ME boards, along with machine-control and edge-control carrier boards. And the power of FPGA boards is getting cheaper and easier to use, with boards like Seeed Studios' Spartan Edge Accelerator.

Adafruit continues to produce a staggering variety of new boards, both as standalones and as purpose-built devices, all with the

Mike Senese, Raspberry Pi

creative, surprising, and useful touches that have brought so many to admire the company. Their Feather form factor has become another industry standard, used by numerous manufacturers like SparkFun, Particle, and Wilderness Labs. But it's the software side of the company that might be the most impressive this year, with active updates to CircuitPython (up to version 7.0.0; now supported by over 230 boards, and even TI calculators) and new releases like their simple-to-use IoT platform WipperSnapper (learn.adafruit.com/quickstart-adafruit-io-wippersnapper).



Arduino's new Nicla Sense ME board is packed with Bosch environmental sensors.

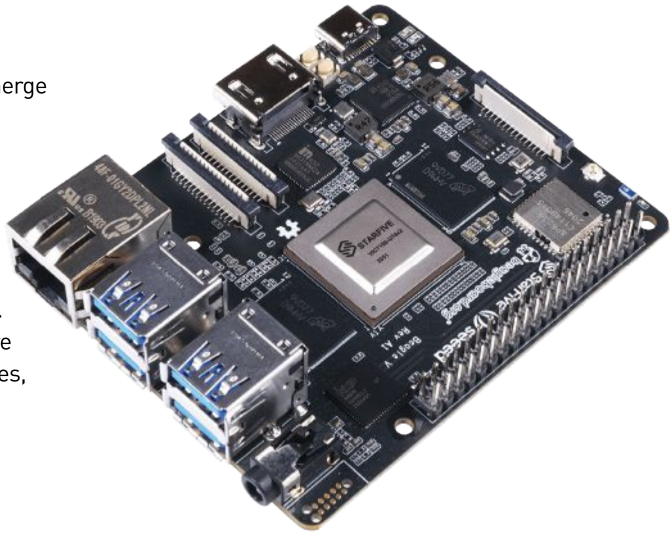
RADIOS

On the board-connectivity side, this year we've finally seen LoRa start to push into more mainstream uses in the U.S. It's getting easier to access LoRaWan networks like Helium, or even to roll your own access points with products from boardmakers like TTGo. One thing that may pose a challenge to deep LoRa adoption is the arrival of free IoT cellular data options. The biggest news in this space came this year from Particle, who announced a new free-for-100-devices tier to their cellular platform. Depending on the scale (and location) of your project, you'll want to weigh which type of connectivity works best for you.

CHIP SHORTAGES MAY LAST UNTIL 2023. BUCKLE UP.

WHAT'S NEXT?

Moving into 2022, we expect for RISC-V to emerge further; we're eager to see more processors and boards using this open source architecture (including the still-underway BeagleV from BeagleBoard). Edge AI is taking center stage and we should be seeing new devices emerge with increased power for accelerated onboard AI processing. And we'll likely see increased prices and more creative solutions to these dang chip shortages, until things eventually settle back to normal. Hopefully soon.

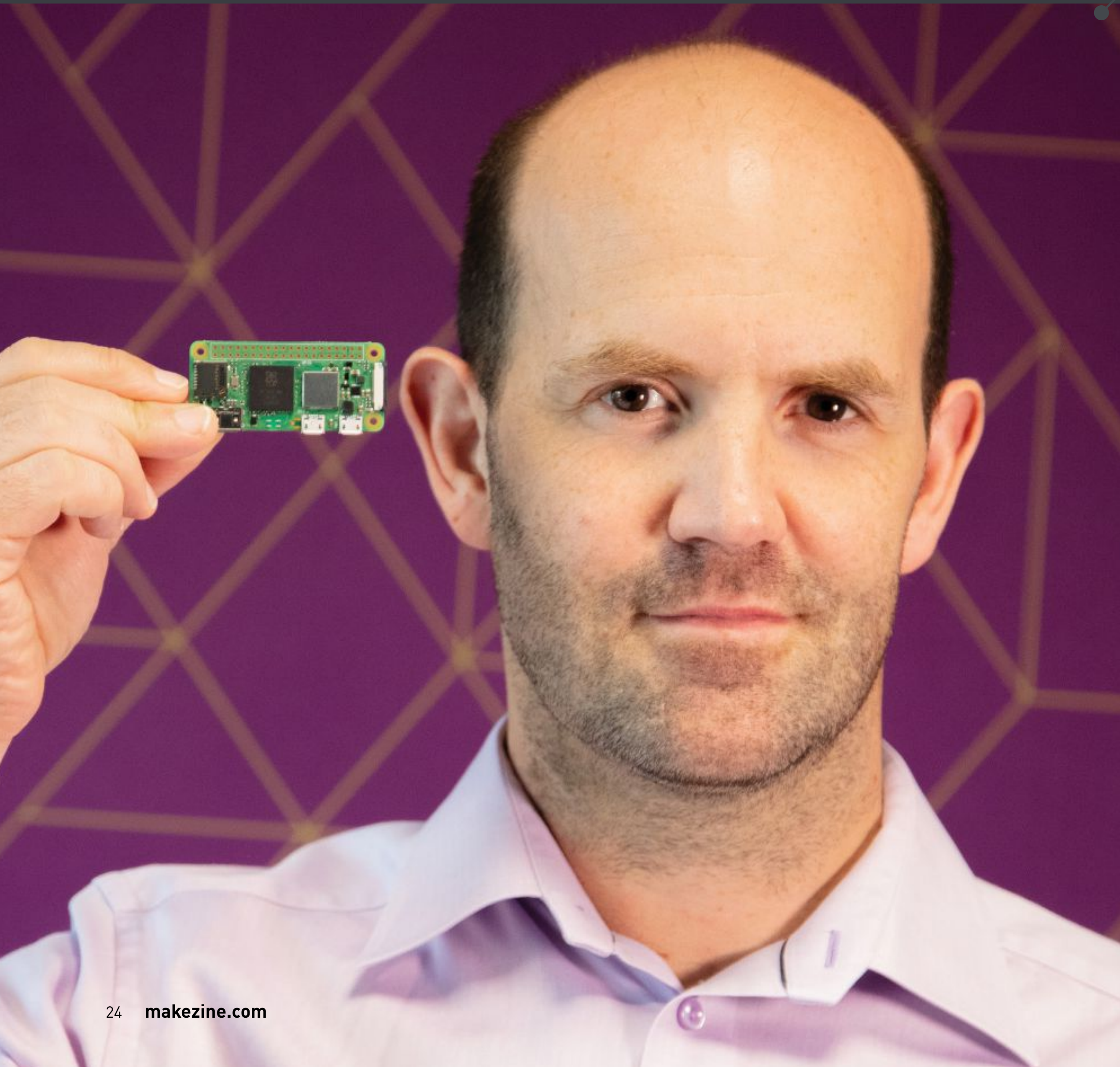


The BeagleV StarLight board went out to beta testers this year; a new variation will be coming in 2022.

PI'S THE LIMIT

Eben Upton talks about Raspberry Pi's most exciting release yet — and reveals the new Pi Zero 2 W

Written by Mike Senese



Just as this year started, Raspberry Pi flipped 2021 on its head with two surprising announcements: they had begun producing their own chip, called the **RP2040**, and they were using it to power their first entry into the world of microcontroller dev boards, the **Raspberry Pi Pico**. To coincide with the launch, four other major board manufacturers released their own RP2040-powered products, followed soon after by a slew of RP2040-based products from the maker community (see "A Chip is Born" on page 3 of the Guide to Boards insert, included with this issue).

We wanted to hear how this new endeavor developed for the organization, so we connected with Raspberry Pi CEO Eben Upton to learn about making modules, silicon, and one last surprise product announcement to close out the year.

Make: Raspberry Pi is now nine years old. Give us a quick overview of where things are at.

Eben Upton: So in nine years we've sold about 42 million Raspberry Pis. We did about 7 million in the last year, so quite high sales right now.

We're about two years into Raspberry Pi 4, which is the biggest step change. Or, I guess technically Raspberry Pi 1 to Raspberry Pi 2 was, because it went from single core to quad core. But in terms of user experience, really 3+ to 4 is the biggest jump that I've seen. That's been really well received by people.

We have the Pi and the Compute Module 4. Then we've added a third form factor, which is the Pi 400, a kind of consumer PC version of Raspberry Pi 4. And those are all going pretty well.

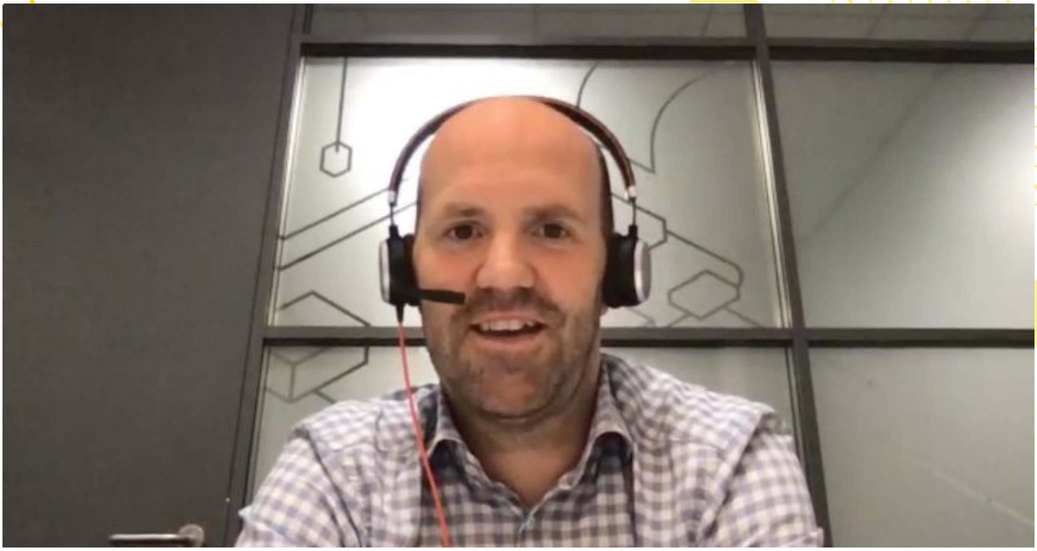
Make: The CM4 has caught our attention more than any other module offering that we've previously seen.

Upton: Compute Module 4 has taken off a lot faster than previous Compute Modules. One, it has wireless, you have an option to have wireless on board so you can just leave wireless to us and we take care of it. And then the other one's the power design. With previous Compute Modules, you had to provide them with several power rails yourself, all of which sequenced properly coming up and coming down. With this one, you just put 5 volts in and it makes its own rails. I think those two things together have made it much easier to work with.

And so we're selling tens of thousands of CM4s a month at a point where, with previous Compute Modules, we probably would have been doing hundreds to thousands. We're seeing volume orders. So yeah, it's busy.

Plus silicon of course. That's the other thing that's going on. Pico is continuing to do really well. We're seeing lots and lots of RP2040 boards, third-party boards built on our silicon. Before we launched, we spoke to SparkFun, Adafruit, Pimoroni, and Arduino. Then after launch, even though we're still in a backlog on Pico, we've been diverting about 15% of chip production to third parties. That's been working out well.

Raspberry Pi



Make: The silicon conversation itself is huge. It's a whole new thing.

Upton: We call it Franchise Two, because it is a new thing. Everything we've done, we do lots of things, but everything we've done up until then is Franchise One — the single board computer and all of the stuff around it, the accessories. And this is the first thing we've done that isn't Franchise One.

Make: It seems like this could end up surpassing Franchise One.

Upton: I think it could. The microcontroller market is a 28 billion-unit-per-year business, and it's an 18 billion-dollar-per-year business. Those two numbers are interesting, because they imply that your average microcontroller only costs 60 to 70 cents.

So you can really see that a microcontroller business is dominated by the ultra low end. The interesting thing about the RP2040 is it is a sub-dollar product, but it has quite a lot of processing power, a lot of memory, and a flexible approach to IO. It's never going to be a "dust" microcontroller. You can buy a 10 cent microcontroller if you only need a few tens of bytes of RAM and a few hundred bytes of program storage. And it's never going to be one of these things at the other end of the bell curve, which has got Cortex M7s, these things they call crossover microcontrollers. Things that are almost application processors. It's never going to be those ends of the bell curve. And of course it's not a connected product as it doesn't have a radio on it. But there's an awful lot of stuff in the middle of that bell curve. I think it's a really competitive product in that space. And they exist. That's the other thing of course for this year. By the middle of next year we will have produced 10, 20 million of these. And so people who want microcontrollers before 2023 may end up looking at it independently of whether it's a good product, [just] because it will exist.

> MAKING CUSTOM CHIPS? **WE CALL IT FRANCHISE TWO, BECAUSE IT IS A NEW THING.**

Make: How have you managed the chip shortages?

Upton: We were lucky that we came into the year in a reasonable stock position, in finished goods and in chips. And we've spent this year depleting those reserves. We've been doing anywhere between 600,000 and 800,000 Raspberry Pis a month this year. In terms of difficulty. I think this between 600,000 and 800,000 Raspberry Pis a month this year. In terms of difficulty. I think this year is probably more of a story about having more demand than we were expecting, rather than having less supply. We have about as much supply as we were expecting, but we have to eke that supply out to service probably 20 to 30% more demand. Last year we did 7 million; this year if things go well in the second half, maybe seven and a half million units. But the really galling thing is there was a 9-million-unit year there for us. And that's what hurts, that we won't get there. Next year — we spent a lot of time early this year making sure that we had our ducks in a row, so I think 2022 shouldn't be too bad for us.

Make: What led you to even start thinking about doing a chip?

Upton: I think that was driven by just a dissatisfaction with what was out there already. I mean there's lots of cool chips, but maybe nothing so exciting as to motivate you to go out and make a whole new sort of product. It was great that the team here were able to think of something a little bit different. I was super pleased with what they've done. This is amazing. It's just an amazing team of engineers here, and they like a challenge.

Make: You must have been working on this for a long time!

Upton: All of these things take a long time, right? That's the big lesson of Raspberry Pi, everything takes a long time. A lot of it's driven by the desire to make things that can be made at scale. There's a huge difference between making things that could be made at prototype scale, or even at small manufacturer scale, 10,000-unit scale. There's a huge difference between 10,000-unit scale and million-unit scale in terms of how much attention to detail you need to put up front into the design, because otherwise you just drown in manufacturing issues, support issues, product quality issues. If you cut corners, you spend more time in the end.

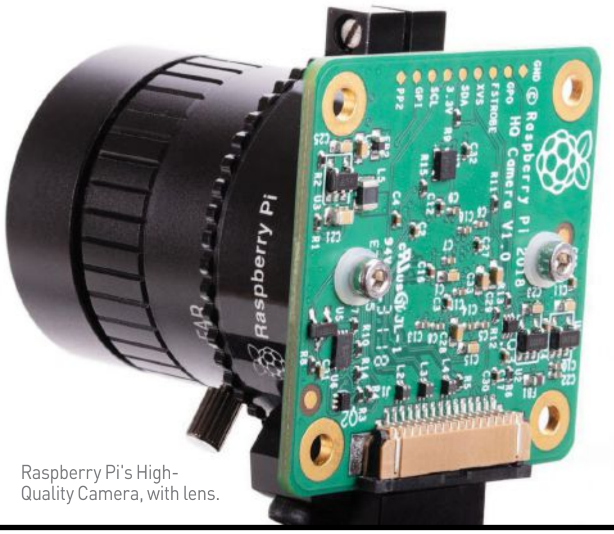
So all these things take a long time. But it's good. The results are worthwhile. And it's always the case at Raspberry Pi that the things people

IT'S JUST AN AMAZING TEAM OF ENGINEERS HERE, AND THEY LIKE A CHALLENGE.

can see that launched were always done by a subset of the team that exists. Very many of the people here are working on things that will appear in the future. Something like Raspberry Pi 4, in terms of the hardware engineering, the board-level hardware engineering, was only a very small handful of people, it was only two or three people. Quite a much larger software team, obviously. But yeah, some of the biggest runners have the smallest teams. Compute Module 4 was one person.

Make: Wow. We wouldn't have guessed that.

Upton: That's a guy named Dominic Plunkett, a fairly recent joiner, and it's his first product for us. The other one is Simon Martin, who joined us quite a long time ago and worked on a number of very long duration projects that all of a sudden all came to fruition at once. The two big ones being the high-quality camera last year and then Pi 400; they were both Simon Martin products. And because of the way that the timescales work, they all end up bunched up together and suddenly



Raspberry Pi's High-Quality Camera, with lens.

Mike Senese

Raspberry Pi Pico can be bought by the reel, similar to individual components used for manufacturing.

Make: (Holds up board) We've got a Pico with the reel packaging that you guys set up. It's brilliant.

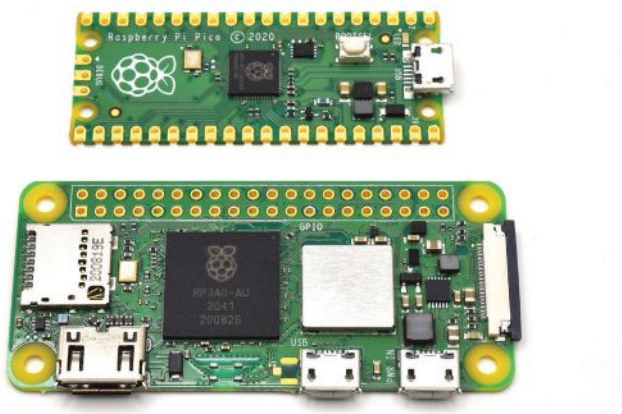
Upton: I love computers on a reel. We have them in the shop. You come in, "I want two yards of computers, please."

Make: (Holds up a different board) Now here's a new thing! Nobody's really seen this before. Tell us about it.

Upton: So that is a Raspberry Pi Zero 2 W. It's a successor to the Zero W, which was itself the successor to the Zero. Zero was our $5 Raspberry Pi, and Zero is a fun product. I still love the Zero, more I think than anything else we've ever done, because it is sort of us challenging ourselves to keep being aggressive after we had some success. What Raspberry Pi has done is it's used Moore's law in a different way from the way people normally use Moore's law. People normally pick a price point and they use the declining, specific cost of computing power to fill that price point up with progressively more computing every year. What Raspberry Pi wanted was to say can we take a PC from 10 years ago and then use Moore's law to squash that PC down to a much lower price point. And we did that. And then we launched Raspberry Pi 2 in 2015, three years later. Then lo and behold, what do we do? We picked our price, $35, and we filled it up with a bunch of computing. Because that's what everyone always tells you to do. That's what your customers always say they want.

Raspberry Pi 2 was a great product, but there was a little bit of a feeling after we launched it that we'd fallen into this trap of being very conventional again. And then Zero is the output of then saying can we challenge ourselves again to squeeze the same performance into a low cost structure. So your $35 Raspberry Pi 1 gets stripped down in a variety of ways and repackaged as a $5 computer. Then Zero W comes along about 18 months later. And that was very popular. And of course now it is the most popular Zero product, at $10. A lot of fun for five extra bucks. So 90% of Zeros that now sell are Zero Ws. But still people ask for more performance. And Zero is a Raspberry Pi 1, so compared to Raspberry Pi 4 it's got about a 40th of the performance of a Raspberry Pi 4 and there are things you can't do with it.

WHAT RASPBERRY PI HAS DONE IS IT'S USED MOORE'S LAW IN A DIFFERENT WAY.



The 2021 releases side-by-side: Raspberry Pi's Pico and Zero 2 W.

So people ask us, "Can we have another Zero?" The problem that we have with the Zero is that design is very dependent on that SoC that we use, the 2835 SOC that we use, which has a PoP package. You have the SoC and then you have the DRAM stacked on top of it. And we don't have any other PoP chips. If you look at a Raspberry Pi 2, 3, 3+, 4, they all have external memory, which is mounted on the PCB. There is no room on that PCB. It used to be, if you want to stay single-sided or really actually I think even a double design would be quite challenging to achieve.

So we kind of backed into a corner and it took us a long time to figure out how to get out. And Zero 2 is the product that is enabled by RP3A0, which you'll see is written on top of the chip. That's not silicon innovation, that's packaging innovation. So effectively what that is is a BCM271081 die, the same piece of silicon that's used in Raspberry Pi 3, put inside a package with an LPDDR2 memory die to make a finished object. And then on the outside it has our logo and RP3, and that's what it is.

So what you're holding is a slightly downclocked 1GHz versus 1.2GHz Raspberry Pi 3 with half as much memory and a smaller set of accessories around it on the board. And it's 15 bucks, we have actually had to grow the price in order to do that. But it's about 10 times the performance, for a multithreaded benchmark, of a Zero.

Make: Pi 3 was where people started looking at Raspberry Pi as not just an educational product, but something that could do serious work. You could turn it potentially into a desktop device that you use daily. And to go from that to this size, for 15 bucks ...

The Raspberry Pi 400 is a complete computer inside a keyboard casing.

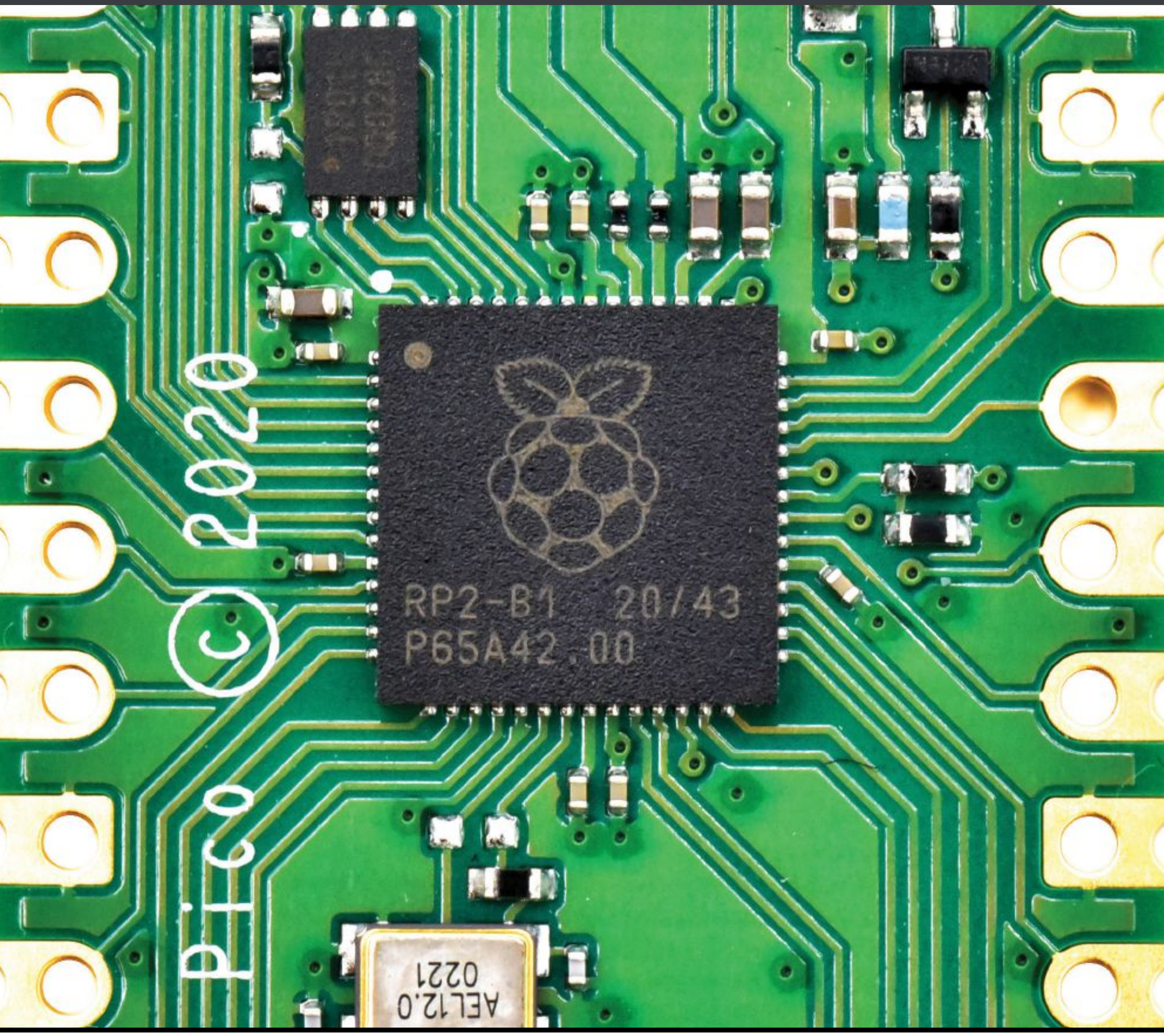
Upton: Yes! It's little, it's a lot of fun. We've always called ourselves a PC company, we've always aspired to be a PC company. And they've always been viable PCs for some subset of the things people do with PCs. Raspberry Pi 4 is the one that finally delivers for all but the most hardcore users. But it's a PC, so you can run a web browser on it very comfortably. Raspberry Pi 3 is probably the one that delivers the same thing, but without the web browser. So pretty much anything else that you want to do, as long as it isn't running that one, horrible, memory-hungry, processor-hungry application. Pretty much anything else you want to do, you can do on the Raspberry Pi 3. And that's why it's exciting to bring that level of performance to the Zero family.

Watch the full interview with additional details and more at makezine.com/go/raspberrypi2021

PIO PRIMER

Learn how to access the Programmable I/O on the **Raspberry Pi RP2040**

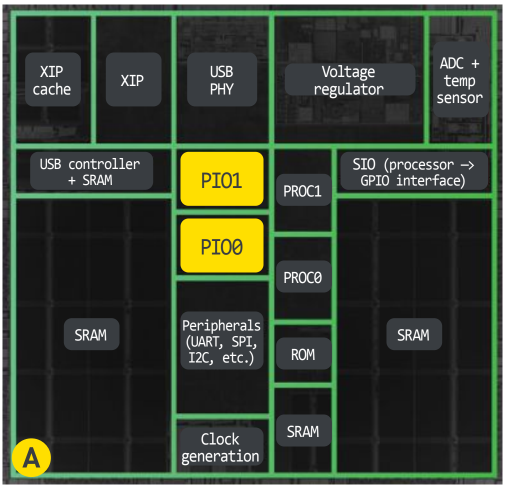
Written by Paul J. Henley

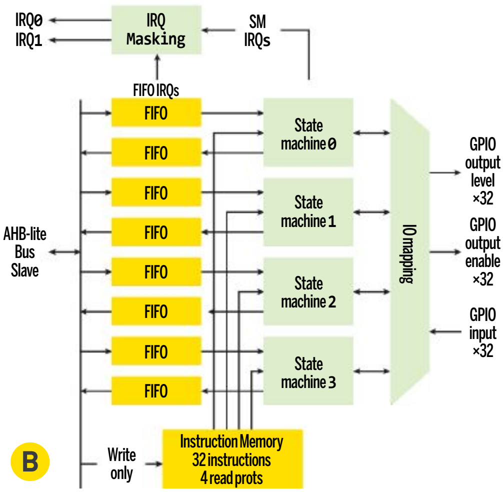


PAUL J. HENLEY is a former math and science teacher. He now writes and makes electronic devices for Four Bit Industries (fourbitindustries.com).

One unique feature of the new RP2040 microcontroller from the Raspberry Pi Foundation is its Programmable I/O (PIO) peripheral. The focus of PIO is serial communication. Most microcontrollers have hardware support for popular serial protocols such as I²C, SPI, and USART. However, this hardware support is always limited both in the number of serial interfaces and the types of serial interfaces that can be used. If you have a non-standard serial interface you need to support, you may have to resort to bit-banging I/O pins to implement it, tying up the microcontroller core to get the timing right. PIO aims to solve this problem by providing a highly configurable, programmable I/O peripheral that will take care of the bit-banging and provide simple input and output FIFO queues to the microcontroller core.

The RP2040 features two PIO blocks (Figure **A**), and each has four state machines. Each state machine has its own input and output FIFO queues for communicating with the ARM cores, and can operate on any GPIO pins (Figure **B**). Each PIO block has memory to hold 32 instructions, with all four state machines reading from the same memory. Hence you can create four instances of the same serial interface on four sets of pins using one PIO block. The instruction set for the state machines consists of just nine instructions, but with flexible parameters aimed at facilitating well-timed I/O.

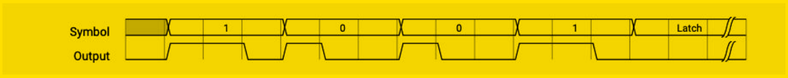




WS2812 LEDS

WS2812 LEDs are driven by a proprietary pulse-width serial format, with a wide positive pulse representing a "1" bit, and narrow positive pulse a "0". Each LED as a serial input and a serial output; LEDs are connected in a chain, with each serial input connected to theprevious LED's serial output.

LEDs consume 24 bits of pixel data, then pass any additional input data on to their output. In this way a single serial burst can individually program the color of each LED in a chain. A long negative pulse latches the pixel data into the LEDs.



C WS2812 line format. Wide positive pulse for 1, narrow positive pulse for 0, very long negative pulse for latch enable.

PROGRAMMING THE PIO

Although programs for the PIO block are written in the PIO's assembly language, those programs can be embedded in either MicroPython or C/C++ programs. Example code for the Pico is provided in the Raspberry Pi Pico Python SDK and the Raspberry Pi Pico C/C++ SDK, with full documentation of PIO assembly in the RP2040 Datasheet (datasheets.raspberrypi.org/rp2040/rp2040-datasheet.pdf) (These datasheets also supply the tables used in this article). Adafruit has provided an introduction to PIO in CircuitPython in their Learning System (learn.adafruit.com/intro-to-rp2040-pio-with-circuitpython). Although support for the Raspberry Pi Pico has been added to the Arduino IDE, there is no support yet for programming the PIO blocks and few libraries support it.

Let's look at one example application of the PIO that demonstrates what it can do, following github.com/raspberrypi/pico-examples/blob/master/pio/ws2812/ws2812.pio. WS2812 LEDs, also known as Neopixels, use a one-wire serial interface where the lengths of the pulses are used to indicate ones and zeros (Figure **C**). I've never encountered a microcontroller with hardware WS2812 support and because the timing of pulses must be precise, the PIO's ability to support these LEDs is welcome. We will look at just the PIO assembly code to get a sense of what it is like. There is much more code not shown that configures the PIO and its default behaviors (Figure **D**).

Starting at line 20, we see an *out* instruction. In this case the instruction pulls *1* bit from the output shift register and sends it to the *x* scratchpad register. There is an output shift register (*OSR*) and an input shift register (*ISR*) for each state machine. These registers keep track of how many bits have been shifted in or out and can be configured to automatically move data to or from the FIFO queues. There are also two scratchpad registers called *x* and *y* that are typically used for counters. We also see two additional parameters in this line. The *side* parameter indicates a constant bit or bits that are to be sent to a previously specified *side-set* pin or pins concurrent with the execution of the

D PIO ASSEMBLY CODE

```
18 .wrap_target
19 bitloop:
20       out x, 1               side 0 [2] ; Side-set still takes place during delay
21       jmp !x do_zero         side 1 [1] ; Branch on the bit we shifted out.
22 do_one:
23       jmp bitloop            side 1 [4] ; Continue driving high, for a long pulse
24 do_zero:
25       nop                    side 0 [4] ; Or drive low, for a short pulse
26 .wrap
```

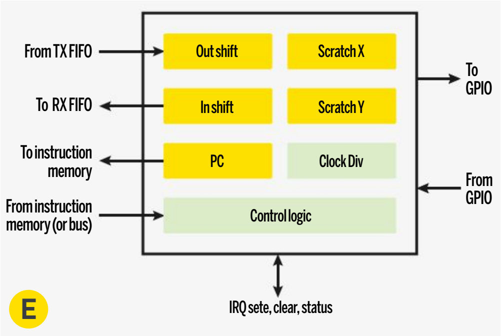
Raspberry Pi, Miroslav Němeček, Mike Senese instruction. In this case we are setting a single pin low. The number in brackets is a delay. It instructs the state machine to wait an additional two cycles before moving to the next instruction. The length of a PIO instruction cycle is determined by the state machine's clock divider, which is set when the state machine is configured (Figure E).

As you can see, a lot can happen in one instruction. Note that because of the way the WS2812 protocol works, the data from the output shift register is never sent to a pin. In more conventional serial interfaces, the *out* instruction would shift data to a pin (or pins) and the *side* parameter would set something like a clock signal on another pin.

Continuing with the code, line 21 performs a jump based on the bit value that was moved into *x*. At the same time, it sets the *side-set* pin high and delays for an additional cycle. If the bit moved into *x* is a zero, the program jumps to line 25 where the output is set to zero for five cycles, making the pulse short. If the bit is one then the program proceeds to line 23 where the output is kept high for five cycles, making the pulse long, before jumping back to the beginning. The directives *.wrap* and *.wrap_target* tell the state machine where the code ends and should restart. It is a special zero-cycle jump.

APPLICATIONS

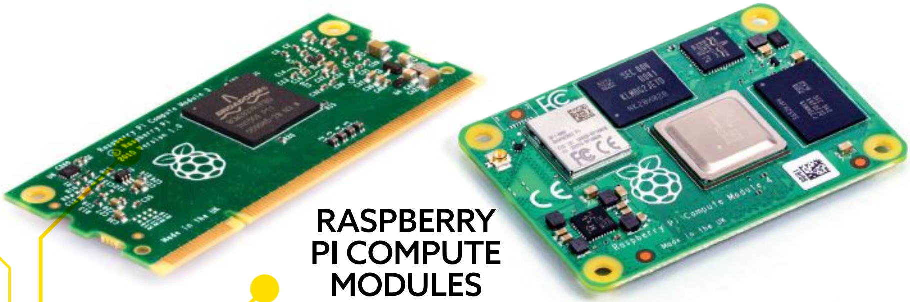
So what kind of interfaces are possible with PIO? For starters, they don't really have to be serial. You can specify that the *out* instruction shifts data to a contiguous set of up to 32 pins, and likewise for *in*. (Note only 30 pins are available in the RP2040's current package, and only 26 are broken out on the Pico.) The datasheet notes that you can implement a parallel bus to communicate with 8080 or 6800 microprocessors. Makers have used output pins connected to a resistor ladder to generate VGA output from PIO (breatharian.eu/hw/picovga/index_en.html) (Figure F). A library available through Arduino IDE charlieplexes LEDs through the PIO (github.com/pierremolinaro/rp2040-charlieplexing). But if you just want more I²C, SPI, or UART interfaces, the PIO can do that too. If an I/O task is simple, repetitive and would tie up the processor, it is a candidate for off-loading to a PIO block.



State machine overview. Data flows in and out through a pair of FIFOs. The stated of machine executese a program which transfers data between these FIFOs, a set of internal registers, and the pins. The clock diveder can reduce the state machine's execution speed by a constant factor.

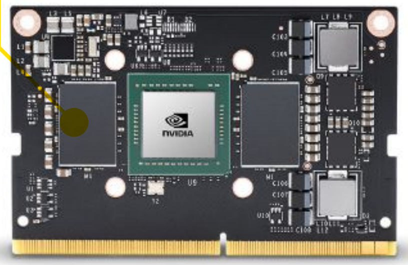
MIGHTY MODULES

If you're taking your microcontroller or SBC project to market, these **pluggable boards** might help get your design built **Written by Tim Deagan**



RASPBERRY PI COMPUTE MODULES

PARTICLE SOMs



NVIDIA JETSON

SPARKFUN MICROMOD

TIM DEAGAN likes to transform things from the digital world into real life in Austin, Texas.

There comes a point in a professional hardware designer's journey where a dev-board-based project needs to turn into something that can be viably manufactured. The form factor of a typical Arduino or Raspberry Pi dev board can present challenges in this regard. But designing full, custom circuitry can be daunting too, especially if it needs guaranteed processing functions. Thankfully, there exists a middle ground: Using vendor produced modules with custom breakout boards.

The IO and computing needs of your product are only one of the design considerations when sending products into the wild. How large will it be? Does it need cooling? Is weatherproofing essential? Does it have to be field-serviced? What are the power requirements? Each of these, and a host of other concerns, drive the choice of where to strike a balance between what you build and what you buy for use in a product. Boards like Arduino Uno or Raspberry Pi Model B are extremely well understood, but can be frustrating for use in productized designs. Adding Wi-Fi to most Arduinos generally requires external components, hence the common use of ESP8266's in commercial designs. Raspberry Pis rely on SD cards that add additional points of failure or attack surfaces to a system.

Ideally, a designer will custom develop only those aspects which represent the unique value proposition of their product, and combine their custom board with commercial products providing just enough, but not too much more than their minimum needs. In some cases, the custom PCB is small enough that it can be a daughterboard (e.g. a "hat" or "shield") to the processor; in other cases, the processor may be considerably smaller than the custom PCB and attach to it as a module. At the high end of production design, processing power may be a commercial chip soldered to the custom PCB.

A third-party board can take care of a wide variety of commodity functions: processing, networking, storage, clock, power management, GPS, etc. Few of these functions make sense to custom develop. Determining the needs of your design will drive the product selection. Why pay for an on-board GPS if your product doesn't need location services?

Form factor and power usage are also major drivers in choosing the right component. Will the custom PCB be larger or smaller than the processor board — who's attached to whom? Can the product's form factor support boards stacked via GPIO pins and headers, or does it need to stay flat and require edge connectors or ribbon cables?

Let's take a look at some system on module (SoM) offerings that have achieved market success: Raspberry Pi Compute Module, Sparkfun MicroMod, Particle SoM, and the Nvidia Jetson series. Many other options exist, but these boards represent major vendors all providing different approaches to product integration.

Adobe Stock-ahmadwahyu27

RASPBERRY PI COMPUTE MODULES

The Raspberry Pi (raspberrypi.org) series of single-board computers have changed the landscape of embedded development by providing a full and common operating system (OS), Linux, on a platform the size of many microcontroller boards. With USB ports, a wealth of GPIO pins to interact with the world, HDMI, and extensibility via Hardware Attached on Top (HAT) boards, developers can leverage decades' worth of Linux apps and toolsets to bring PC-level power to embedded problems.

Last year 10% of the boards Raspberry Pi sold were Raspberry Pi Compute Modules (CM), a smaller form factor version of the Pi with on-board memory instead of an SD card, pins but no ports for interfaces like Ethernet and HDMI (Figure A), and fewer GPIO pins. The company forecasts CMs will jump up to 15% of boards sold in 2021. Users are expected to create a custom carrier board to expose needed ports, or have to use one of the Pi IO boards. Targeted at industrial and commercial developers, the CM3 and CM4 have proven extremely popular for "deep embedded" applications.

Important features for these markets are often unfamiliar to makers. The CM4 offers asymmetric secure booting that cryptographically confirms each step of the boot process as protection against malware. This may be merely interesting to DIY users, but it's critical to a large IoT-installed base. Additionally, Raspberry Pi has invested in creating an Integrator Programme that connects users to a dedicated team at UL to navigate the complicated world of testing and paperwork required to meet international compliance standards, an essential requirement for many products.

You'll now find CM boards inside various consumer products, such as NEC digital signage displays (Figure B), Sfera Labs' industrial process controls and monitors (Figure C), and the CutiePi tablet (cutiepi.io) (Figure D). There are also loads of baseboards that you can buy, insert your own CMs, and build into entirely new projects. The Turing Pi 2 (turingpi.com) (Figure E) and StereoPi (stereopi.com) (Figure F) options are two good examples.

PARTICLE SOMs

Particle (particle.io) hit the market with a very clear target of becoming a premier IoT vendor. Their offering provides an end-to-end ecosystem consisting of hardware, OS, connectivity, and cloud infrastructure. Edge computing applications can be rapidly developed and deployed with an out-of-the-box suite of communication protocols (Wi-Fi, cellular, and BLE), management tools, reporting, and security, without assembling pieces and parts from a variety of vendors.

Particle has invested in developer kits and provides free plans for up to 100 devices, which makes for a well-documented and engaging entry path for developers. Nevertheless, for use in applications that rely on a single device, especially for DIY makers, much of Particle's investment in handling large numbers of devices, advanced cloud computing, IoT security, and industrial hardening may not provide much value. But if you're interested in learning how to build professional IoT solutions or you're planning a set of devices that are going to live in the wild and communicate back home, all of these attributes, and the well-thought-out integration between them, shine.

Particle offers a broad mix of hardware offerings that provide various radio, processor, memory, and form factor configurations.

Choosing the right board depends on the specific needs of your application. A Particle developer kit combined with their free-data plan offers an amazing value for a comprehensive edge-computing introduction.

Many consumer companies are now using Particle controllers as part of their IoT solutions, including Jacuzzi (Figure G), Watsco (HVAC) (Figure H), Scientific Aviation (methane emissions for oil and gas) (Figure I), and Dynamis (personnel tracker for the Air National


J


K

Guard) (Figure J). If you're looking for a breakout board for Particle SoMs, MikroElektronika has created an option that connects to their Click boards: mikroe.com/click-shield-for-particle-gen-3 (Figure K).

NVIDIA JETSON

Nvidia (nvidia.com) is perhaps best known as a premier graphics card vendor but their Tegra System on a Chip (SoC) processor has been an important business line for them. The Jetson line of products offers a range of Tegra-based boards intended for embedded AI development. The most recent of these boards, the Nano, is priced specifically to target the hobbyist and educational market, while the more powerful options (such as the Jetson TX2 and NX/AGX Xavier) cost quite a bit more but bring serious professional-grade performance to the table.

Jetson Nano has many specs in common with Raspberry Pi. Processor, memory, I/O, and display are almost the same. Both platforms run a Linux distro; for Nano, it's Nvidia's Ubuntu-based Linux4Tegra. The biggest fundamental difference is that the Nano has a 128-core Maxwell graphics processing unit (GPU) at 921MHz that's beautifully suited to handle AI, deep learning, and machine vision applications.

To harness the complexity and power of the Jetson boards, you'll need to develop with Nvidia's JetPack SDK. This collection of libraries, APIs, examples, and developer tools is the same Linux-based suite of components used by Nvidia's professional customers. While developers coming from an Arduino rather than Linux background may find the collection of components initially daunting, Nvidia has prepared a strong set of examples, tutorials, and projects to help bring devs up to speed.

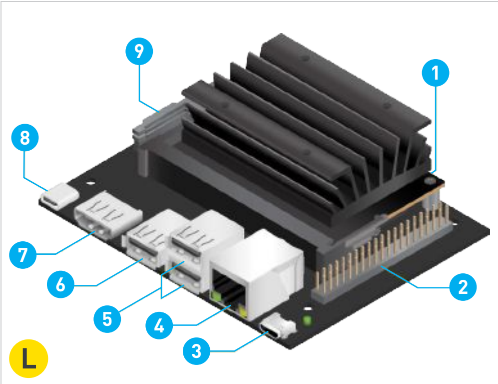
Jetson and JetPack SDK put a stunning amount of power for AI and computer vision projects in the hands of makers, allowing for everything from robots that can tell weeds from crops, to cars that can do real-time pothole identification, to drones that can detect fires.

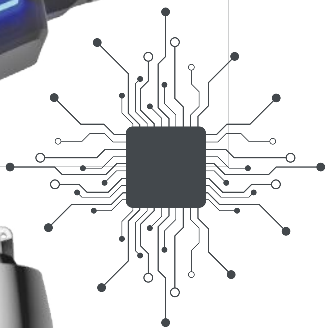
Most makers will first interface with one of the Nvidia developer kits, which consists of the Jetson module and a carrier board that breaks out all its features into easily accessible ports and pins (Figure L). There are dozens of third-party carrier boards if you're looking for more specific interfaces.

Many consumer devices use Jetson modules, such as autonomous drones from Skydio (Jetson TX2) (Figure M) and Coretronic (Jetson Nano and NX). The Staige K2 soccer field camera (soccerwatch.tv) (Figure N) is powered by the Jetson TX2 as well. And that's just scratching the surface; find a comprehensive list at developer.nvidia.com/embedded/community/ecosystem.

SPARKFUN MICROMOD

SparkFun (sparkfun.com) takes a different aim at the commercial and embedded market. Their MicroMod ecosystem provides a mix-and-match set of boards intended to speed up prototyping and proof-of-concept work. When time-to-market is the essential driver for a company, getting stuck or making mistakes in the prototyping phase can be fatal. MicroMod provides modular processors and sensors that can be used with different carrier boards (Figure O). The processors all use standard M.2 connectors and can be easily swapped out to allow rapid



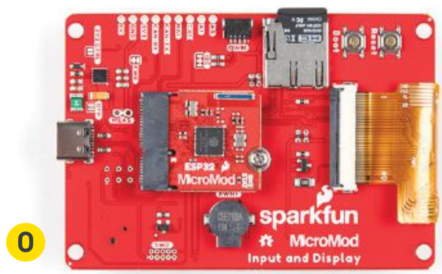


comparison of the different capabilities.

Current processor boards include ESP32, Artemis, SAMD51, Teensy, nRF52840, STM32, and RP2040 versions. MicroMod carrier boards offer configurations for data logging, machine learning, input and display, and ATP ("All the Pins"). Integration with SparkFun's I^2C Qwiic Connect System provides a simple 4-pin JST-connectable and chainable method to add over 100 sensors, shields, LCDs, relays, and other accessories.

While targeted at the prototype phase of the product development cycle, MicroMod is potentially appealing to DIY makers who'd like to experiment with microcontrollers using a modular toolset. The incredible number of permutations available from 7 processors, 7 carrier boards, and more than 100 sensors means that users are able to easily try lots of approaches without extensive soldering or breadboarding — think of this like the microcontroller version of SnapCircuits.

SparkFun provides guides on how to create both MicroMod carrier boards and processor boards, at learn.sparkfun.com/tutorials/designing-with-micromod. They also offer a SparkFun MicroMod DIY Carrier Kit (sparkfun.com/products/16549) for those making their own MicroMod carrier boards.



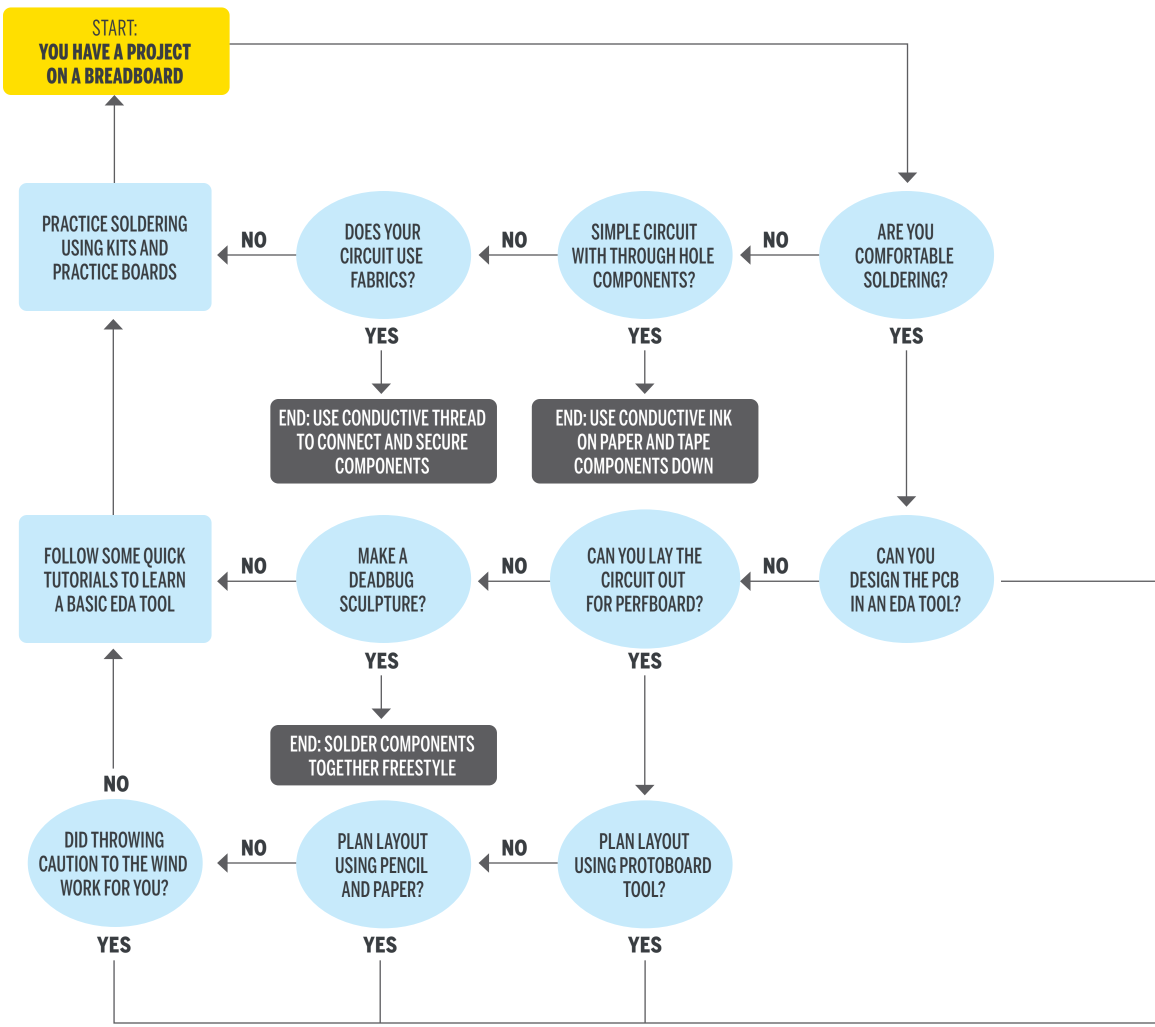
• • •

Each of these different offerings represents somewhat different paths beyond the Arduino and Raspberry Pi workhorses. SparkFun MicroMod makes me want to play. Raspberry Pi Compute Modules make me want to get to work and churn out products. Particle's boards make me want to get VC funding for a world-dominating IoT solution, and Nvidia's Jetson line makes me want to come up with a wild new art/robotics/AI project that breaks new ground. There have never been so many great options for scaling up SBC and microcontroller development!

CIRCUIT BOARD (RE)FLOW CHART

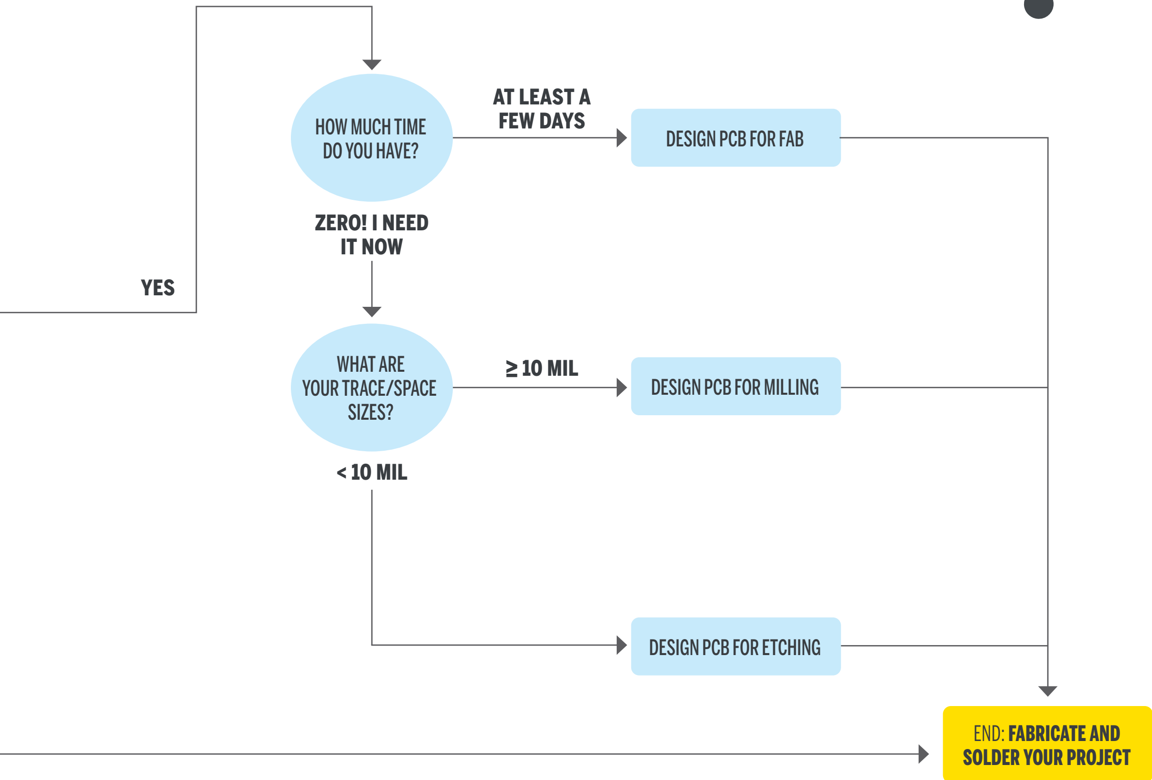
So you've breadboarded a project ... now what?

Written by Daniel Hienzsch

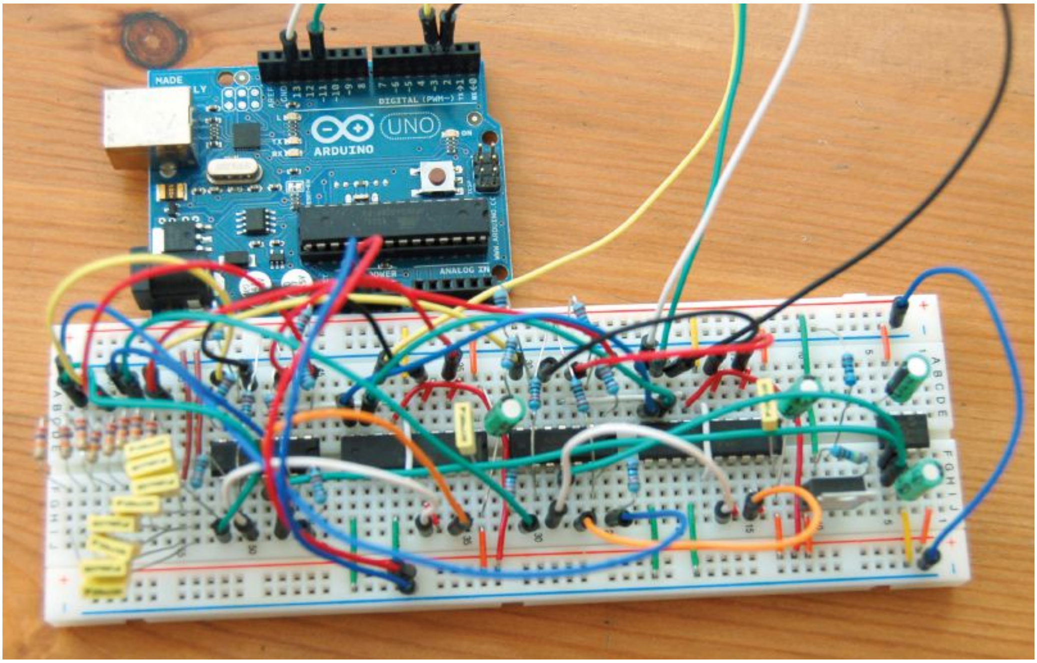


DANIEL HIENZSCH is a Las Vegas-based electrical engineer who started with a breadboard and a 555 timer just like everyone else. Through his online tutorials and in-person training, he loves sharing his knowledge of electronics and fabrication with hungry minds, and runs a consultancy to help demystify hardware for product design teams. You can find him at rheingoldheavy.com

With tremendous care and patience, you've constructed a masterpiece on your breadboard, functionality built on hours of component swapping, jumper wire connection, and painstaking troubleshooting. Your circuit deserves to be preserved! Let's have a look at what methods can make your project permanent, and some related tools, materials, and notes that can level up your circuit-making skill set.



START HERE: YOU HAVE A PROJECT ON A BREADBOARD



1. ARE YOU COMFORTABLE SOLDERING?

On a circuit board, solder connects electronic components to copper traces in a way that is both low resistance and structurally sound (Figure A). A good solder joint ensures excellent conductivity while protecting the circuit from motion, vibration, knocks, drops, and general clumsiness.

I've taught basic soldering classes for years and the #1 concern for beginners is breaking a sensitive component by touching a hot soldering iron to it. I can promise you that if you use a temperature controlled soldering iron and lots of flux, you'll find soldering a lot easier, and those components can take more heat than you think! Don't let surface mount components intimidate you either. A small magnification visor, some solder paste, a stencil, and a reflow system makes it easy! And actually faster than soldering through-hole equivalents.

It's still not for everyone or every project though, so while soldering is usually the route most will take, there are other solutions available.

➜ **IF YES:** Go to Note 2

➜ **IF NO:** Go to Note 6

2. CAN YOU DESIGN THE PCB IN AN EDA TOOL?

EDA means ***electronic design automation*** — it's basically CAD for electronics. Usually, the tool is broken into two major halves: in the first half you design your circuit by creating a logical diagram called a schematic, and in the second half you design the circuit board as the physical output of the logical design. There are loads of these tools out there, from the free and open-source variety to the high-end corporate enterprise flavor.

I've recommended KiCad (Figure B) to engineers for years, primarily because it's free and open source, has a supportive and active community of users and a fantastic suite of tools. It's supported by CERN after all.

All of these tools have a learning curve, but once you climb that curve, a world of capability opens up to you! Set aside time to build some PCB layout skills, and your electronics will thank you the rest of your life.

➜ **IF YES:** Go to Note 3

➜ **IF NO:** Go to Note 9

George_P._Macklin - wikipedia, Soldering-PCB-a (by Tlapicka - wikipedia, Kicad-PCBnew (by Pparent - wikipedia, OSHPark PCB (by Nathan Duprey - flickr

3. HOW MUCH TIME DO YOU HAVE?

Rapid prototyping is called "rapid" because it's fast, not because it's low cost. "Like the sign says, speed is just a question of money." Sending a board design out for fabrication is going to take at least a few days, unless you spend a fortune for quick-turn service, in which case you probably aren't just trying to convert a breadboard design to a permanent PCB. In most cases, that's at least a week including shipping time. You get a solid, professional level result that will withstand all the abuse you can throw at it. And while you're waiting for the board to come back from the fab, do what I do: start working on the next design! You also get silkscreen and solder mask, which are helpful for circuit protection and labeling and add layers for which you can explore creative design ideas.

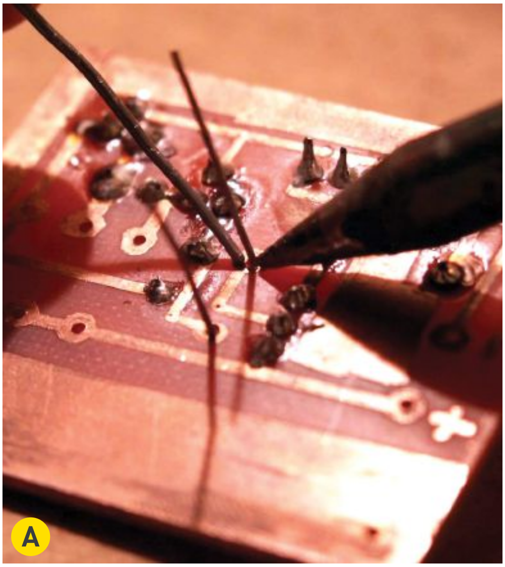
PCB Manufacturers:

- OSH Park (Figure C)
- Digi-Key DKRed service
- PCBWay
- JLCPCB

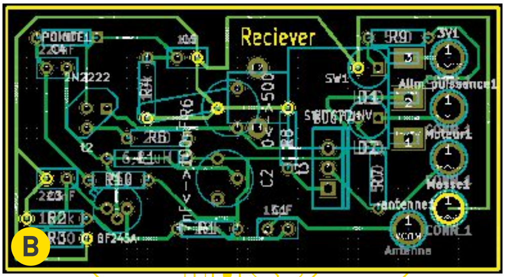
Find Inspiration:

- Boldport PCB designs
- Elkei Education PCB designs
- DEF CON badges

➔ **A LITTLE BIT OF TIME:** Go to Note 4
➔ **A LOT OF TIME:** Go to Note 5

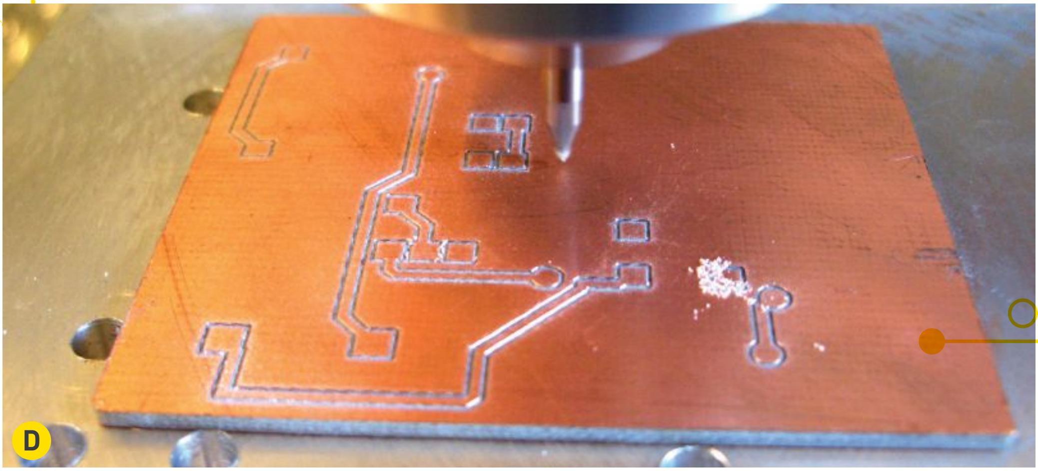


A



B

C



D

4. DESIGN PCB FOR MILLING

PCB milling, called ***isolation milling***, is a process that uses CNC equipment to cut away areas of copper from a piece of circuit board, leaving the conductive traces of your design behind (Figure D). With a little practice, this is hands-down the fastest way to a circuit board.

I have a Bantam Tools PCB milling machine that can mill a perfectly functional board in 15 minutes instead of waiting days. Of course, I had to buy the mill, the cutting tools, and the copper clad board stock, and develop design practices compatible with the fabrication process.

The primary drawbacks are a lack of solder mask and plated through-holes. You'll have to work with those constraints, and I've spent a lot of time soldering jumper wire to either side of non-plated vias.

On the plus side, you have a CNC mill and they're good for all kinds of stuff besides PCBs.

→ **NEXT:** Go to Note 15

Milling Machines:

- Bantam Tools
- LPKF
- Inventables X-Carve
- Carbide 3D Shapeoko
- FlatCAM software

Guides:

- "KiCad PCB Design for Rapid Prototyping" talk: youtube.com/watch?v=Yzq3lTg0yKo
- PCB milling video playlists: youtube.com/c/WEGSTRCNC/videos

5. DESIGN PCB FOR ETCHING

Chemical PCB etching is yet another circuit board fabrication method, and it still excels beyond any other home fabrication method in enabling designs that require thin traces and tight tolerances. If you have trace widths below 10mil / 0.25mm or are working with teeny QFN surface-mount parts, then etching will produce tighter and cleaner layouts than milling.

Chemical etching uses copper clad with a masking etch resist to protect your circuit from the ferric chloride etchant, which will eat away the copper from the rest of the PCB, leaving your traces behind (Figure E).

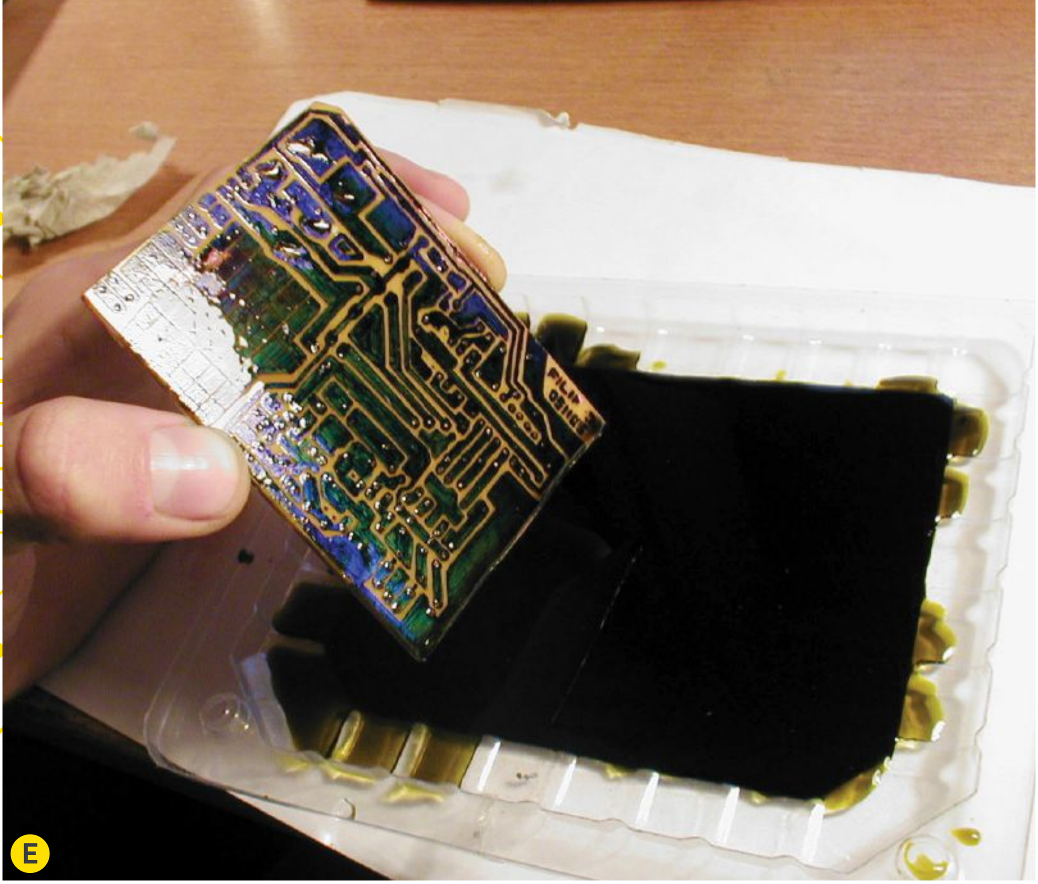
The real limitation on the quality of the etching is the resolution of the laser printer making the mask, and all laser printers have exceptional resolution these days. The PCB etching process supports incredibly fine detail and is well suited to a single board you want to take your time beautifully hand crafting.

As with PCB milling, you don't get solder mask and you don't get through-hole plating, in fact you don't even get holes — you'll need to drill those yourself. You also have to deal with the storage and eventual disposal of the ferric chloride etchant and any wash solutions you use.

→ **NEXT:** Go to Note 15

Etching Materials and Resources:

- MG Chemicals
- Laser printer transfer paper
- UV-sensitized copper clad boards
- Photoresistive laminates



E

PCB Milling (by Jiskar on flickr, Filip Dominec wikipedia, bareconductive.com

Guidance and Inspiration:

- Circuit etched on glass: hackster.io/news/here-s-how-to-etch-a-pcb-right-onto-glass-84aff70148de
- "Etching Professional Printed Circuit Boards" from *Popular Electronics*, 1966): rfcafe.com/references/popular-electronics/etch-printed-circuit-boards-popular-electronics-march-1966.htm
- Fritzing etching guide: fritzing.org/learning/tutorials/pcb-production-tutorials/diy-pcb-etching

6. CAN YOU MAKE A SIMPLE CIRCUIT WITH THROUGH-HOLE COMPONENTS?

➜ **IF YES:** Use conductive ink on paper and tape down the components. CONGRATULATIONS, YOUR PROJECT IS COMPLETE.

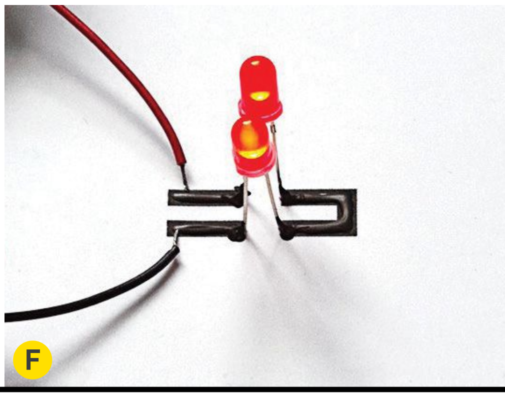
Conductive Ink/Tape Materials and Resources:

- Circuit Scribe
- Bare Conductive (Figure F)
- AgIC
- Conductive Copper Tape

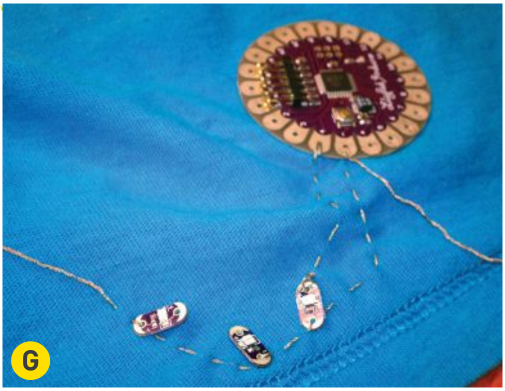
Find Inspiration:

- "Future with Bright Lights" video: youtube.com/watch?v=zRbQvEefhXc
- The Great Big Guide to Paper Circuits: learn.sparkfun.com/tutorials/the-great-big-guide-to-paper-circuits/conductive-ink-traces

➜ **IF NO:** Go to Note 7



F



7. DOES YOUR CIRCUIT USE FABRICS?

➔ **IF YES:** Use conductive thread to connect and secure components. CONGRATS, YOUR PROJECT IS COMPLETE.

Textile Circuit Resources:

- Silver-coated thread
- Stainless steel thread
- Threadwitch Light & Fabric
- LilyPad microcontroller
- Flora microcontroller

Find Inspiration:

- Lumen Couture video channel: youtube.com/c/LumenCouture/videos
- Sewable Circuits: learn.adafruit.com/lets-put-leds-in-things/sewable-circuits (Figure G)

➔ **IF NO:** Go to Note 8

8. PRACTICE SOLDERING USING KITS AND PRACTICE BOARDS

➔ **NEXT:** Return to Note 1

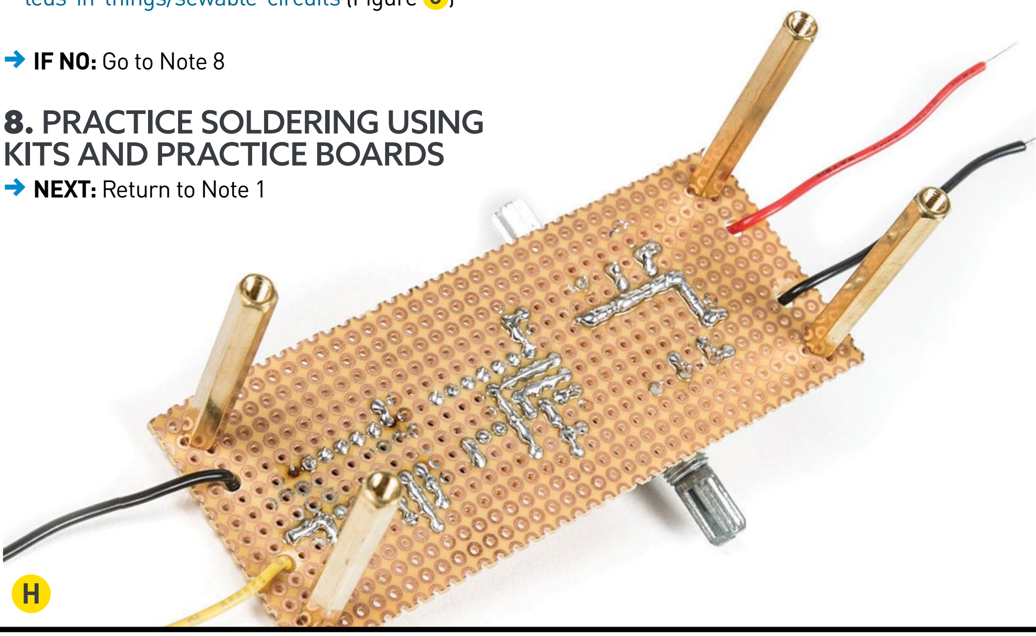
9. CAN YOU LAY THE CIRCUIT OUT FOR PERFBOARD?

Ahhh… perfboard / Veroboard / protoboard / stripboard: you either love it or hate it. Before the explosion in PCB fabrication services, this stuff and etching were your only real solutions for making a project permanent. There must be thousands of guitar effects pedals out there that have these circuit prototypes inside.

Protoboard is a piece of single-sided copper clad that has been pre-etched into a helpful pattern. The original form just provided individual copper plated holes that you soldered your component to, then hooked everything up with "hookup wire" (Figure H).

Modern layouts now include breadboard style, replicating the connective pattern of your breadboard, and stripboard, with long strips of copper that you either cut or solder together to form your circuit. If you get the form that emulates a breadboard, then layout is easy, because you just duplicate on perfboard what you have on your breadboard. More complex designs will probably go onto stripboard and will benefit from planning. That way everything gets connected the way it should.

There are even prototype boards that have a pinout along the edge that matches the standard Arduino Uno arrangement, making Arduino designs quick to assemble.



Christopher.nunez - wikipedia, Hep Svadja, Mohit Bhoite

Prototype boards are cheap and plentiful, and if you can get the hang of using them, are a great way to quickly solder up a project without needing a full EDA tool.

→ **IF YES:** Go to Note 10
→ **IF NO:** Go to Note 14

10. CAN YOU PLAN THE LAYOUT USING PROTOBOARD TOOLS?

→ **IF YES:** Go to Note 15
→ **IF NO:** Go to Note 11

Layout Tools and Resources:

- Fritzing
- VeeCAD Stripboard Editor
- LochMaster
- Arduino Proto Shield

Find Inspiration:

- Protoboard how-to: makezine.com/2015/10/15/how-and-when-to-use-protoboard
- GreatScott! YouTube channel: youtube.com/user/greatscottlab
- Arduino on protoboard: youtube.com/watch?v=vMu1UjBx0iQ

11. CAN YOU PLAN LAYOUT USING PENCIL AND PAPER?

→ **IF YES:** Go to Note 15

Layout Resource:

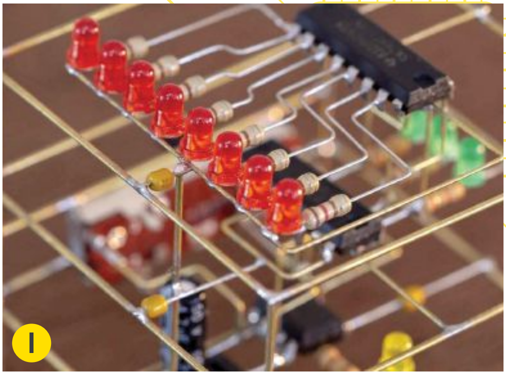
- Stripboard layout planning sheet: rheingoldheavy.com/stripboard-planning-worksheet

→ **IF NO:** Go to Note 12

12. DID THROWING CAUTION TO THE WIND WORK FOR YOU?

→ **IF YES:** You're an animal! Just decided to solder together an unplanned freeform circuit on prototype board, and it worked? Wild! You certainly don't need any advice from me! Go to Note 15.

→ **IF NO:** Go to Note 13



13. FOLLOW SOME QUICK TUTORIALS TO LEARN A BASIC EDA TOOL

EDA Learning Resources:

- KiCad
- Fritzing
- Autodesk EAGLE
- EasyEDA
- Contextual Electronics

→ **NEXT:** Return to Note 8

14. WOULD A DEADBUG SCULPTURE WORK?

→ **IF YES:** Solder components together freestyle. CONGRATS, YOUR PROJECT IS COMPLETE.

Deadbug Materials and Resources:

- Solid core wire
- Copper clad board
- Mohit Bhoite: bhoite.com/sculptures (Figure I)
- Manhattan-style circuits

Find Inspiration:

- Sculpt Tronics on YouTube: youtube.com/channel/UCiAj75VejDI_tlVix2QAHLA
- Deadbugs on *Make:* makezine.com/go/deadbugs

→ **IF NO:** Go to Note 13

15. FABRICATE AND SOLDER YOUR CIRCUIT.

CONGRATULATIONS, YOUR PROJECT IS COMPLETE.

MEET THE BOARD MAKERS

These two companies are helping make your projects smart with **FPGAs** and **SBCs**

Written by Mike Senese

KRTKL

- **Website:** krtkl.com
- **Key People:** Ryan Cousins (co-founder, CEO), Jamil Weatherbee (co-founder, CTO), and Russell Bush (co-founder, CDO)
- **Number of staff:** 5
- **Founded:** 2014
- **Location:** San Francisco
- **Key Products:** snickerdoodle FPGA boards, piSmasher and breakyBreaky carrier cards

Their story:

Krtkl CEO Ryan Cousins grew up exploring, adventuring, and building electronic kits with his dad. He went on to study mechanical engineering at UCLA, where he became interested in the entrepreneurialism that drove the successful people behind the companies he studied.

After graduating, Cousins worked in the medical devices field, and then in engineering consulting, where he and fellow coworkers Jamil Weatherbee and Russell Bush decided to launch a company focused on its own product. The team spit-balled ideas from drones to educational robots before having the realization that every time they built a product for a consulting client, they were essentially rebuilding the same custom controller components. Why not turn that into their product?

The trio decided the resulting controller board should be versatile, but affordable. They wanted it to fit alongside Arduino and Raspberry Pi, and be attractive to professional-level hobbyists and engineers, hoping "they might just pick this up because it's cool," says Cousins.

Their first product, crowdfunded in 2015, was the **snickerdoodle** board, using the Zynq FPGA SoC platform ("a perfect fit"). Since then they've updated their lineup to include **snickerdoodle one** and **snickerdoodle black** FPGA boards, along with **piSmasher** and **breakyBreaky** expansion boards. Dozens of products now run on snickerdoodle boards, including Scorbitron (scorbit.io), a pinball-networking device that lets distributed players compete against each other, track high scores and more, on machines of any age.

Moving forward, Cousins finds robotics and edge AI interesting opportunities for krtkl, but plans to keep things in the physical world. "I don't ever see the company turning into a cloud company. As painful as hardware is, we wouldn't do this if there weren't hardware. We build stuff, that's what's in our DNA."

Snickerdoodle FPGA mounted to a purple Scorbitron baseboard.

KRTKL
Key People

Ryan Cousins


Jamil Weatherbee


Russell Bush


HACKBOARD
Key People


Mike Callow


Jon Prove


Dean Kline

HACKBOARD

- **Website:** hackboard.com
- **Key People:** Mike Callow (co-founder, CEO), Jon Prove (co-founder, COO), and Dean Kline (VP, marketing and communications)
- **Number of staff:** 6
- **Founded:** February 2020
- **Location:** Austin, Texas
- **Key Products:** Hackboard 2

Their story:
Hackboard is a new board maker, launching with an Intel Celeron-based, Windows 10 Pro single board computer (SBC) called the **Hackboard 2**. The executive team, consisting of Mike Callow, Jon Prove, and Dean Kline, had previously worked together on different Raspberry Pi-based endeavors, including Kano and pi-top, and individually at technology companies like IBM, Element 14, and Dell.

Hackboard started when, due to the emerging Covid pandemic, Callow found himself stuck in China without a computer. Realizing he had access to components and pieces, he assembled a functional substitute to keep himself productive with work. Later relaying this experience to friend and former colleague Prove, the two realized there was an opportunity to create low-cost, fully capable computer options for students and communities that don't have access to pricier consumer devices.

Callow and Prove brought Kline aboard and put together a test run of Hackboard 1 SBCs to get market feedback. Part of what they learned was that some backers, while appreciating the Windows 10 Pro access, still wanted the tinkerability of Linux, so they added that as a crowdfunding option. The board also carries an M.2 hard drive for up to 512GB of storage, and 4GB–8GB of memory, along with the GPIO pinout for further customization.

The team says that part of making Hackboard 2 accessible is keeping the prices down; the base option with Linux costs $179, while a maxed-out version with Windows 10 Pro is $299. "Our mission is to get computing into the hands of as many people as possible, with the power that they need," Kline says. "And we know that to do that, there's probably some profitability sacrifice that has to happen. We really want to make a difference in the world at this stage in the game."


The Hackboard 2 single board computer.

Adobe Stock-ahmadwahyu27

HACK ATTACK!

When a new board drops, makers around the planet immediately start hacking new projects with it. Here are **17 fun projects** to try on fresh hardware.

RASPBERRY PI PICO

1 Add MIDI to Toy Keyboard

Kevin @diyelectromusic, UK

Upgrade a kiddie keyboard with a Pi Pico to send MIDI notes, via UART and USB MIDI at the same time. diyelectromusic.wordpress.com/2021/06/28/

2 Joke Texting Phone

Sai Yamanoor, Buffalo, New York

Hack a vintage touch-tone phone with a Pico and Blues cellular Notecard, then connect it to the Dad Jokes API to text lame jokes to whatever number you dial. hackster.io/diy-champs/pico-phone-c0f1eb

3 Automated Model Railroad

Kushagra Keshari, Jabalpur, India

Hack your tracks with a Pico, SparkFun motor drivers, and IR proximity sensors to choreograph precise, point-to-point locomotive locomotion. hackster.io/Tech_build/raspberry-pi-pico-controlled-point-to-point-model-railroad-80d961

4 Automatic Guitar Tuner

Guyrandy Jean-Gilles, Philadelphia, Pennsylvania

Build this robo-tuner inspired by Band Industries' Roadie 3. A Pico listens to an electret mic and drives a motor to spin each tuning peg up to perfect pitch. gitlab.com/guyjeangilles/automatic-guitar-tuner/-/tree/rev2

[+] *For cool RP2040 keypad builds, check out "All Hands on Deck," page 80.*

MICRO:BIT V2

5 Smart Tea Cozy

Kitty Yeung, Berlin, Germany

Physicist and tech-fashion designer Kitty Yeung sewed and wired this temperature-sensing timer cozy for brewing tea, coding the new micro:bit V2 in Make:Code. hackster.io/kitty-yeung/smart-tea-cozy-micro-bit-v2-72299e

[+] *For more about visual coding see "Skill Builder: Block Based Programming," page 116.*

ADAFRUIT NEO TRINKEY

6 USB Rubber Ducky

Dylan Herrada, Washington, DC

Code a NeoTrinkey with Adafruit's new Circuit-Python Ducky library to make it a "rubber ducky" — the hacker's notorious "bad USB" for injecting keystrokes into unwitting computers. learn.adafruit.com/neo-trinkey-circuitpython-rubber-ducky

[+] *For more about duckyScript, see "Hot Keebs" on page 72 and build a duckyPad macro keypad.*

7 Zoom Shortcut Buttons

Liz Clark, Boston, Massachusetts

Hide, mute, or just bail from that Zoom meeting with the touch of a real button. learn.adafruit.com/neo-trinkey-zoom-shortcuts

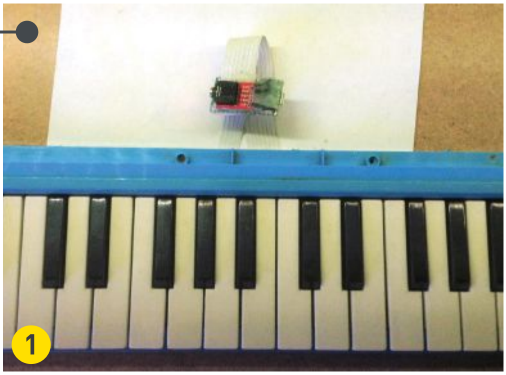
ARDUINO NANO RP2040 CONNECT

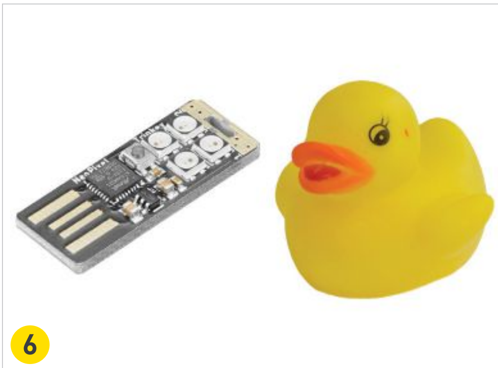
8 DIY Cocktail Mixer

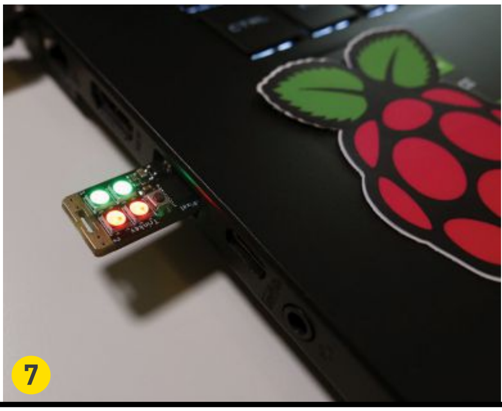
Jithin Sanal, Trivandrum, India

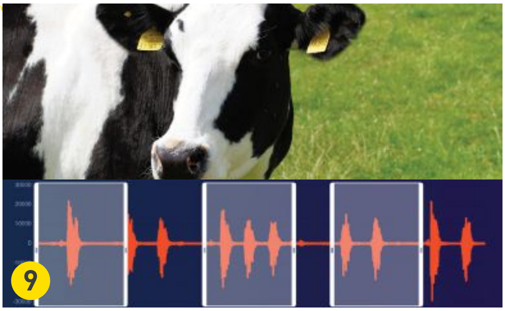
Tap your phone and this bar-bot will mix your

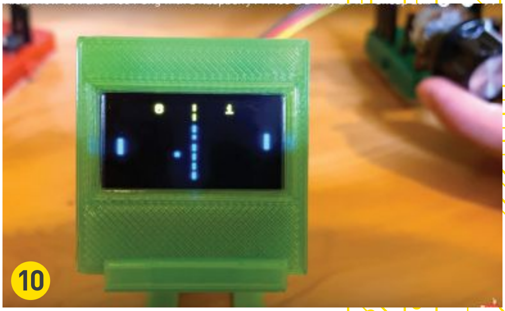
Kevin (diyelectromusic), Sai Yamanoor, Kushagra Keshari, Guyrandy Jean-Gilles, Kitty Yeung, Dylan Herrada, Liz Clark, Jithin Sanal, Adobe Stock-ahmadwahyu27







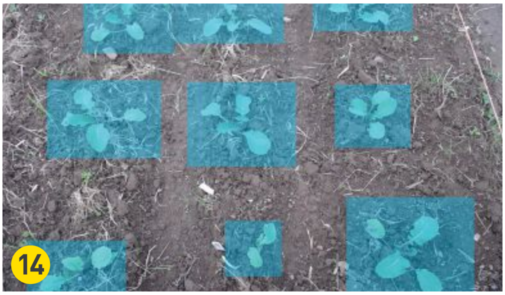

9

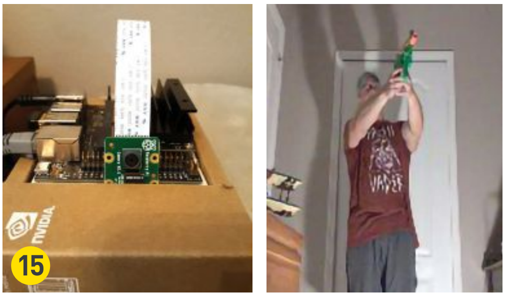

10

11


12

13


14



15


16

17

Francesco Azzola, Gurgle Apps, Tom's Hardware, Jernej Furman / focusonmore.com, SkyHub, Tegwyn☠Twmffat, Nick Bild, Varun Mehta, Adam Taylor cocktail to order, using the Nano RP2040 Connect to trigger peristaltic pumps, by way of the Arduino Cloud IoT platform. create.arduino.cc/projecthub/jithinsanal1610

9 Animal Sound Classifier

Francesco Azzola, Perugia, Italy

Cluck! Oink! Moo! Challenge your toddler to a duel with this animal noise identifier. Use Edge Impulse to train a machine learning model to classify critter samples, then deploy the model to your Nano RP2040 Connect. survivingwithandroid.com/arduino-rp2040-sound

PIMORONI TINY 2040

10 Tiny Pong

Aylesbury-Jarvis family, Harrogate, England

Breadboard two mini analog controllers using potentiometers, add a 0.96" OLED screen and speaker, and your Tiny 2040 becomes Tiny Pong, with excellent gameplay and sound effects. youtube.com/watch?v=WvZvtElvWlM

11 Smartphone Camera Remote

Les Pounder, Blackpool, England

Take better selfies and wildlife shots with a DIY remote shutter button for your Android phone camera. The trick is coding your Tiny 2040 to behave like a USB input device, using CircuitPython. tomshardware.com/how-to/raspberry-pi-pico-camera-button

MICROCHIP CURIOSITY NANO

12 IoT Contactless Thermometer

Greg Toth, Washington, DC

Scan those foreheads straight to the cloud — use an AVR128DA48 Curiosity Nano to read a contactless temperature sensor and send data to the Medium One platform for analysis. Maybe you'll spot outbreaks and superspreaders before it's too late. mouser.com/applications/iot-enabled-contactless-temperature-sensor

NVIDIA JETSON XAVIER NX

13 SkyHub UFO Tracker

Skyhub.org, Houston, Texas

Wow, and I thought the RasPi Meteor Network was cool (*Make:* Volume 77) — now there's a network for spotting and classifying UAPs. Connect a fisheye camera and Adafruit GPS module, then train your Jetson Xavier or Nano 4GB to recognize aerial anomalies, alien or otherwise, and report them to this global citizen science network for study. instructables.com/Sky-Hub-UAP-Tracker

14 Crop-Recognizing Agricultural Navigation

Tegwyn Twmffat / Goat Industries, UK

Farmbots that navigate by GPS can easily trample crops. Train your Xavier to see each individual plant, then watch your autonomous tractor tiptoe through the tulips. hackaday.io/project/165021-crop-orientated-agricultural-navigation-system

15 Doom Air

Nick Bild, Orlando, Florida

Demon hunters: Play *Doom* life-size, projected on the wall, with your body as the controller, by training a Xavier with CSI camera (a Pi Cam V2.1) to detect your gestures for run, jump, next weapon, shoot, crouch, and more. hackaday.io/project/166696-doom-air, github.com/nickbild/doom_air

SEEED SPARTAN EDGE ACCELERATOR

16 16-Bit Graphics with ESP32+FPGA

Varun Mehta, New York City

For video projects most of us would reach for a GPU or at least a CPU like a Raspberry Pi. Instead, this hacker tweaked his Spartan Edge Accelerator to use the FPGA as a GPU and the onboard ESP32 module as a CPU, to generate 16-bit graphics with sprites and x-y scrolling of multiple background layers, in 720p HDMI output. hackaday.io/project/175763-spartan-edge-accelerator-graphics-esp32fpga

17 MIPI Imaging on FPGA

Adam Taylor, Harlow, UK

And this hacker pulled a similar trick, using the FPGA to read and display HDMI video from a Pi Camera 3, with *no* processors. The secret sauce: clever Vivado design using IP blocks from Trenz, Seeed, and Diligent. hackster.io/adam-taylor/mipi-imaging-with-the-spartan-edge-accelerator-board-50cab1

Build a Mini Simpsons TV

Ay, caramba! There's always something good on — when you print, solder, and code this tiny tube

Written and photographed by Brandon Withrow

TIME REQUIRED: 4–5 Hours +10 Hour Print Time

DIFFICULTY: Intermediate

COST: $50–$100

MATERIALS

Buying through my Amazon affiliate links at withrow.io/simpsons-tv-build-guide would help me cover my time making this guide — at no cost to you! Consider it one back scratching another, or something like that.

» **Raspberry Pi Zero W, without header pins** This is important!
» **TFT LCD touchscreen display, 2.8", 640×480** iUniker Raspberry Pi Zero W Screen
» **Audio amplifier, 2.5W mono** Adafruit 2130
» **Audio speaker, 1.5", 4Ω 3W** Gikfun LYSB01LN8ONG4-ELECTR
» **MicroSD card, 64GB** SanDisk SDSQUA4-064G-GN6MA
» **Micro USB to DIP adapter board** Amazon B07W844N43
» **Micro USB male solder plug** Amazon B08SC94GP1 or B07G5ZY7MH
» **Trim potentiometer, 1kΩ** Amazon B07QX3BX9R, B07C3XHVXV, or B07Q8RSNP7
» **Switch, micro pushbutton** Amazon B07Y3C1MKR, B01MRP025V, or B07WSTFB47
» **Wire, 22 gauge, black/red pair**
» **3D printed parts** Download the free STL files from thingiverse.com/thing:4943159. I used 3 colors of Inland PLA+ filament: Purple (amzn.to/38lAaAi), Light Blue (amzn.to/2UWFmrc), and Black (amzn.to/3yoYiMX).

TOOLS

» **3D printer**
» **Soldering iron** with a fine point tip
» **Solder** I like 60/40 rosin core, around .032" diameter.
» **Hot glue gun** The bedazzler!
» **Hobby knife** Always cut away from yourself!
» **Snips**
» **USB flash drive with micro-USB adapter**
» **Slotted screwdriver, small**
» **Cyanoacrylate (CA) glue** aka super glue
» **Acrylic paint, dark purple**
» **Helping hands (optional)** such as QuadHands QH-WB-68. Not necessary, but makes soldering loads easier.

BRANDON WITHROW is a software developer and maker living in Kansas City, Missouri. A passion for animation led him into a career as a software developer. While working at Airbnb he developed Lottie, an open source animation library. In his free time he likes to build things in his workshop.

What is it? A working desktop TV that plays *The Simpsons* on loop. This project was born from a childhood spent in front of a TV, playing with Legos. I wanted to recreate the "always on" random nature of television, in a tiny desktop format. The videos are always "playing," even when the screen is off. Like the television in the ancient days before the internet, you just turn it on and watch whatever it gives ya.

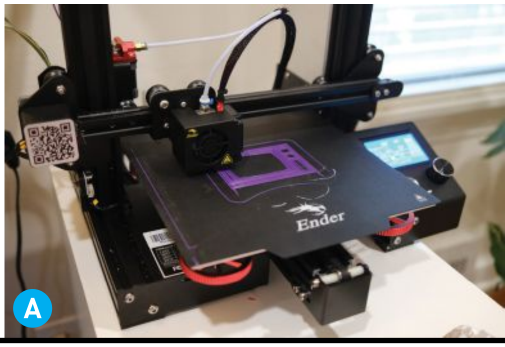
This guide will walk you through printing, building, and coding a small TV that will play videos (that you provide) at random. By the end of this guide you will have your very own tiny TV, with power and volume control. You can easily buy all the parts except the enclosure, which needs to be 3D printed. The TV is built with a Raspberry Pi microcomputer, running Linux. It might sound intimidating, but stick with me and it can be fun!

This guide is aimed at all skill levels but admittedly a few parts, like the soldering, are more intermediate than beginner. I believe you can do it, though — I've had reports from people completing the guide that this was their first time doing any of it! Pretty cool. The entire project, including print time, takes about 14 hours to complete.

1. PRINT THE PARTS

I designed the TV and parts in Fusion 360 and printed them on an Ender 3 Pro (Figure A). It's a good idea to start these prints and let them run while you set up the hardware. All of the parts together take about 10 hours.

- Filament — I used 3 colors of Inland PLA+ filament: Purple for the case, Light Blue for the knobs, and Black for the VCR/antenna — but have fun with it! If you print it in some fun colors, send me a pic at twitter.com/thewithra.



- Print settings — Almost all the pieces can be printed without supports, and with minimal infill. The only exception is the front piece. Print it with a minimal support and a low infill (10%). All pieces should be printed with their build plate orientation matching Figure B for best results. The larger parts I printed in Standard Quality (0.2mm), the smaller parts in Dynamic Quality (0.16mm). Get those pieces printing and then come back so we can get to work on the fun stuff.

2. VIDEO ENCODING

This could take some time. so it's best to start this early as well. Your videos — your completely legally-owned videos — must be encoded into a specific format and put onto a thumb drive to transfer them to the Pi later. We're using the H264 format with a height of 480 pixels. Other codecs would be better/smaller, but support of codecs on the Pi is limited with the player we are using.

NOTE: The videos should *not* be encoded directly on the Pi Zero, unless you want to wait until the year 2050 before you can watch them. The Pi Zero is a single-core computer with minimal processing power. It's great for an endless number of things, but video encoding is not one of them.

There are of course a hundred ways we could do this. I've written a convenient script that will automatically encode all the videos in a folder to the proper format. First you need to install the video encoder FFmpeg onto your computer. There are plenty of guides out there for installing it; check out their GitHub page github.com/adaptlearning/adapt_authoring/wiki/Installing-FFmpeg. Once you have FFmpeg installed, restart your computer and collect all your videos into a folder. Download my encoding script github.com/buba447/simpsonstv/blob/main/videos/encode.py and place it in the folder next to your videos.

Open terminal, navigate to the folder with:

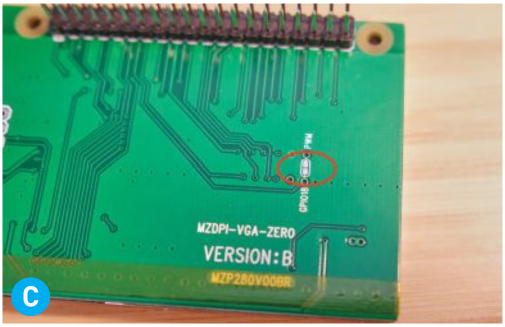
```
cd /path/to/the/folder
```

and run the script with:

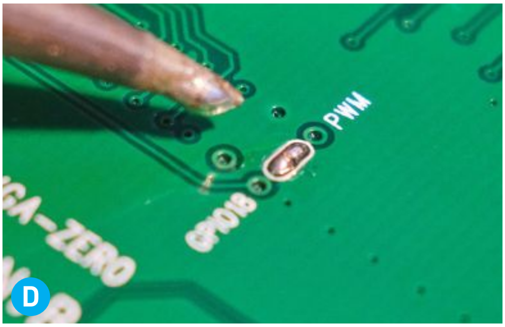
```
sudo python encode.py
```

The script encodes all the videos one by one, and places them in a new subfolder called *encoded*.

B



C



D

E

Be patient! Do not close this terminal window, and keep your computer from sleeping while it runs. When it's done, go ahead and move the *encoded* folder over onto your thumb drive. I'll explain later how to move the videos onto the Pi.

3. HARDWARE SETUP

In order to have TV we need to have a screen, so let's start there. We must perform a little bit of surgery first. The screen has a function to programmatically turn the backlight on and off. Unfortunately, that functionality is disabled by default. There's a jumper on the back of the screen that must be soldered to enable it.

We are going to make a quick solder. Plug in your soldering iron and warm it up. If you've never soldered before, don't fret — today you get to learn a new skill. I suggest getting a good set of helping hands; they really make it easier to solder small bits. Check out this beginner's guide to soldering (makerspaces.com/how-to-solder) and by the end of this project you will be a pro.

Lay the screen facedown on your work surface with the pins facing north. You'll see the jumper off to the right (Figure C). You're going to solder this jumper, connecting the two pads. Place the iron against the pads to heat them up. After a second, apply the solder. Remember, always apply the solder to the pads, not the iron. The finished solder should look like Figure D.

The next bit is the hardest bit of soldering. If you are newer to soldering maybe skip ahead to Step 6 for practice and come back here afterward. Don't worry, you'll be a soldering star in no time.

Place the Raspberry Pi Zero onto the 40-pin connector on the back of the LCD screen. It's important to keep the Raspberry Pi facing outward! I like to start by holding the corner of the Pi firmly in place and then quickly soldering the two leftmost pins (Figure E). I'm a southpaw like Stupid Sexy Flanders, so you might want to start with the two rightmost pins. Go slow and remember to heat the tiny pad and the pin with the iron, and apply the solder to the contacts, not the iron. Take breaks every few pins, you don't want to overheat the Pi board.

OK. Phew! Wipe the sweat off your forehead and get the shepherd's pie out of your knickers. You're a solder champion now.

4. INSTALL THE SOFTWARE

Before we set up the rest of the hardware components, let's boot up the Pi and make sure everything works.

I've been updating the guide for this build as people run into issues with the software. So follow my latest instructions at withrow.io/simpsons-tv-build-guide to prep your SD card, set up the Pi for headless boot, install the display driver, and connect to the Pi from your computer using SSH. Then you'll run a few command-line instructions to install USBmount (it allows us to easily mount our USB drive when the time comes), cleverly route the audio over a spare GPIO pin, and install OMXPlayer (a lightweight video player).

F

Finally, install my script that will play videos on loop and also read the input from the front buttons. Run these two lines:

```
cd ~/
git clone https://github.com/buba447/simpsonstv
```

5. MOVE THE VIDEOS ONTO YOUR PI

It's time to transfer the videos over! First, move all your videos to your USB thumb drive in a folder named */encoded*.

On the Pi, let's change directories:

```
cd ~/simpsonstv/videos
```

This is the directory where the video player will look for videos to play.

Plug your USB drive into the spare USB port on the Pi (Figure F); you'll need a Micro USB adapter for this. Once the USB drive is connected, copy all the videos from it to the Pi. Run the following command to copy the videos onto your Pi:

```
sudo cp -R /media/usb/encoded/. ~/simpsonstv/videos
```

This will probably take some time. Take a break, have a glass of water, go ponder the clouds.

OK now, final step! We want our video player and button control scripts to run every time the Pi boots. We will create a ***service*** for both scripts and set the Pi to run them whenever it boots. First, set up the button service:

```
sudo touch /etc/systemd/system/tvbutton.service
sudo nano /etc/systemd/system/tvbutton.service
```

Now paste the following into the editor:

```
[Unit]
Description=tvbutton
After=network.target

[Service]
WorkingDirectory=/home/pi/simpsonstv/
ExecStart=/usr/bin/python /home/pi/
simpsonstv/buttons.py
Restart=always

[Install]
WantedBy=multi-user.target
```

Next, set up the player service:

```
sudo touch /etc/systemd/system/
tvplayer.service
sudo nano /etc/systemd/system/tvplayer.
service
```

Paste the following into the editor and save:

```
[Unit]
Description=tvplayer
After=network.target

[Service]
WorkingDirectory=/home/pi/simpsonstv/
ExecStart=/usr/bin/python /home/pi/
simpsonstv/player.py
Restart=always

[Install]
WantedBy=multi-user.target
```

Now set these two services to start on boot:

```
sudo systemctl enable tvbutton.service
sudo systemctl enable tvplayer.service
```

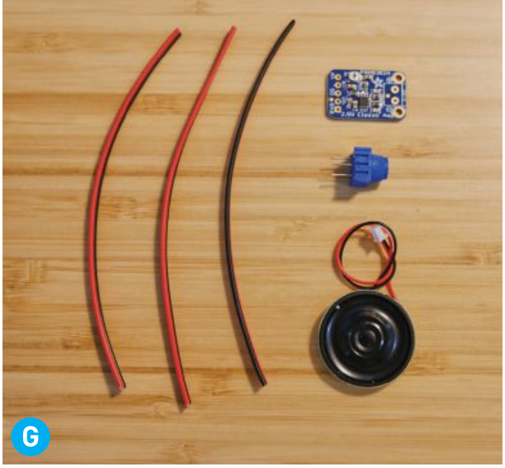
Now shut down the Pi:

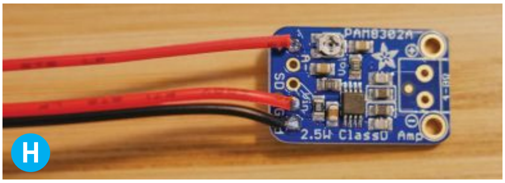
```
sudo shutdown -h now
```

Then unplug it from the power source. Phew, all this hacking has made me thirsty. Let's order a Tab. Take a break, and then let's finish up the hardware side.

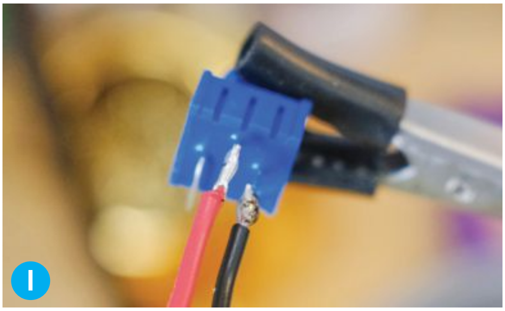
6. SOLDER THE AUDIO CIRCUIT

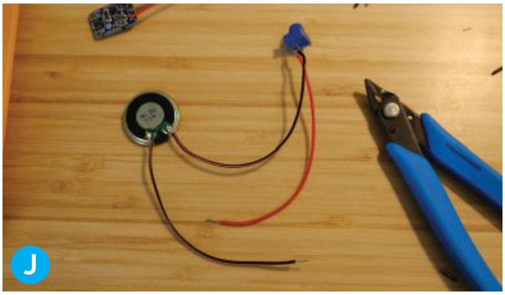
Grab your trusty soldering iron and the rest of your components; you've got a little bit of

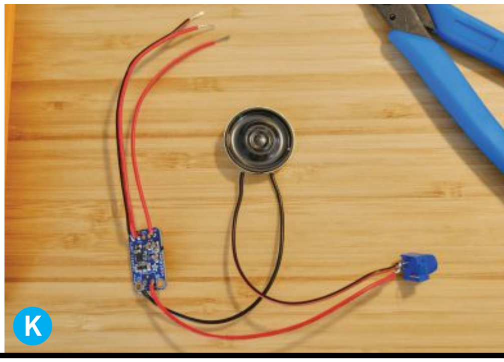

G



H


I


J


K

soldering left to do. It might be helpful to set this all up on a breadboard first, but if you are feeling embiggened by the noble spirit, go ahead and solder it up. I cut nearly all the wires to 4"–5" lengths. This makes final assembly a bit easier.

Let's start with the audio circuit. You'll need 3 wire pairs, the 1K potentiometer, the amplifier board, and the 4-ohm speaker (Figure G).

To start, let's solder up the outgoing wires on the amplifier. Take one black/red pair and solder the black to the ground and the red to the Vin pad next to the ground. These are the wires that supply the amplifier with power. Mark them if you'd like, or take a mental snapshot. Next, solder a single wire to the Audio In+ pad (also marked A+) which should be the outermost pad (Figure H).This is the wire that supplies the audio signal to the amp. We will connect these to the Pi later.

Now solder the speaker and potentiometer. A potentiometer, or *pot*, is a variable resistor with three legs. As the knob is turned, the resistance applied to the output is increased or decreased, which results in a lower or higher current. Essentially, it makes the signal weaker or stronger. This will control the volume going into the speaker. Solder a wire to the input (the center pin) and one to the output (the rightmost pin as shown in Figure I).

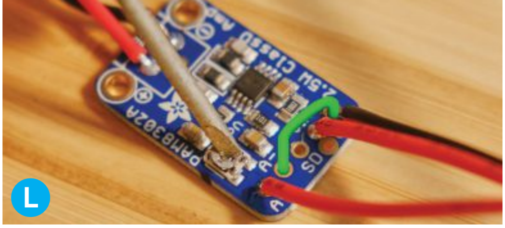
Solder a wire to one of the two speaker leads (doesn't matter which one). Next solder one of the wires coming from the pot to the other speaker lead (Figure J).

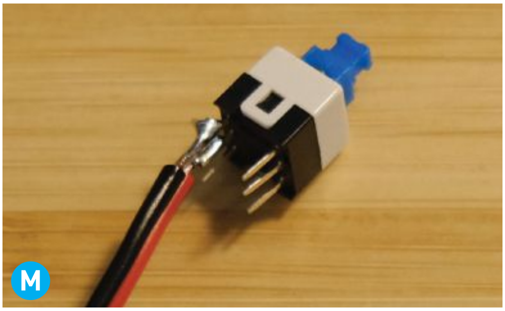
Connect the speaker circuit to the amplifier. There are two output pads on the opposite side of the amp from the input. Solder the loose speaker wire to one, and the loose wire from the pot to the other (Figure K).

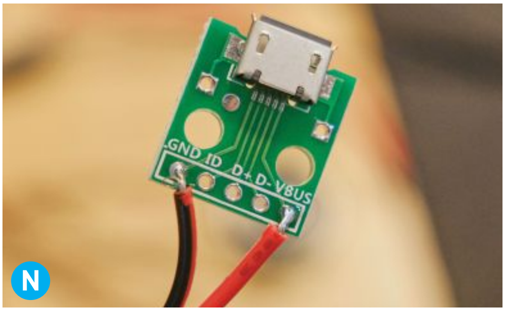
Finally, solder a short jumper wire between the A– and GND pins of the amp (Figure L), and make sure the amplifier is turned all the way up. On the board you'll see a small silver knob with a Phillips screwdriver marking. Use a small screwdriver to turn the knob up (clockwise). Your speaker circuit is ready!

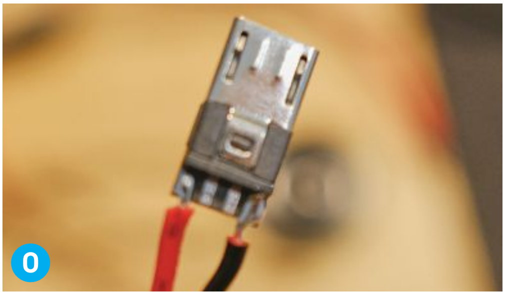
7. SOLDER THE POWER CIRCUIT

The power button is super easy to hook up. Solder a wire to the center pin, and another wire to the outermost right pin (Figure M).



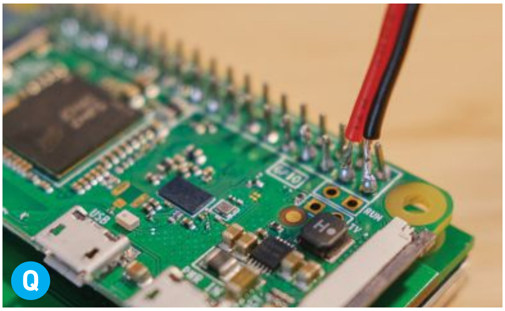


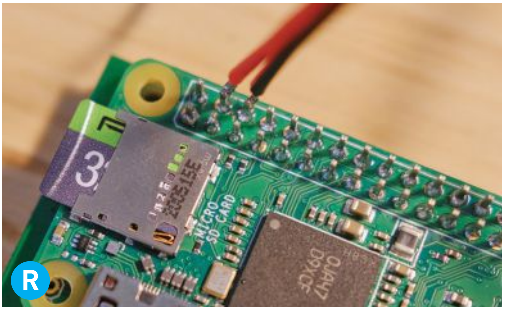


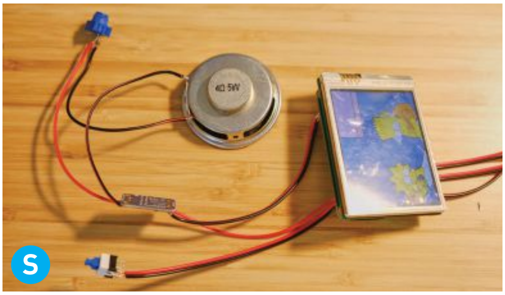


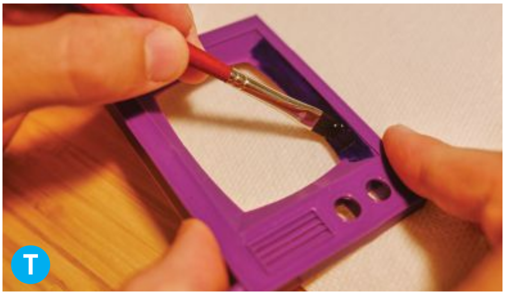
The power cable is essentially a USB adapter cable. We need our Micro USB breakout board, a Micro USB header, and a short length of wire. Solder a red wire to the positive, and a black wire to the ground of the breakout board (Figure N).

Now solder the other end of the two wires — being careful to match the colors — to the USB header (Figures O and P).









8. CONNECT IT ALL TO THE PI

With the SD card facing north, solder the two wires of the power button to the Raspberry Pi Zero's GPIO pin 26 and the Ground pin that's below it (Figure Q). Check the pinout for your Pi online at pinout.xyz.

Next, connect the audio circuit. Take the two power wires that you marked earlier and solder the red to the 5V+ pin and the black to the Ground pin in the upper right hand, next to GPIO 14 (Figure R).

Take the remaining wire and solder it to GPIO 19, which is next to the power button. OK, all the soldering should be done! Holster that bad boy, but do unplug it and wait for it to cool down first.

Let's make sure everything works. Plug in your Simpsons TV and wait for it to boot. The screen should stay off unless the screen power button is engaged. Eventually some videos should start playing (Figure S)! Test that the power button toggles on and off and that the audio knob raises and lowers the volume. Phew! All the scary stuff is over. Time for the final stretch!

9. MOUNT COMPONENTS TO TV FRONT

Now let's put the housing together. If you want to go the extra mile (why would you come this far to not go a little further?), you can take some dark purple acrylic paint and paint a few details onto the case. Paint the legs and the inlay around the front of the screen (Figure T). It should be quick work; the case was designed for easy painting. If you mess up, use a wet paper towel to wipe off the paint. You'll probably want to do two coats.

Take the power knob, the one with the square on the back, and push it down onto the power button (Figure U). There should be a bit of resistance and a satisfying snap.

Next take the volume knob and apply a bit of super glue to the back. Press the front of the pot into the knob. Hold it. Hold it. Hold it. There we go (Figure V).

Now you'll mount the components into the front of the TV. Be sure to remove the anti-scratch sticker from the front of the screen. Mmmmmhmmm, pleasing. Also remove the power extension cable from the Pi. Place the screen facedown onto the front housing. Make sure the screen is oriented properly, with the Pi on the south side (Figure W).

Plug in your trusty hot glue gun. First dab the top left corner of the screen (Figure X). Cover it with glue from the top edge down to the housing. Next run a bead of hot glue along the top and bottom edges of the screen (Figure Y). Be careful to not entirely fill the void in the case at the top or bottom where the casing snaps go.

Now place the power button in the top knob housing. First make sure it's disengaged: press the button and release until it's at its tallest position. Gently press it into the housing until the knob comes out the front (Figure Z). Be careful to not press the button itself in, or it will never be able to disengage. Use some hot glue to secure the button housing in place. Hold in place until the hot glue sets.

Follow the same procedure for the volume knob (Figure Aa). Don't press it in too firmly, or the knob won't turn.

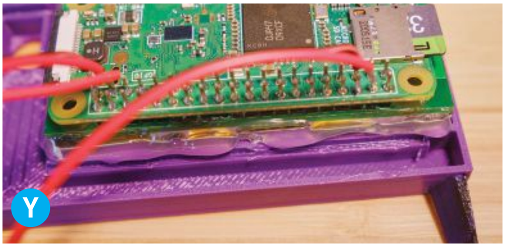
10. MOUNT COMPONENTS IN TV HOUSING

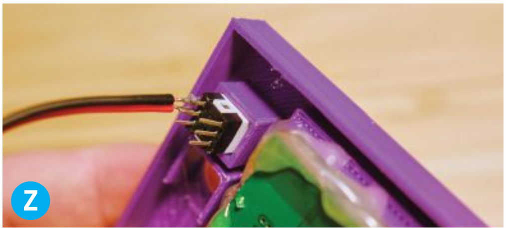
OK, home stretch! Grab your USB power extension cable. Place the Micro USB board on its seat in the bottom left corner of the large TV housing (Figure Bb). I find this next step is easier if you plug a Micro USB cable into it after placing it (leave everything unplugged from any power source). The cable helps to hold it in place.

Use a bit of hot glue to tack it into place. Then, generously cover the whole thing, connecting it to the housing. Wait for the glue to cool and turn white (Figure Cc) before moving on.

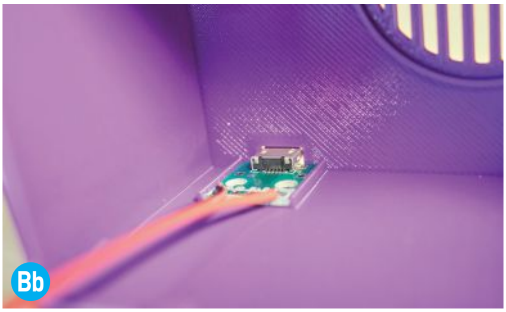
Place the speaker facedown on the back ring.

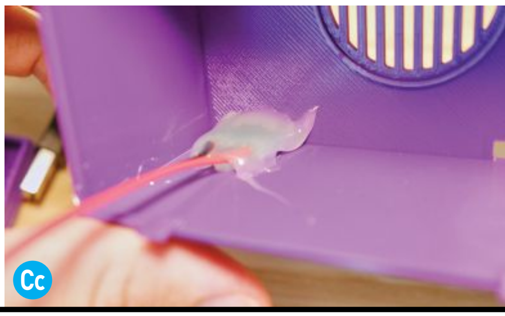

X


Y

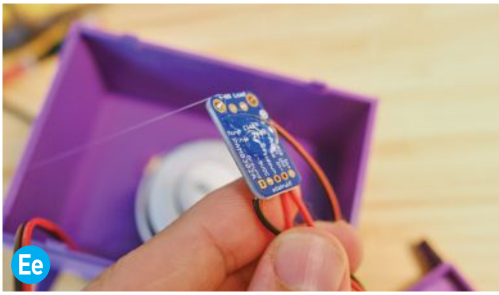

Z

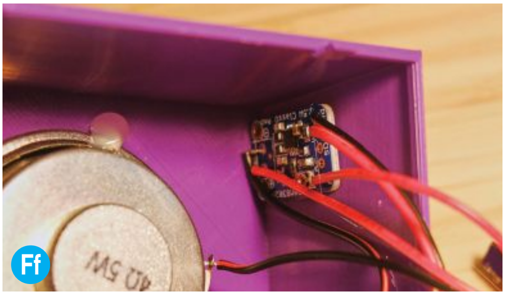

Aa

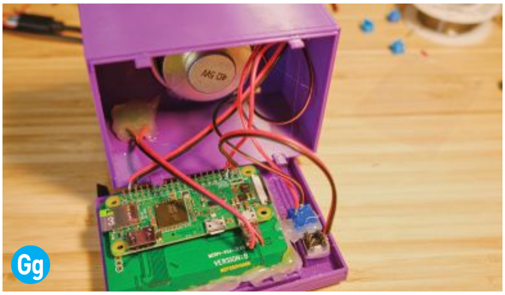

Bb

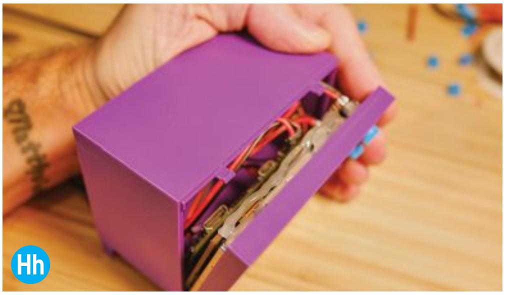

Cc


Dd


Ee


Ff


Gg


Hh


Ii


Jj


Kk

Use hot glue to secure it around the edges. You don't need to glue the whole thing, just a few points. Again, wait for the glue to set and check that it's secure before continuing (Figure Dd).

Finally, the amplifier. Put just a dab of glue on the underside (Figure Ee) and press it into the side wall of the enclosure (Figure Ff). Easy peasy. OK! Great work. You're done with the hot glue gun, so holster it next to your soldering iron. Check everything and make sure it's secure. Remove any hot glue spider webs that you find.

One more thing before we close it up: connect the power extension cable into the Pi's USB power port (Figure Gg).

Time to close it up! The front of the TV snaps onto the housing using four case snaps. Align the two pieces, and carefully fold any wires into the enclosure. Starting at the bottom, press and snap the front onto the case (Figure Hh). Nice!

Time to assemble the VCR. Cut two short lengths of black printer filament for the antenna. Using a bit of super glue, press them into the two antenna holes on top of the VCR (Figure Ii). Then glue on the two antenna toppers (Figure Jj).

Put a bit of super glue on the bottom and place the VCR on top of the TV (Figure Kk). Don't worry about being too precise; do you think Homer had his VCR perfectly centered?

Et voilà! You have a tiny TV.

Congrats on making it this far — you are awesome! Tweet me a pic of your finished TV @theWithra; I'd love to see it.

IT'S KRUSTY TIME!

When the Pi is powered up through the USB port on the back of the TV, it starts playing episodes at random. When one episode ends, the next is randomly selected. The top button turns the screen on and off, while also muting the volume. The bottom knob turns the volume up and down.

After this project went viral on Reddit and Twitter this summer, I decided to make 10 TVs and raffle them off — with 100% of the proceeds going to four awesome charities: Living Lands and Waters, To Write Love on Her Arms, American Modeling Teachers Association, and the Highlands Museum & Discovery Center in my hometown of Ashland, Kentucky.

LEVEL UP

But wait! There's more. Here are a few ideas for upgrading your Simpsons TV.

- **Faster boot time** — Yeah, I hate waiting for my Simpsons TV to boot too. The first version was built on PipaOS (pipaos.mitako.eu), a super lightweight version of Raspbian Jessie that booted almost instantly. It was fabulous. Unfortunately, as I was writing this guide, Jessie was discontinued and is no longer supported. There is a newer version of PipaOS built on Stretch, but I haven't tried it yet. You should be able to follow this guide with PipaOS, but I haven't tested it. You might run into some errors.
- **Gapless playback** — This is a fun one! I've managed to get near gapless playback working with OMXPlayer by running two instances of OMXPlayer and controlling them using DBUS. This is a little advanced and takes some finessing. The other catch is that you need to know the runtime of each video, which can be found with Fprobe. This adds another layer of complexity to the whole thing.
- **TV on-off graphics** — I was able to add TV graphics such as interstitials, on-off blinks, and even a little logo in the bottom corner using Kivy. Kivy is a free, lightweight graphics package for Python. It must be compiled for your OS of choice, which is a more advanced topic. I find that something like Kivy is best for the Pi Zero, since it is not powerful enough to play video and also render out HTML. If you decide to go down this rabbit hole there are a lot of interesting quirks to get OMXPlayer and Kivy to play together. I'd be happy to write a guide sometime if there's enough interest.

BE COOL

Please, please do not build these and sell them, and definitely do not sell them with copyrighted material on them!

Check for the latest version of this guide at withrow.io/simpsons-tv-build-guide. And If you find this guide helpful and want to buy me a coffee, you can donate at makezine.com/go/withra-paypal.

DIY Film Wigglegram Lens

Salvage 3 disposable cameras to build this 3D printed, 3D lens

Written and photographed by Joshua Bird

TIME REQUIRED: 1–2 Hours

DIFFICULTY: Easy

COST: $20–$30

MATERIALS

- **Disposable cameras with 30mm lens (3)** Fujifilm QuickSnap Flash 400
- **3D printed parts: body back and front, FD mount bottom and top, and viewfinder attachment** Get the 3D files free from github.com/jyjblrd/wigglegramLens.
- **Self-tapping screws, M1.4×3mm (7)** such as Shuntian KB1.4X3 or Amazon B07GKXMSVR

TOOLS

- **3D printer**
- **Small screwdriver, Phillips head**

FLIP BOOK example of a distinct foreground, midground, and background. Flip through this article for the effect.

JOSHUA BIRD is an avid maker and amateur photographer currently studying computer science at the University of Cambridge. Check out his photography and programming projects at joshuabird.com.

I gotta admit that half of the reason I bought a film camera was to post cool-looking pictures on Instagram, so when I saw these things called wigglegrams on IG, I immediately wanted to make my own. Wigglegrams are very short animated 3D-ish videos created by looping together three or four images taken at the same time. I found out that they are typically made with a Nishika camera which has four lenses to capture four images at once. Originally these Nishika images were printed on lenticular paper to create the 3D viewing effect, but nowadays people animate them into wigglegram GIFs.

Unfortunately, these cameras have skyrocketed in price over the last few years (due to film normies like me) and are also notorious for their terrible build quality. I wasn't ready to spend a couple hundred dollars just to make my IG feed look a bit less boring, so instead I designed my own Canon FD-mount wigglegram lens to use with my Canon A-1.

I was inspired by a similar creation by George Moua, for Sony E and Fuji X mounts (georgemoua.com/3d-wiggle-lens), however his version is for digital cameras and he's selling them for $70–$80. Mine is open source; you can get the 3D files for free on my Github repo at github.com/jyjblrd/wigglegramLens, and the STEP files are included so you can adapt it for any camera.

SQUEEZING 3 PHOTOS INTO 1 FRAME

The Nishika camera that I was trying to emulate took four photos across two frames of film (thus creating four half-frame images). However, since I wanted to use this lens on a normal, unmodified film camera, I instead decided to squeeze three images onto one frame. This unfortunately creates a pretty thin wigglegram, so framing is a bit hard — especially since there is no viewfinder (more on this later). It also means that the lenses are closer together — the distance from the rightmost to leftmost lens is 24mm on my wigglegram camera compared to the 54mm of the Nishika — so to get a good wigglegram, you have to get pretty close to the subject (Figure A on the following page).

CAMERA LENSES

The only parts of this lens that aren't 3D printed are the three plastic lenses and a few screws. The lenses came from expired disposable cameras I got for dirt cheap on the Hong Kong version of eBay. After passing out the disposable cameras at a party, I was left with 3 rolls of pretty terrible

Here's a wigglegram straight from the lab. You can see the three separate images on one frame of the negative.

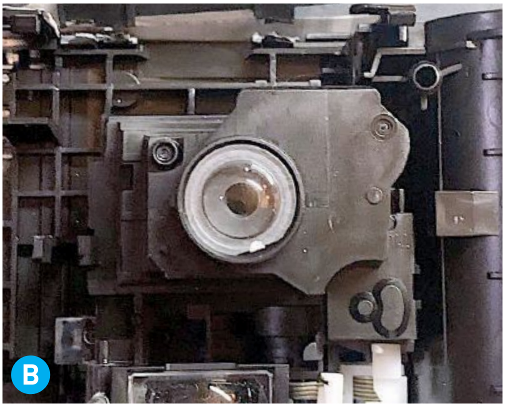
photos and three plastic 30mm lenses. It's pretty easy to get the lenses out of the camera. Just rip it apart until you reach the lens (Figure B).

NOTE: While you're at it, salvage the flash module and build a small camera flash (joshuabird.com/blog/post/making-a-compact-camera-flash)!

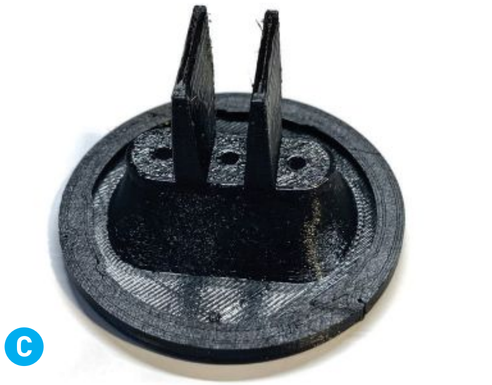
THE DESIGN

I modeled up the lens in Fusion 360. It's a pretty simple design, however it was a bit of a pain to figure out how far away the lens should be from the film plane to get the image in focus. I wanted subjects from about 0.2m–1.5m to be in focus since I would be primarily shooting things close to the lens. At first, I tried to calculate the distance to use, but eventually I just decided to make a ton of designs to slowly hone in on the right distance. The lens has a relatively small aperture of f/10, so it doesn't have to be super precise.

The final distance from the lens to the film plane was somewhere around 30mm (obviously, since the lens has a focal distance of 30mm), which meant that the lenses are recessed into the camera body by about 10mm.



Disassembled disposable camera with its lens.



You can see the two dividers which will extend into the camera's body.

ASSEMBLING YOUR 3D LENS

Printing the parts takes about 30 minutes, depending on your 3D printer. Set infill to 100% and make sure to use black filament to prevent light leakage.

To assemble the wigglegram lens, simply place the three lenses into their recesses and screw the part together using 3mm M1.4 self-tapping screws. The lenses should be held snugly in place once the screws are tightened.

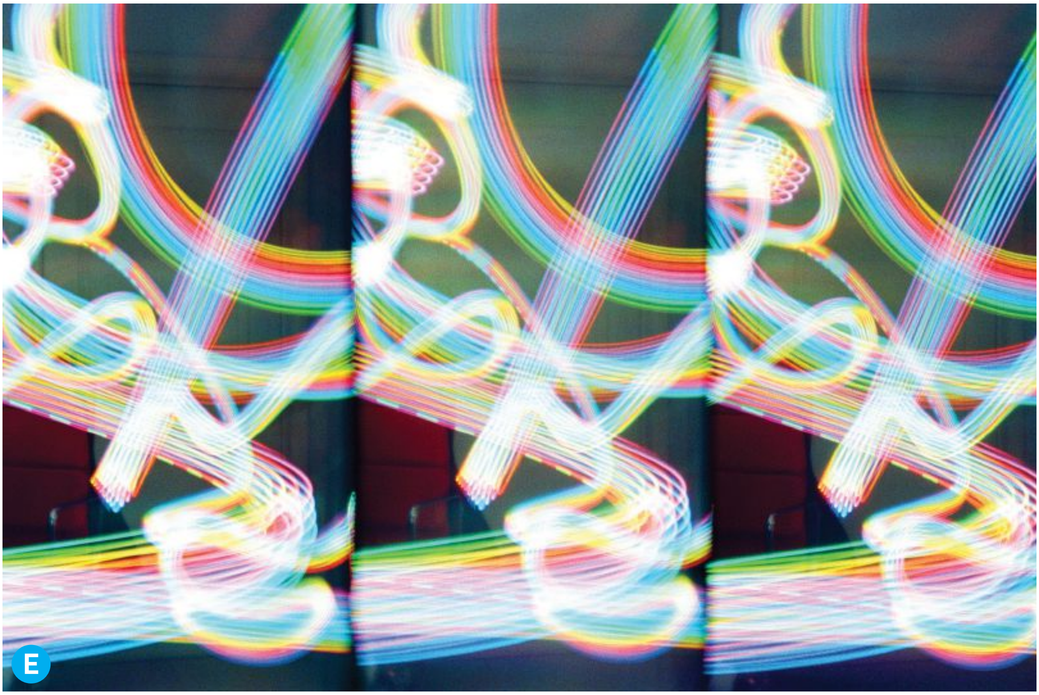
NUDGE DON'T SMUDGE

To keep the three images separated, there has to be a divider between each lens that extends into the camera almost all the way to the film plane (Figure C). These dividers stop about 5mm before the film plane, so they don't hit the shutter mechanism — still close enough to give three distinct images on the film.

This was a bit of a problem for my Canon A-1 since it is an SLR (single-lens reflex) camera, meaning that the mirror for the viewfinder is in the way. Some SLR cameras have a "mirror lockup" feature which allows you to flip this mirror up and out of the way (normally used to prevent camera shake when taking a photo), however my camera doesn't have this. I instead just kinda push the mirror up when putting on the lens (Figure D), being careful not to smudge the mirror in the process. Unfortunately this also means you can't use the viewfinder to preview the image, but that's a luxury I can live without.

D

Pushing the viewfinder mirror out of the way.



A long-exposure light painting of the rainbow LED cube I made — go check out my blog post (joshuabird.com/blog/post/rgb-led-cube) about it!

The Boomerang effect — a single frame of film, when divided into three, actually captures a few milliseconds of the subject's own motion. You really have to look at this one on-screen to see how cool it is! joshuabird.com/blog/post/3d-printed-wigglegram-lens

STITCHING THE IMAGES TOGETHER

A common complaint of wigglegrams is that it's a pain to animate the frames together in Photoshop, but thankfully there's another method. Recently a company called Reto Project released the Reto 3D, a wigglegram camera that takes three half-frame images. Alongside this product, they also published an app (retoproject.com/app) which allows you to easily create wigglegrams. It costs a few dollars but it's definitely worth it.

SOME EXAMPLES

I've put a couple rolls of film through this lens and I've been really impressed with the results (Figure E). It turned out far better than I expected, and the plastic disposable camera lenses are surprisingly sharp. A few of the photos didn't turn out, mainly due to bad framing (there has to be a very well-defined foreground, midground, and background).

FILM BOOMERANG?

OK, this is a completely unexpected (but very welcomed) feature. The shutter of my camera moves from right to left, meaning that the rightmost frame is exposed slightly before the leftmost frame (probably by only 10ms or so). This means that there is a slight "Instagram Boomerang" effect — a tiny bit of the subject's own motion is captured too. This really is the perfect lens for IG! This Boomerang effect only happens when not using flash, however, since the camera flash exposes the whole frame at the same time (Figure F).

WIGGLE ON

I am extremely impressed with how well this project worked out — even if I was given the option, I would still choose to use this lens over a Nishika! I hope you'll build it and share your wigglegrams on IG @joshua.bird.

See all these wigglegrams in motion at joshuabird.com/blog/post/3d-printed-wigglegram-lens.

CROSS-EYED STEREOSCOPY

To simulate the 3D effect of the wigglegram animation on the printed page, try this old trick for viewing stereoscopic photos:

1. Look at the two side-by-side images.

2. Cross your eyes slightly until you see a third image between them. If you see four images then cross your eyes slightly less, until there are just three. Find that sweet spot where the middle image is lined up, and try to relax and hold it.

3. Now focus on that middle image until it pops out in 3D!

For me, it helped to sway my head back and forth just slightly; this seemed to force both my eyes to track the middle image together, and then — boing! — the bottle in the foreground was clearly waving back and forth out in front of the woman's face in the midground, and the woman was clearly separated from the wall behind her. Cool!

—Keith Hammond

FOUR EYES CAMERA

A DIGITAL TWIST ON A VINTAGE CLASSIC By Brandon Withrow

The digital Four Eyes Camera.

Brandon Withrow, John Alan Elson – 3dham.com

I've always been obsessed with camera projects. I love making cameras and photography-related hardware. One of my favorite vintage cameras is the Nimslo. The Nimslo featured four individual lenses and shutters; all four would fire at the same instant. The resulting developed film images could be printed out into special lenticular paper that would filter the images as you shifted it from side to side, giving a fun three-dimensional effect.

I had a few cheap cameras lying around after building a 360 virtual reality camera, so I decided to use them to create a new, all digital, Nimslo camera (withrow.io/four-eyes-camera), shown above. Using the circuitry I designed for the 360 camera, I linked the shutters of the four cameras together. I built a small cardboard enclosure with a single power button and a single shutter button that would cause all four cameras to capture a photo at exactly the same moment. But I needed a way to combine the photos.

I wrote an iPhone app that would connect to the cameras via an ad-hoc Wi-Fi network and download the photos. Next, the app would combine four photos into a looping GIF movie that would give them a 3D effect (Figure E).

After some time playing with the camera, I decided to ditch the hardware and just make a fully featured iPhone app! You can find it at foureyesapp.com.

The original Nimslo 3D Quadra Lens camera A, released in 1982 by inventors Dr. Jerry Curtis Nims and Allen Kwok Wah Lo; its film chamber B; a 4-image negative C; and the resulting positive D for printing on lenticular 3D viewing paper. After 1986 cheaper versions were made by Nishika. Today collectors use these to create 3D GIFs. Learn more at outsidetheshot.com/nimslo-3d-camera.

Some "fouries" taken with my digital Four Eyes camera.

Hot Keebs

Build duckyPad — a light-up USB macro keypad to speed up work, gaming, streaming, everything!

Written and photographed by Allen Wong

ALLEN WONG is an embedded systems, retro computing, and Chatot Pokèmon enthusiast in Nottingham, England.

TIME REQUIRED: 1–4 Hours

DIFFICULTY: Easy–Intermediate

COST: $60–$80

MATERIALS

» **duckyPad Macro Keyboard Kit** various options at tindie.com/products/21984, includes:
- **duckyPad circuit board** It's got 15 Kailh hot-swap key sockets, 15 SK6812 RGB LEDs, microSD and USB-C sockets, four tactile switches, a powerful STM32F072C8T6 microcontroller, and a 1.3" OLED display.
- **microSD Card**
- **Cherry MX-style key switches (15)**
- **Keycaps (15)**
- **Front and back plates**
- **Silicone anti-slip pads (4)**
- **Machine screws, size M2: 8mm (6) and 2.8mm (8)**
- **Brass standoffs, size M2: 2mm (7) and 4mm (6)**

— OR —

» **duckyPad DIY Version** duckyPad is fully open source. To build your own PCB and keyboard from scratch, get the complete bill of materials at github.com/dekuNukem/duckyPad/blob/master/build_it_yourself.md, along with PCB files, dimensions drawings and vector files for the housing, and advice on sourcing parts.

TOOLS

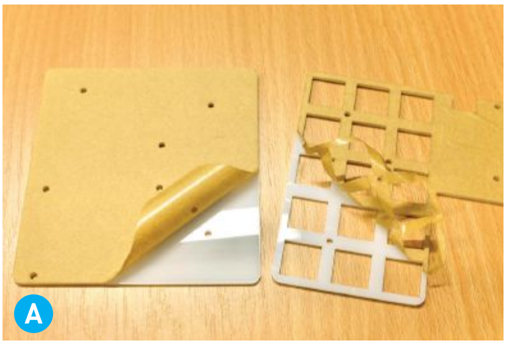
» **Computer with internet connection** You can download all the software and additional instructions mentioned in this tutorial from github.com/dekuNukem/duckyPad.

» **Small screwdriver, Phillips head** included in kit

» **Key puller (optional)** included in kit

» **Small pliers (optional)** helpful for tightening fasteners

» **Solder reflow oven (optional)** if you're making your own PCBs


A

DuckyPad is a 15-key, do-it-all mechanical macro pad that helps streamline and speed up your workflow by automating keyboard (and mouse!) inputs. It features a sleek design and all the goodies of a high-end keyboard — mechanical switches, hot-swap sockets, RGB LEDs for each key, and USB-C.

But more crucially, duckyPad also features previously-unseen-in-macro-pads innovations such as:

- OLED screen that maps what each key does
- Sophisticated multi-line scripting with duckyScript
- 32 profiles of 15 keys, for 480 macros total
- Automatic profile switching based on active window
- microSD card storage — move between devices without losing macros
- Works with all major OSs, no driver needed.

When pressed, each key executes a user-created duckyScript to automate keyboard actions and speed up your everyday routine. The scripts can be as simple as shortcuts like Control+C, or as sophisticated as launching applications, managing livestreams, moving the mouse cursor, even creating root backdoors or grabbing passwords. 😈 It's all up to you!

Best of all, duckyPad is fully open source and *respects your privacy.* No need to create an account, no data collection whatsoever, no internet connection required. You don't even have to use the companion app; you can set up your duckyPad manually instead.

You can assemble your duckyPad from a kit in a half-hour, no soldering required, or build your own from scratch if you're handy with surface-mount soldering. Here's how to build it and use it.

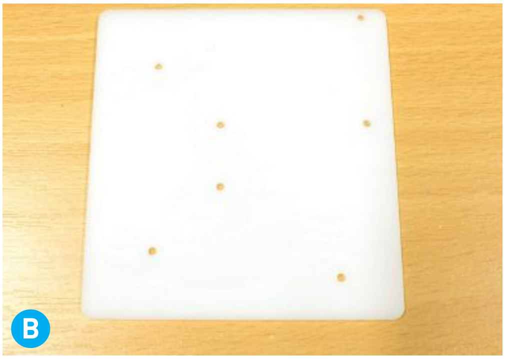
BUILD YOUR DUCKYPAD

1. PREPARE THE BACK PLATE

Start by peeling off the protective cover on the front and back plate (Figure A).

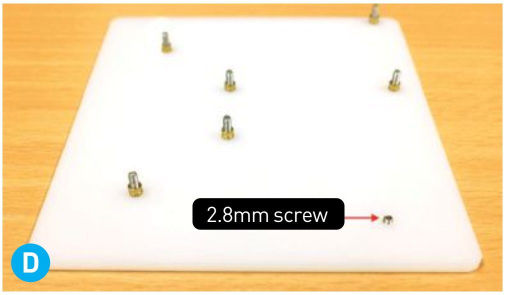
Place the Back Plate in front of you as shown in Figure B on the following page. Note the orientation of the holes.

Starting from a hole in the middle, put an 8mm screw through the backside. Then screw on the

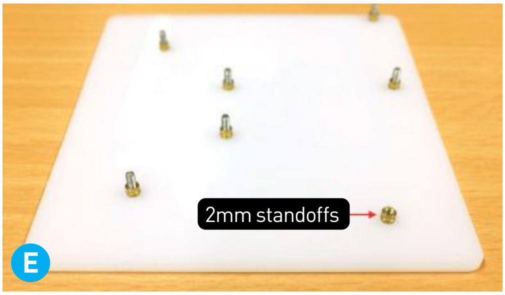


B

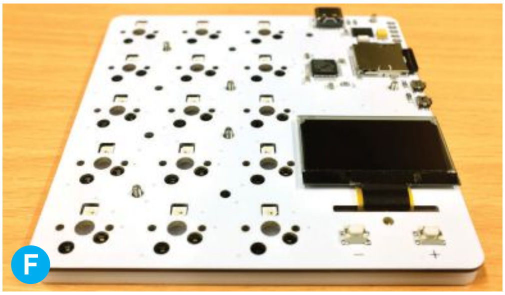
C



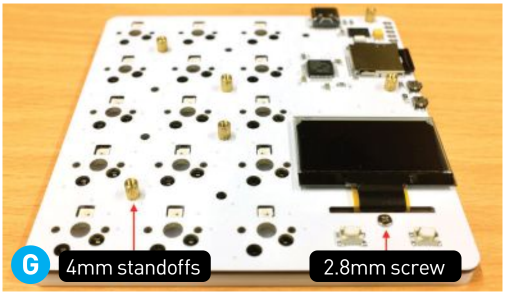
D



E



F



G

2mm standoff. Don't overtighten! Doing so might deform or crack the acrylic plate. Finger-tight is OK, just make sure it's not loose and doesn't rattle. Do the same with all the holes except the bottom right (Figure C).

Put a 2.8mm screw through that last hole (Figure D) and install the 2mm standoff (Figure E). This is gonna be a PITA, but you'll eventually get it. Careful not to drop the small parts.

2. MOUNT THE PCB

Now go wash your hands, grab the circuit board by its edge, line it up and install it as shown in Figure F.

Install all six 4mm standoffs, and moderately tighten. Also install the 2.8mm screw on the lower right (Figure G).

HANDLING BARE CIRCUIT BOARDS

When handling the bare circuit board, remember to:

- **Wash your hands first**
- **Always hold the circuit board by its edge**
- **Don't touch the components.**

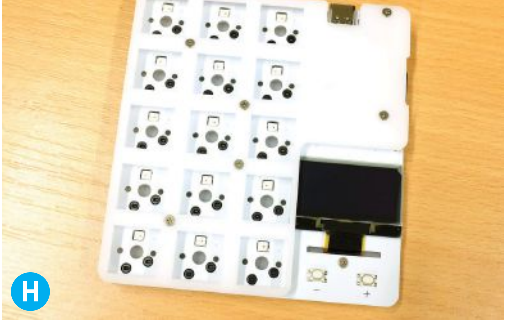
The reason for hand-washing is to reduce the risk of static electricity (SE) on your body zapping the board and damage it, especially in dry weather. By washing your hands, you release the existing SE through the metal water pipe, your hands are moist to prevent further SE buildup, and your clean hands won't leave greasy fingerprints all over the place, win-win-win!

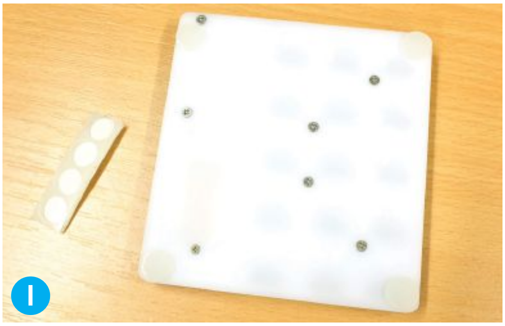
Of course, your hands can't be dripping wet either, but that's just common sense.

3. MOUNT FRONT PLATE AND FEET

Install the front plate using the remaining 2.8mm screws (Figure H). Do not tighten each screw as you install it! Put them all on loosely first, then tighten them one by one. This ensures the plate is properly lined up. Again, don't overtighten.

Flip it over, clean the back plate with a dry cloth if dirty, and stick on the anti-slip pads on each corner (Figure I).



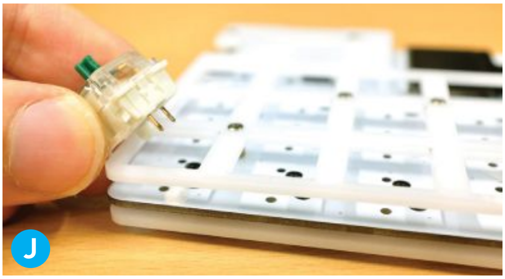


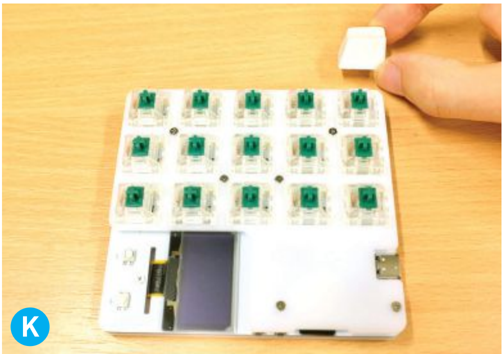
4. INSTALL THE KEYS

Time to install the key switches! duckyPad supports hot-swap, so no soldering is needed. Make sure the pins are straight, otherwise you might damage the circuit board or the switch itself! Then line up with the holes, and gently push the switch straight down until it clicks in place (Figure J).

Keycaps next! Note the keycap orientation ("slope" side pointing left in Figure K). Just push it down. Each key should line up evenly. If not, try to push it down or wiggle it a little.

Finally, peel off the protective cover on the OLED screen, and you're done (Figure L). Congratulations!



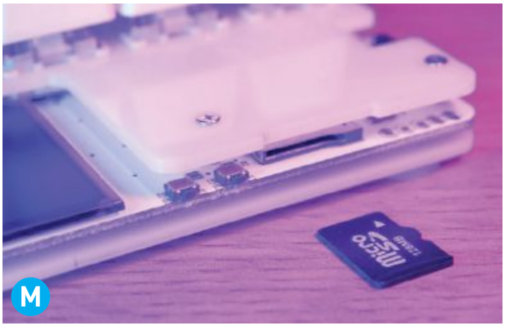


USING YOUR DUCKYPAD

Your duckyPad kit comes with a microSD card already installed, containing some demo profiles. Simply plug it into your computer, and voilà!

- Press a key to execute the corresponding script.
- Press + and – button to switch profiles.
- Hold down + button for profile quick-switch.
- Hold down – button to change RGB backlight brightness.
- To change keyboard layout, hold down the top left key while plugging it in, then select your layout.

The samples might be fun, but duckyPad's true purpose is to do what you want! duckyPad uses a microSD card to store your profiles, scripts, and settings. You can insert/remove the SD card while duckyPad is on (Figure M). If you decide to use your own SD card, it should be formatted in FAT32 or FAT.



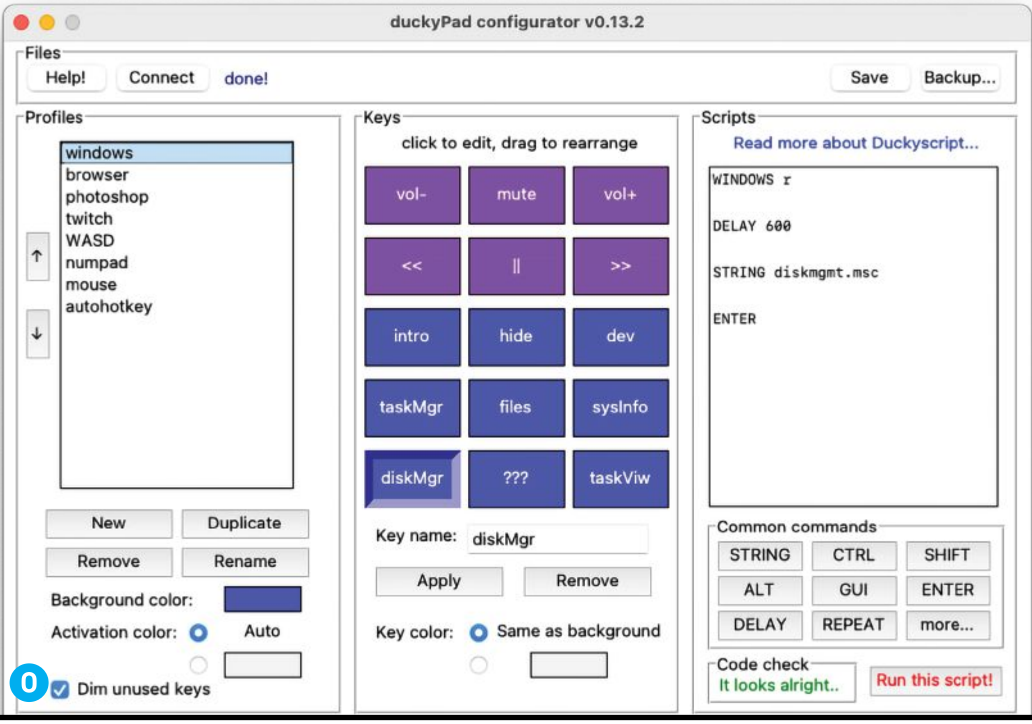
USING DUCKYPAD CONFIGURATOR

Head to github.com/dekuNukem/duckyPad/releases to download the latest release of duckyPad Configurator for Windows, MacOS, or Linux. Extract the *.zip* file and launch the application by clicking *duckypad_config.exe* (Figure N). For Linux users, simply run the *duckypad_config.py* script with Python 3.6 or higher.

NOTE: Your system might complain this software is "untrusted." This is because I haven't had the code digitally signed, which costs hundreds of dollars a year. Feel free to review the code file *pc_software* at the Github repo. If you really don't trust the app, you can run the *duckypad_config.py* script itself with Python 3, or configure your duckyPad manually instead (*manual_setup.md*).

After launching the app, make sure your duckyPad is plugged in, and press the Connect button. The app should then load all the data from duckyPad (Figure O).

You can also load from SD card. Remove the card from duckyPad, mount it on your computer, and select the entire SD card. It should load just the same (Figure P).



- **Profiles** Let's break down the GUI (Figure Q). The leftmost column is for profiles. Each profile contains a group of scripts corresponding to the keys on the duckyPad. As you can see, you typically create a profile for each app you want to control.

  Use the buttons to create a new profile, duplicate an existing profile, and remove or rename a profile. Use the up/down arrow buttons to reorder your profiles.

  Choose a background color for the RGB backlight. "Activation color" is the color that a key changes into when you press it. On the Auto setting it would be the inverse of background color. By default, unused keys are dimmed; you can override this by unticking the checkbox.

- **Keys** In the middle column, you can configure the 15 keys on your duckyPad. Click a key to select it, type to give it a name, or click Remove to delete it. You can assign a unique color to each key, or use the same color as background. Drag a key to rearrange its order.

- **Scripts** Finally, the rightmost column is for scripts. When you select a key, the script it will execute is displayed here.

WRITING DUCKYSCRIPT

duckyPad uses duckyScript, a simple scripting language for automating key presses. It was originally developed for USB Rubber Ducky, the famous "bad USB" thumb drive that hackers and pentesters use to "inject keystrokes at superhuman speeds" (shop.hak5.org/products/usb-rubber-ducky-deluxe).

duckyScript is very easy and straightforward to write, you basically just tell it what key to press! Let's take a look at some examples first:

Open the Task Manager:

```
CONTROL SHIFT ESC
```

Open a webpage on Windows:

```
WINDOWS r
DELAY 400
STRING https://www.youtube.com/watch?v=
dQw4w9WgXcQ
ENTER
```

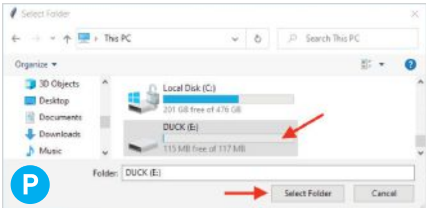
Save a webpage then close it:

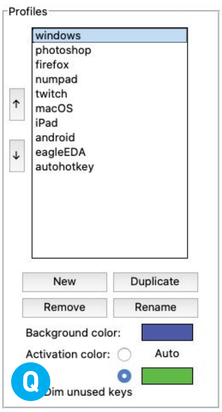
```
CONTROL s
DELAY 600
ENTER
DELAY 600
CONTROL w
```

Use **REM** for comments, **DELAY** to pause the script (useful for waiting for the UI to catch up), and **STRING** for typing out text. Code check will be performed as you type, and errors will be highlighted in yellow.

DuckyScript supports all special keys and function keys (**ESC**, **CTRL**, **RCTRL**, **COMMAND**, **CAPSLOCK**, **BACKSPACE**, **F1–F24**, etc.), plus media keys (volume up/down, play/pause, etc.), numpad keys, mouse buttons, scroll wheel, mouse moves (x-y coordinates), and more! It also supports dozens of keyboard layouts, from the default English-US layout to Dvorak, French, Belgian, and more. Dig into all the details at github.com/dekuNukem/duckyPad/blob/master/duckyscript_info.md.

Press the "Run this script" button to test-run it on your computer. *Make sure you trust the script!* Test-run starts after a 2-second delay. It can't get past Windows User Account Control screens, so you'll have to run it on the real thing. On macOS, if





test-run doesn't work, go to System Preferences → Security & Privacy → Accessibility, unlock, remove then add the app.

Test-run might not be 100% accurate; some keys like **CAPSLOCK** and **NUMLOCK** can't be emulated in software. Run your scripts on the real thing if you're not sure.

Finally, press the Save button to save the changes.

RUN IT!

duckyPad should automatically reboot when you press Save, and your new scripts will be ready to use! (If you're configuring via SD card, insert it back in duckyPad and then power it on.) Again, press a key to execute the corresponding script, and press the + and – buttons to switch profiles.

duckyPad can also switch profiles automatically based on your current active window, say, from Windows (Figure R) to Twitch (Figure S) to web browser (Figure T) to code editor (Figure U)! Check out the Profile Auto-Switcher app at github.com/dekuNukem/duckyPad-profile-autoswitcher.

Here are just a few examples of what you can do with your duckyPad:

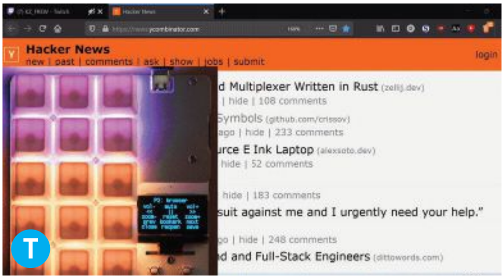
- **Photoshop / CAD** — Put all your commonly used shortcuts in one place, for a more efficient workflow (Figures V and W).
- **Managing livestreams** — Switch scenes, start/stop stream, start/stop recording, play ads, and a lot more (Figures X and Y).
- **Security research** — With the ability to automate keyboard inputs, it is possible to take over an entire computer with the push of a button (Figure Z). This is known as BadUSB attack, and it was the original purpose of duckyScript and USB Rubber Ducky.
- **One-finger Twitch chat** — (Figure Aa). 😉

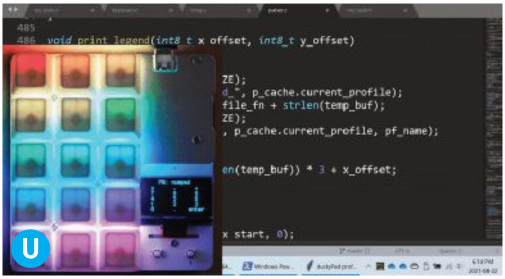
TIPS AND TRICKS

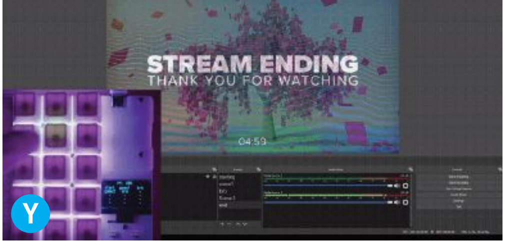
CONSOLIDATE KEYBOARD SHORTCUTS

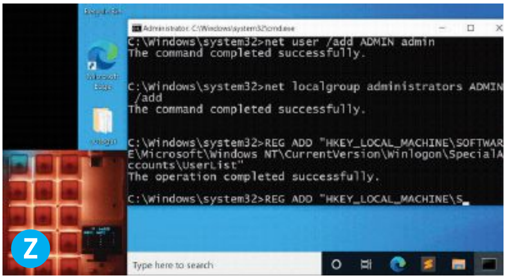
The most obvious use case is putting your commonly used hotkeys on duckyPad! Simply create a profile and add them in.

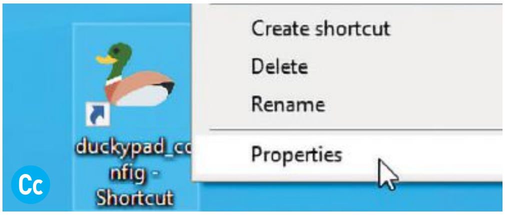
For many applications, you can find an official list of keyboard shortcuts. Just search "<app name> shortcuts" on Google.

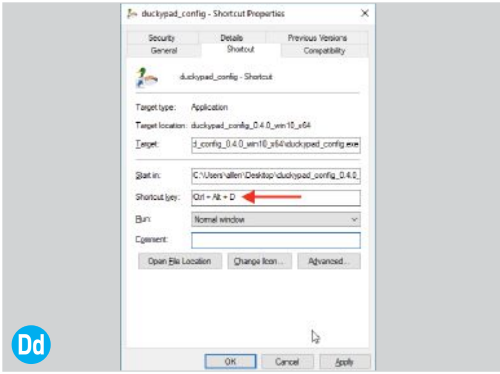


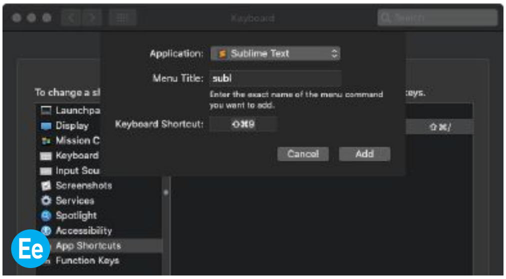












LAUNCH APPS ON WINDOWS USING TASKBAR

Another popular usage is launching apps. There are two ways of doing this. Here's the first. Find the app, then Right-click → More → Pin to taskbar. Now you can use **WIN + <number>** to launch the apps on the taskbar (Figure Bb). In duckyScript, it would be **WINDOWS 1**, **WINDOWS 2**, etc.

LAUNCH APPS ON WINDOWS USING SHORTCUTS

This method works with any file, not just apps! Right-click on any file, then select Send to... → Desktop (create shortcut). Find the new shortcut on your desktop, right-click, and select Properties (Figure Cc).

Set a hotkey in the Shortcut Key box (Figure Dd). Then you can use duckyPad to press this combo to launch anything with a push of a button! In this case, the duckyScript would be **CONTROL ALT D**.

LAUNCH APPS ON MACOS

Go to System Preferences → Keyboard → Shortcuts → App Shortcuts. Press the + button, select an app to open, assign a hotkey, and set up duckyPad accordingly (Figure Ee).

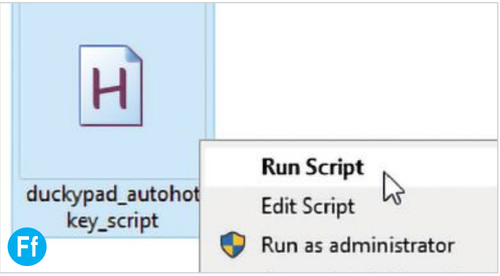
ADVANCED SCRIPTING WITH AUTOHOTKEY

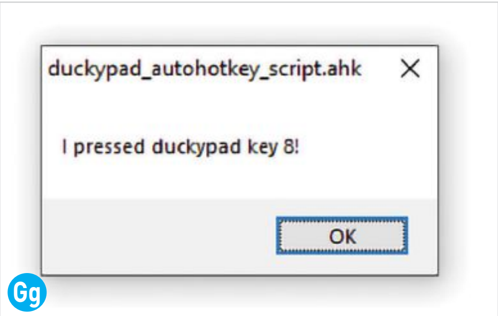
You can use the free and open-source AutoHotkey (authotkey.com) for even more sophisticated needs, such as controlling mouse movements, executing scripts, etc. To do this, set up duckyPad to press a simple combo like **`WIN + F1`**, which then gets captured by AutoHotkey to execute a more complex custom script on your PC.

Download and install AutoHotkey. Then download and save this sample autohotkey script as an *.ahk* file: raw.githubusercontent.com/dekuNukem/duckyPad/master/resources/duckypad_autohotkey_script.ahk. Switch the profile on your duckyPad to *autohotkey*, then run the script by right-clicking and selecting Run Script (Figure Ff).

AHK will execute the script. Press any key on duckyPad, and a corresponding message box will appear (Figure Gg). Now you can write your own AutoHotkey script to do whatever you want! Learn more in *getting_started.md* at the Github repo.

For macOS, you can try BetterTouchTool (folivora.ai) to accomplish the same thing.





Join the duckyPad Owners' Club official Discord at discord.com/invite/4sJCBx5 for discussions, asking questions, sharing scripts, and latest updates!

ALL HANDS ON DECK

"Keeb" nerds are passionate about their hobby: customizing full-size mechanical keyboards. Sure, it's partly about speed — faster typing, coding, or gaming — but it's also about aesthetics: cool design themes, artisanal limited-edition keycaps, colorful lighting, and, obsessively, the sound and feel of different key switches: smooth, quiet ***linear*** switches, ***tactile*** switches with a moderate feedback bump, or loud-and-proud ***clicky*** switches. For a taste of keeb world, check out reddit.com/r/MechanicalKeyboards and the vendors listed at alexotos.com/keyboard-vendor-list and keebmap.xyz.

We're stoked to see makers taking hardware developed for the keeb scene, and using it to build unique DIY input devices like live-streaming decks, video editing decks, and macro pads for work and play. The new Raspberry Pi Pico and other RP2040 boards can act as USB HID devices, making them perfect for all kinds of keyboard projects. Here are some fun builds and kits to try.

1 MACROPACT

github.com/kbjunky/MacroPact

Heat up your iron and solder this 17-key macro pad with IPS display, two rotary encoders, and low-profile Kailh Choc V1 switches and keycaps. Designed by Sean Yin and coded by <kbjunky>, it's powered by a Pico running CircuitPython and KMK — a Python firmware layer for mechanical keyboards — all packed into a sleek 3D-printed housing with RGB lighting.

2 ADAFRUIT MACROPAD RP2040

adafruit.com/product/5128

Compact and functional, this kit packs a lot: 12 hot-swap Kailh linear keys each with RGB NeoPixel, a monochrome OLED display, rotary encoder, plus reset button and on-board speaker. Based on the RP2040 chip, you can code it in Arduino or C but of course

Christian Fehmer in Cologne, Germany, spent 3 years perfecting this custom 60% keyboard in a CNC'd walnut case with Zealio V1 switches, DZ60 PCB, and Signature Plastics "1976" SA-profile keycaps. Pretty sure I wore an OP shirt that looked exactly like this. keebs.fehmer.info

Jai Hidro in the San Francisco Bay Area is running out of space (and money) for his keeb hobby. Those jars are full of key switches. Current favorite: TKC Kiwis. @jaihidro

Christian Fehmer, Jai Hidro, Sean Yin, Adafruit, Pimoroni, Kota Morishi

it's geared to CircuitPython, with several Adafruit libraries to support it. Like the duckyPad kit, this PCB comes presoldered with sockets, OLED, etc., so assembly is simple. More than just a keyboard sidekick, people have turned these into a LinuxCNC control pendant, Home Assistant automation controller, Simon Says game, and much more.

3 NEOKEY EMOJI KEYBOARD

learn.adafruit.com/neokey-emoji-keyboard

Start out super easy by building Liz Clark's fun, four-key macro pad for your favorite emojis. Just plug an Adafruit 1×4 NeoKey Stemma board into a QT Py RP2040, no soldering required.

4 PIMORONI KEYBOW 2040

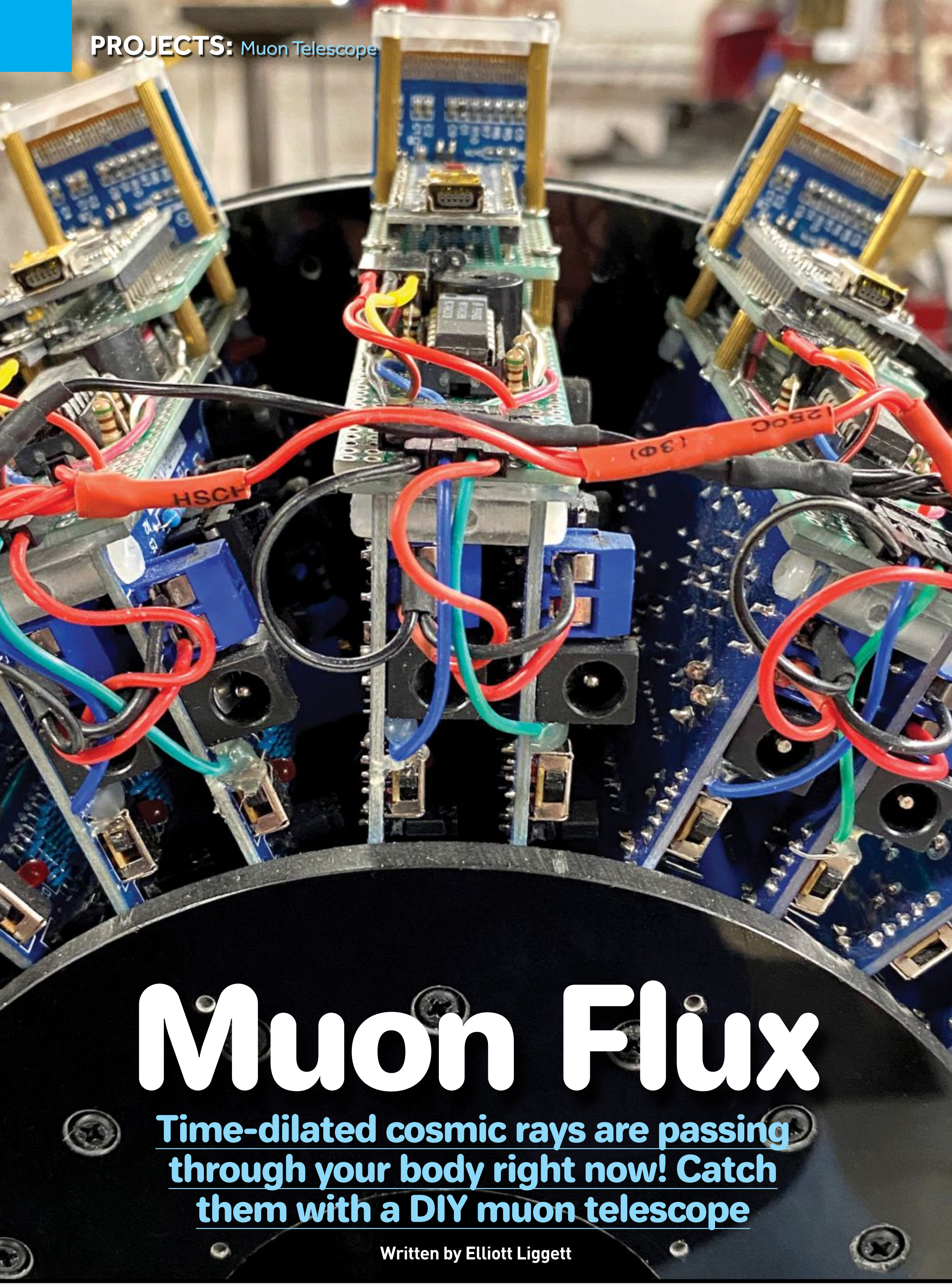
shop.pimoroni.com/products/keybow-2040

Straightforward 16-key pad with per-key RGB LEDs, Kailh Speed switches, and transparent DSA-profile keycaps. The circuit board comes presoldered with RP2040, 2MB flash, reset button, and user-programmable BOOTSEL button, so it's another easy build. Available in linear, tactile, or clicky versions, and supported by a CircuitPython library.

5 PICO RGB KEYPAD STREAM DECK

github.com/kotamorishi/pico-rgbkeypad-streamdeck

Live streamers: Connect a Pico and OLED to Pimoroni's 4×4 Pico RGB Keypad to make this stream deck for OBS Studio. Built and coded by Kota Morishi, it runs on CircuitPython and the powerful Pico RGB Keypad Python module by Martin O'Hanlon. —*Keith Hammond*



Muon Flux

Time-dilated cosmic rays are passing through your body right now! Catch them with a DIY muon telescope

Written by Elliott Liggett

TIME REQUIRED: 30 Hours

DIFFICULTY: Advanced

COST: $600

MATERIALS

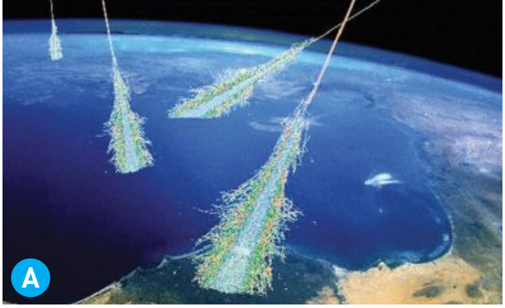
- **Geiger detectors with J3005Br Geiger tube (18)** Geekcreit CAJOE RadiationD-v1.1, assembled back to back in 9 pairs
- **NAND gate ICs, MM74HC08N (9)** used as coincidence detectors for each pair of Geiger counters
- **IC sockets, 14-pin (9)** for the NAND gates
- **Arduino Nano microcontrollers (9)** programmed to record coincidence events for a pair of Geiger counters
- **OLED displays, I²C, 0.96" white, SSD1306 driver chip (9)** for viewing muon counts per second and total time
- **Resistors, 510Ω, ¼W (18)** to reduce input voltage from Geiger counters to the MM74HC08N
- **Piezo speakers, 5V, 12mm (9)**
- **Transistors, S8050 D331 (9)** to drive speakers
- **Resistors, 1kΩ (9)** to modify signal to speakers
- **LEDs, 3V (198)** in 9 × 21 LED strings, to indicate approximate direction of muon travel
- **Lead sheet, 2mm thick** for radiation shielding between Geiger tubes
- **Perf board** as needed to mount components
- **DIP connector pins**
- **Spacers and screws**
- **Acrylic sheet, 6mm and 12mm, and acrylic tube, 4" diameter** for muon telescope mounting

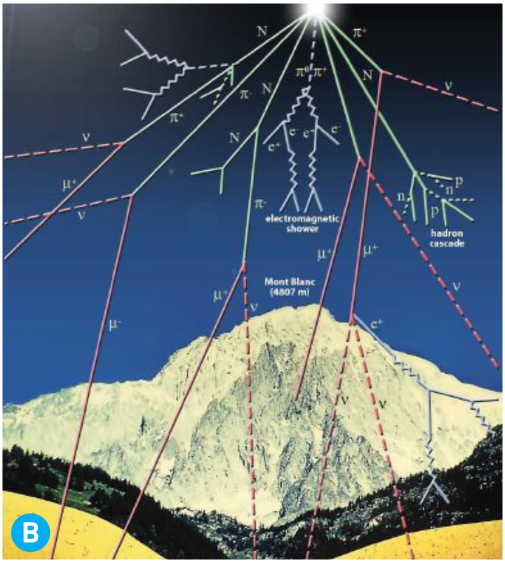
TOOLS

- **Soldering iron** temperature controlled with micro tapered tip
- **Drill press**
- **Table saw**
- **Lathe**

ELLIOTT LIGGETT is a retired materials and mechanical engineer from Lockheed Martin, Loral Defense Systems, and Goodyear Aerospace in Akron, Ohio, but he has always been interested in nuclear physics. He enjoys working in his machine shop, building electronic projects, and reading *Make:*.

Elliott Liggett, Carol Liggett, Simon Swordy (U. Chicago)/NASA), KTH Space Center/CERN





My muon telescope is not an optical telescope — instead, it is used to detect high-energy particles called *muons* and to determine the relative direction they came from. The collision of interstellar particles, called ***primary cosmic rays***, with atoms in the Earth's upper atmosphere creates muons and other particles, called ***secondary cosmic rays*** (Figure A). A muon ("MEW-on," symbol μ^-) is one of several elemental particles that are indivisible, the basic building blocks of all matter.

Muons are interesting because they're heavy compared to other secondary cosmic rays, they travel incredibly fast at 0.998 times the speed of light, and they can penetrate hundreds of feet into the Earth. The lifetime of a muon is only 2.2 microseconds, which is actually not enough time to reach the surface of the Earth from the upper atmosphere. However, at 0.998 light speed, the time experienced by the moving particle increases in accordance with Einstein's theory of general relativity and allows it to reach the Earth (Figure B). This is called ***time dilation.***

Muons can come from every direction of the sky, from straight above to almost horizontal. With my muon telescope I can detect and record muons coming from 9 directions, between 0 to 180 degrees, at the same time. The highest percentage I've detected come from 90° (straight above) and the percentage rapidly decreases as the angle of travel approaches 0° (horizontal). I've detected about 0.176 muons coming from east or west for every 1 muon coming from straight above.

You can learn more about cosmic rays, muons, and muon telescope construction at these sites:

- cosmicray.com.au/what-is-a-cosmic-ray — good graphics and explainer
- kth.se/en/sci/centra/rymdcenter/utstallning/myondetektor-1.813976
- physicsopenlab.org/2019/12/21/muon-telescope

The muon telescopes I've seen on these sites and others use only one pair of Geiger counters or scintillation counters mounted on an arm that can be rotated from 0 to 90 degrees (horizontal to perpendicular) and thus can only measure muon flux from one direction at a time.

MY 180° MUON TELESCOPE

My muon telescope has 9 pairs of Geiger counters spaced 22.5 degrees apart in a semicircle to detect the relative number of muons coming from different areas of the sky, ranging 180 degrees from horizon to horizon.

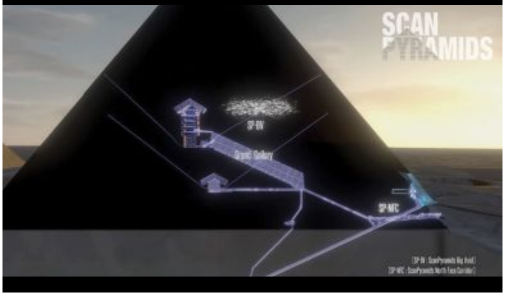
Figures C, D, and E show the front, back, and top of my muon telescope assembly and Figures F and G show the breakdown of one of the nine sub-assemblies. Each contains two Geiger counters for detecting high energy particles, a NAND gate coincidence detector, an Arduino Nano microcontroller to count muons, an OLED screen to record the muon count, and a 2mm-thick lead sheet sandwiched between the Geiger detector tubes to minimize the influence of common background radiation on the muon count. One of the tubes is moved off its circuit board so it's directly in line with the second tube.

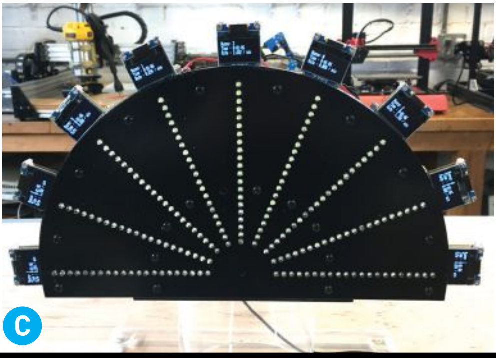
The coincidence detector sends a signal pulse to the Arduino only when a particle passes simultaneously (almost) through both Geiger tubes. This signal is assumed to be a muon because it is the only particle energetic enough to pass through both tubes and the lead sheet at the same time. The passage of a muon is also indicated by an audible click and the illumination of a string of 21 LEDs showing the relative direction of the muon travel (Figure H). Two of the LEDs in each string will also light independently, indicating background radiation passing through only one of the Geiger tubes (Figure I).

The muon telescope can easily be powered

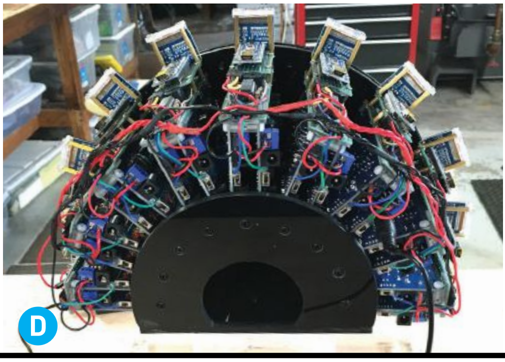
SCANNING THE GREAT PYRAMID

Recently, because of the ability of muons to penetrate massive rock formations, muon telescopes have been used map the inside of the Great Pyramid in Egypt. In 2017 this process, called ***muography***, revealed additional chambers that were not previously known to exist! *—Tom Hammond*



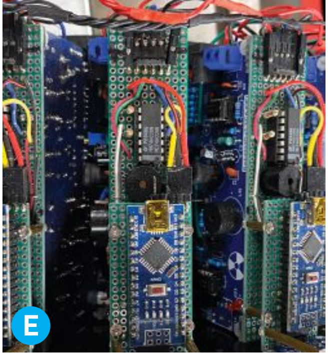


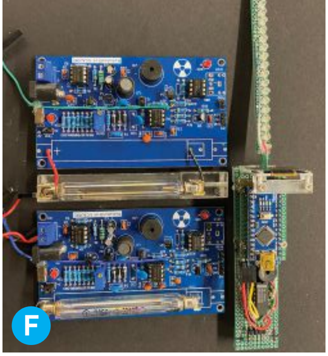
C

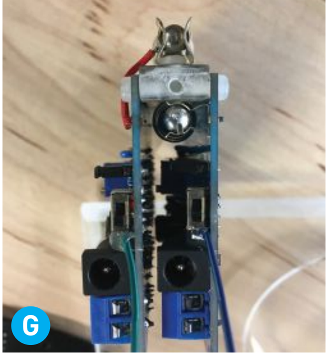


D

ScanPyramids Project/HIP Institute, Elliott Liggett







with a USB cellphone charger providing 5 volts and more than 300 milliamps.

Recently I tilted the whole muon telescope 11.25 degrees to obtain data from additional areas of the sky and repeated north-south, east-west testing for about 5 hours in each orientation. Plots of the combined original and tilted data are shown in Figure J. It appears that currently there may be higher muon flux from the west at about 45 degrees above the horizon at my location here in Ohio. This could change at any time and probably will not be repeatable.

I had some problems with one of the detector modules when I tried to operate the muon telescope for 20 hours. I disassembled it but couldn't find the problem, then reassembled it and it worked as expected. I now think that some of the Geiger counter modules may not be stable over long periods greater than 6 hours — but I can't be sure at this time.

ALTERNATIVE MUON PROJECTS

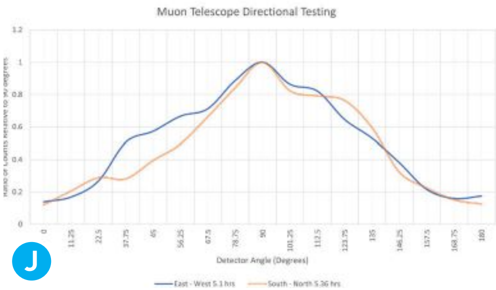
For most people my build would be an advanced project, requiring electronics knowledge, soldering, and machining skills. Here are some easier ways to get started catching muons.

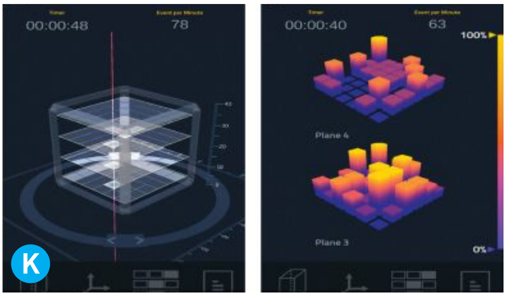
COSMIC WATCH MUON DETECTOR KIT — Developed by Spencer Axani at MIT, Cosmic Watch is a great project kit that can be built for less than $100 and used to detect and count muons. Learn more at cosmicwatch.lns.mit.edu and build it at github.com/spenceraxani/CosmicWatch-Desktop-Muon-Detector-v2.

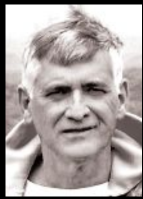
COSMIC RAYS LIVE APP — This iPhone app provides a 3D view of muons passing through detectors in several locations around world in real time (Figure K). The one at Canfranc Underground Laboratory in Spain is 850 meters beneath rock "overburden." The angle, size, and magnification of the detector view is controllable within the app and the results are fascinating when you can visualize the muon tracks and then come to the realization that these highly energetic particles are continuously penetrating our bodies more easily than they can penetrate rock!

Muon strike detected from 45 degrees elevation and displayed as a flash of 21 LEDs.

Non-muon background radiation causes single LEDs to flash.







FORREST M. MIMS III is an amateur scientist and Rolex Award winner, was named by Discover magazine as one of the "50 Best Brains in Science." He has measured sunlight and the atmosphere since 1988. forrestmims.org

Cosmic Dust & Space Debris

Tweak the *Make:* twilight photometer to detect micrometeors and orbital junk 400 miles high

Written and photographed by Forrest M. Mims III

Six years ago, I described in these pages how to build a twilight photometer (Figure A) to determine the presence and altitude of overhead layers of smoke, dust, and smog ("Build a Twilight Photometer to Detect Stratospheric Particles," *Make:* Volumes 44 and 45, makezine.com/projects/twilight-photometer).

That two-part article has become a serious project to detect and measure the elevation of stuff in the sky, including dust arriving from Africa's Sahara Desert, smoke from agricultural fires in Mexico and Central America, and most interestingly, high-altitude cosmic dust that accompanies meteor showers as Earth orbits the sun. Some of this "dust" might actually be space debris — manmade junk orbiting the Earth.

My twilight photometers are much simpler than professional versions, and that original *Make:* article has been read by 76 scientists who are members of the ResearchGate community and 719 non-members (and, hopefully, many *Make:* readers). It hasn't yet been cited in a peer-reviewed scientific paper, but at least one is on the way. Maybe you can build a twilight photometer and cite it in your research.

This article can be considered Part 3. Here I describe how the original project, which uses an LED as a photodiode to detect the very weak twilight glow, has been expanded to detect particles high above those in the troposphere and stratosphere. I close with a description of a new kind of low-noise LED twilight photometer.

MICROMETEORS

According to the American Meteor Society, most meteors that flash across a dark sky range in size

TIME REQUIRED: 2–5 Hours

DIFFICULTY: Easy/Intermediate

COST: $15–$225

MATERIALS

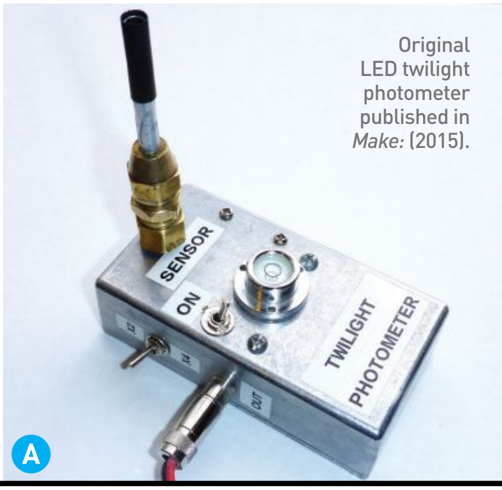
FOR THE ORIGINAL TWILIGHT PHOTOMETER, see detailed parts and tools lists at makezine.com/projects/twilight-photometer.

FOR THE NEW LOG CONVERTER TWILIGHT PHOTOMETER shown in Figure H, you'll need the following:

- **AD8304 160dB logarithmic current-to-voltage converter breakout board** available from Amazon, eBay, and others
- **RG174 coaxial cable with female SMA connector, 6"–8" long** Amphenol CO-174SMARSMA-000.6 or similar
- **Heat-shrink tubing, black, 4"–6" long, width of your LED** to form optical collimator
- **Near-infrared (IR) LED, 880nm wavelength** commonly used in TV and other remote controls
- **Rechargeable battery back, 5.5V** with short USB connector cable
- **16-bit data logger (optional)** Ideally you'd use something like the Onset HOBO 4-channel logger. But you can also record the data by hand by connecting the output of the log converter to a digital voltmeter (DVM) and recording the signal at regular intervals of 10–30 seconds.

TOOLS

- **Soldering iron and solder**
- **Drill and bits**
- **Screwdriver**
- **Digital voltmeter**
- **Compass or compass rose**
- **Computer with internet access and LibreOffice spreadsheet software** free download at libreoffice.org



Original LED twilight photometer published in *Make:* (2015).

from a grain of sand to a small pebble. Meteors can occur at random or during predictable showers, and each year millions of them survive their trip through Earth's atmosphere. According to an international study in Antarctica recently reported in *Earth and Planetary Science Letters* (doi.org/10.1016/j.epsl.2021.116794), 5,200 tons of micrometeors reach Earth's surface every year.

The timing of meteor showers is predictable because they occur when Earth passes through the remnants of the tail of a comet with a well-known orbit around the sun. My twilight photometers detect multiple, well-defined layers of cosmic dust during meteor showers. They also detect random layers of cosmic dust when Earth's orbit passes through swarms of dust, some of which may have originated from collisions between asteroids.

SPACE DEBRIS

A recent paper in *Monthly Notices of the Royal Astronomical Society* by M. Kocifaj and team (doi.org/10.1093/mnrasl/slab030) reports that space debris has increased the intensity of light from the night sky by 10%, which is raising serious concern among both professional and amateur astronomers. As more satellites are launched, the sky glow problem will become even worse.

It's possible that some of the aerosols detected by my twilight program are space debris. Millions of fragments from satellite collisions and at least one intentionally destroyed satellite are orbiting Earth. Figure B (on the following page) is a NASA image of space objects within 2,000 km (1,243 mi.) of Earth's surface as of January 1, 2019, from the Orbital Debris Program Office (orbitaldebris.jsc.nasa.gov/photo-gallery). Some 95 percent of these objects are space debris, and only those large enough to be tracked are shown.

THE ATMOSPHERE

The atmosphere protects Earth from most meteors. That's because meteors are moving so fast (25,000mph to 160,000mph) they began to incandesce and burn up when they enter the upper atmosphere. Most space debris eventually meets a similar fate.

The lowest layer of the atmosphere is the ***troposphere***, which is around 10 to 15 kilometers

(6–9 miles) thick. The troposphere is where weather occurs. The next layer, the ***stratosphere***, extends from the top of the troposphere (the ***tropopause***) to 50km (31 mi.). Most of the ozone layer is within the stratosphere. The ***mesosphere*** lies above the stratosphere and reaches an elevation of 85km (53 mi.). The ***thermosphere*** then takes over and reaches several hundred or more miles. Most meteors burn up in the thermosphere from 80–120km (50–75 mi.) high.

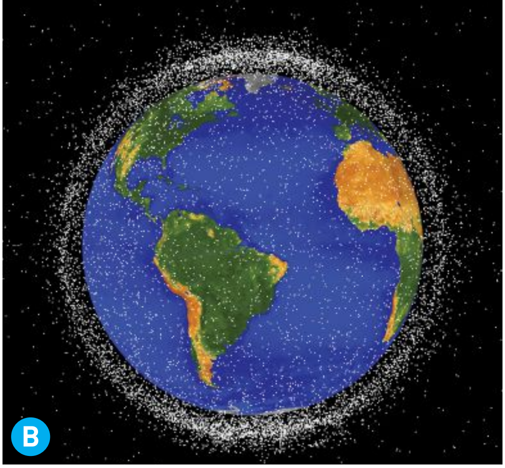
Figure C is a photograph of the atmosphere and Earth's limb over Patagonia on October 27, 2016. This remarkable image was captured by an astronaut of the International Space Station's Expedition 49 crew while in orbit 400km (250 mi.) over the South Atlantic Ocean (nasa.gov/image-feature/fiery-south-atlantic-sunset).

Look carefully and you'll see multiple layers just like those that are detected with a twilight photometer. The red and orange layer is the troposphere, brightly colored by dust from the desert below. The tropopause over Patagonia is around 12km (7.5 mi.) high, which means the fuzzy brown, blue, and white layers above the troposphere indicate the stratosphere, which is usually much more uniform in appearance.

Figure D shows an intensity scan superimposed over a narrow section of the photo in Figure C. This scan was made with ImageJ image analysis software (free from imagej.nih.gov/ij). The estimated scale correction factor on the vertical *y* axis assumes that the top of the orange-red layer is the tropopause. This means the top of the stratosphere is approximately at the 685 point on the *y* axis, and this is where the mesosphere begins.

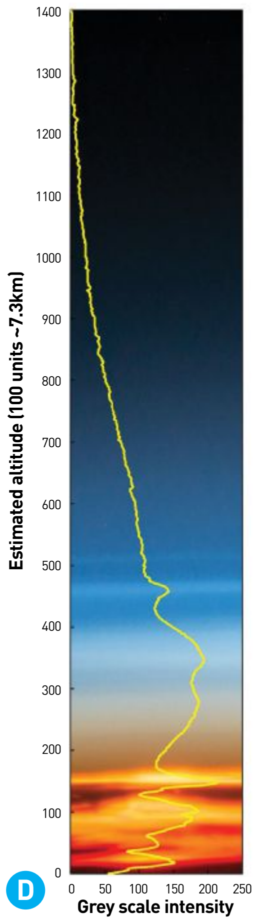
The intensity scan in Figure D clearly shows dust layers in the troposphere and unknown aerosol layers in the stratosphere. Based on my hundreds of twilight photometer scans of the evening sky, even very small undulations suggest the presence of aerosols. Therefore, some of the small bumps in the scan within the mesosphere are probably caused by aerosols too.

Figure E is a twilight photometer profile over my Geronimo Creek Atmospheric Monitoring Station (a field by my office) that shows a thick layer of dust from the Africa's Sahara Desert. This stuff arrives over my site most summers.



Space objects and debris in low Earth orbit actively tracked by NASA.

ISS photo of Earth's limb and sky above Patagonia.



ImageJ intensity scan of a slice through the atmosphere in Fig. C.

My photometers also detect smoke, smog, and even volcanic plumes like the one from Russia's Raikoke volcano.

ULTRA-LOW NOISE TWILIGHT PHOTOMETERS

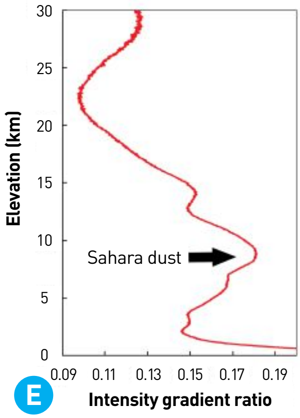
The *Make:* columns that describe how to build and use my original LED twilight photometer are still valid. I've since built two additional twilight photometers with a total of six channels from the red to the near-infrared. Figure F shows them in a plastic ammo box along with their data loggers. Unfortunately, two of the channels produce very noisy data when detecting aerosols in the stratosphere. I've since learned that an LED connected to the input of a logarithmic current-to-voltage converter produces exceptionally clean data up to and even exceeding 300km (186 mi.).

The log converter I use is the discontinued Dawn Electronics DN120 10-decade current-to-voltage converter (Figure G). I bought two on eBay for $33 each, not bad for devices that cost more than $700 new. If you cannot find a surplus DN120, a possible substitute is the Analog Devices AD8304 8-decade log converter chip. Several offshore companies sell compact AD8304 breakout boards; I bought two for $25–$30 each.

Figure H shows a quickly assembled twilight photometer made by stacking an AD8304 board atop an Onset 16-bit data logger and a 5.5-volt backup battery designed to charge a mobile phone. While these log converters do not have as much dynamic range as the DN120, preliminary data from two twilight sessions looks good. Testing will resume once the clouds clear out between my site and several hundred miles west.

GOING FURTHER

So far, I've accumulated several hundred twilight scans using the two instruments based on my original design in *Make:* and around a hundred scans using the DN120 current-to-voltage converter. After clear skies return this fall, I'll add an AD8304 photometer and lengthen the twilight scans to two hours or more to see what happens up to 1,000 km (621 mi.) or more overhead. It's possible that space debris might reveal itself as brief spikes in the intensity scans. Meanwhile, I'm working on a formal paper about this research.



Twilight photometer scan showing Sahara dust over Texas (July 8, 2020).

Two of the author's twilight photometers and data loggers have been in active use since 2014.

Advanced DN120 10-decade current-to-voltage converter, configured as a twilight photometer.

Readily available Analog Devices AD8304 8-decade current-to-voltage converter board atop a data logger and 5.5-volt battery forms a simple LED twilight photometer. The black heat-shrink tube forms a collimator when pressed over the LED.

CHRIS BORGE is a hobbyist designer and 3D printer in South Australia. He works in media and is studying for a marketing degree.

Turn It Up!

Learn basic wood turning with this cheap 3D-printed lathe

Written and photographed by Chris Borge

Wood turning is a fun hobby that is simple and satisfying to pick up. Turning has been around for centuries and it's no surprise why — the results are stunning, and even easy turning projects can give a professional feel to many a design. I've been surprised where I find myself using it. I like the personal feel of being able to turn small pieces of timber I've gathered myself into useful parts for various projects, from tool handles to kitchen utensils.

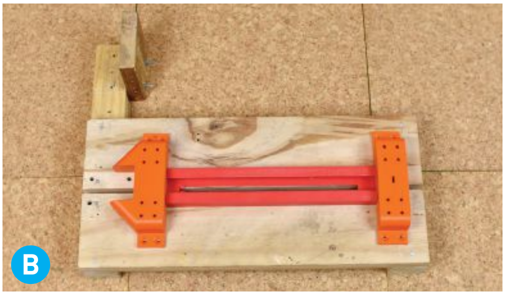
In this project I show how to make your own simple wood turning lathe using 3D-printed parts and some hardware you may already have on hand. The print job is simple but the build requires some hand tools and electronics knowledge.

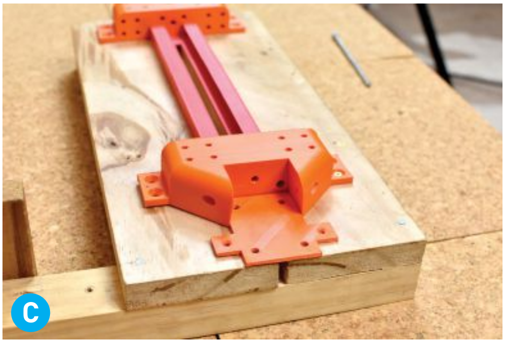
A full video tutorial is available online, along with all the files to print the parts, at thingiverse.com/thing:4874288. Here I'll provide some build notes about the important components.

THE BASE

The base of the lathe consists of a few simple pieces of lumber, arranged as shown in Figure A. Apart from the obvious size decisions, there are two essential design elements in the base: the gap down the middle of the boards and the gap under the boards. These two gaps allow you to adjust the tailstock and tool rest clamping mechanisms; both of these components have a threaded rod that clamps under the lathe using a knob. Originally I didn't have this system in place and the tailstock was unusably wobbly.

You could do without this setup and attach the





lathe directly to a bench if you were to use a fixed tailstock, but this limits you to turning one length of timber, which could be wasteful for smaller projects.

THE BED

The bed consists of the four components shown in Figure B. The two end pieces (orange) attach to the V bed (red) via four threaded inserts. The headstock is then bolted on, and this assembly is used as a template to drill the holes for mounting it to the base.

Once this assembly is mounted, the headstock is then removed and a special headstock template piece is used to align the final four holes, as shown in Figure C. This assembly is designed to be simple to assemble and mount. The V bed is shaped this way so when the tailstock clamps down, the bed is compressed on the V of the tailstock, keeping it more firmly in place. The only downside of a V-shaped bed like this is that it is harder to clean, but not by much.

TIME REQUIRED:
2–3 Hours + 50 Hours (Print)

DIFFICULTY: **Intermediate**

COST: **$15–$50**

MATERIALS

» **3D-printed parts** Download the free files for printing from thingiverse.com/thing:4874288. I actually did this entire project in PLA, but ABS or another heat-resistant plastic would almost certainly be the better choice.

» **Steel rod, 5/32" (4mm) diameter, *x*mm lengths (2–5 total)** See text for details about these dimensions.

» **Threaded rod, 3/16" (5mm) diameter, lengths of: *x*mm (1), *y*mm (1), *z*mm (4), and *w*mm (1)**

» **Small wood screws (13)** less than 4mm in thread diameter

» **Bolts, 3/16", Phillips head, 40mm long**

» **Bolts, 3/16", hex head, *x*mm long**

» **Aluminum strip, 32mm wide × 3mm thick, 10cm total length** in three 32mm sections

» **M4 bolts, 18mm (2)**

» **M3 bolts, 20mm (4)**

» **M2 bolts, 20mm (2)**

» **M2 brass threaded inserts, 10mm or shorter (2)**

» **M3 brass threaded inserts, 10mm or shorter (4)**

» **Nuts, 3/16" (15)** ideally 3mm thick and 8mm wide between the flat sides of the hexagon

» **Ball bearings, 5mm ID, 10mm OD (6)**

» **ER11 collet chuck, 5mm bore** for mounting the workpiece

» **Stainless steel rod, 5mm diameter, 90mm long** Can be bought or can be found in printers and photocopiers, which are both commonly thrown out and are a great source of parts for projects including rods, belts, steppers, encoders, wires, and more.

» **Motor and belt setup** An old sewing machine motor provides a great torque-rpm range, at around 3,000rpm which is good for small workpieces. A **cordless drill** chucked onto the back of the headstock is another simple option.

» **Scrap lumber** for the base

TOOLS

» **3D printer** with at least 240mm of Z travel. I use an Ender 3.

» **Hacksaw**

» **Drill press or hand drill with bits: 5mm and 2mm**

» **Wood saw**

» **Hammer**

» **Safety gear** for operating the lathe, ideally a full face shield and hearing protection

» **Chisels** for using the lathe. Regular wood chisels can work but turning chisels are ideal.

» **Screwdrivers**

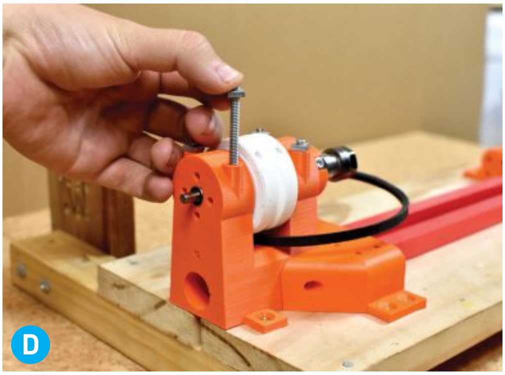
» **Adjustable wrenches** aka spanners

THE HEADSTOCK

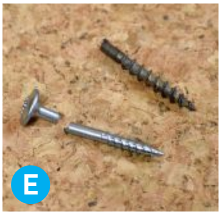
Once the headstock template is removed and the headstock is replaced, it can be secured using two wood screws and four 3/16" threaded rods (Figure D). These rods run the length of the headstock and clamp it down securely under the base. The headstock is the strongest part of the whole lathe, and for good reason. Because the workpiece is attached to it, if something goes wrong the headstock will have to take the bulk of the force. 3D prints are weakest along their layer lines, and I wanted to avoid failure via layer separation. These rods prevent it — for the layers to separate completely, the nuts on the threaded rods would also have to fail — making the lathe much stronger.

You could theoretically print the headstock in another orientation, however this runs the risk of splitting the print when tightening the rods, which are still required regardless to keep the headstock rigid.

Apart from these rods the headstock is constructed from four bearings, a length of 5mm stainless steel rod, a pulley, and an ER11 collet chuck. I chose this collet chuck because it's pretty affordable and allows me to mount using a wide range of hardware. To mount my workpiece I usually just use a wood screw with the head cut off (Figure E), and this has worked well for me.



D



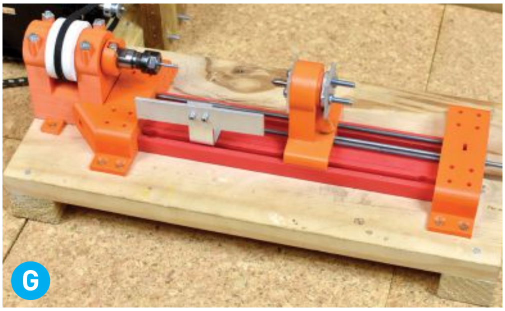
E

F

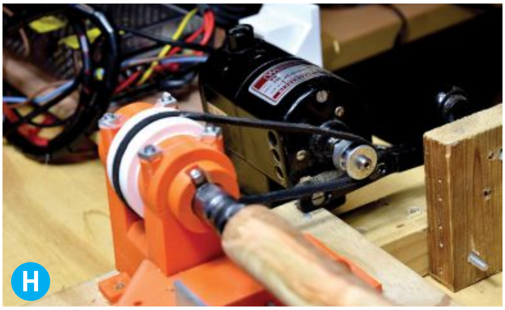
THE TAILSTOCK

The tailstock was actually the most complicated part to get working, and had the most failures, but this latest design has worked well. The tailstock consists of three aluminum plates, two bearings, a center (a bit of sharpened rod), and a threaded rod to clamp it all down. The tailstock can also have pressure applied from behind as shown in Figure F, to further increase its rigidity. Early tailstock designs without this clamping force would shake loose and break if I let them run for long enough. (Usually it's immediately evident that a design didn't work and I'd stop the test.)

Another issue I had with the tailstock was the bearings overheating and melting the housing, which interestingly isn't an issue I've had with the headstock. The aluminum plates are how I overcame this. The three plates function as a stronger guide for the tailstock, and also act as a



G



H

I

heat sink. With this setup I've had zero failures of the tailstock.

TOOL REST

The tool rest comes in two versions: a simple 8mm rod that runs the length of the workpiece, and a new, more compact, adjustable "T rest" version (Figure G). I've used Version 1 for many projects without much issue, but being able to get the tool rest closer to the workpiece is useful for wood turning, so I designed the new T rest to allow adjustment for a variety of workpiece sizes.

MOTOR

The best option here is a sewing machine motor. They are powerful, readily available secondhand, spin at an ideal rpm, and usually run off a foot pedal, which is great for both control and safety. I use the default belt that came with the sewing machine motor, and I designed a pulley to fit it (Figure H). I've also experimented with leather belts, like those used for small horological (watchmaking) tools; these worked fine and are easy to make in whatever size you need, but I found the original belt to be more convenient.

Some other motor options include drills, which run at a good rpm of up to around 2,000 with high torque, or a geared-down DC motor off from somewhere like eBay. I've found that 2,000–3,000rpm is OK for a tiny lathe like this but you would not want to do big turning at those speeds! Rotary tools are unfortunately too fast with not enough torque — a Dremel for example runs at 36,000rpm which is much too fast to be safe. It might be possible to gear down a Dremel to work, however.

CHISELS

To use the lathe you will need chisels, and while not ideal, regular wood chisels can work, at least at the beginning. If you find you enjoy woodturning, it's definitely a good idea to get proper ***turning chisels***, aka ***lathe chisels*** — it's faster and won't ruin your bench chisels.

TURNING PROJECTS

Finally, pictured here are some projects I've done on this lathe, including what each one started as, and how it ended up. As you can see, it's more capable than just turning bits of dowel! The tool handle shown in Figure I uses a chunk of Australian red gum that's over a century old and hard as a rock, as well as being very unbalanced. I considered this the ultimate test for the lathe and it handled it flawlessly.

Other projects include a honey dipper stick from an olive branch (Figure J), and a tiny hammer from Tasmanian oak (Figure K), which found use as a gavel for an 11-year-old's debating class, and survived!

J

K

FUN AND FUNCTIONAL

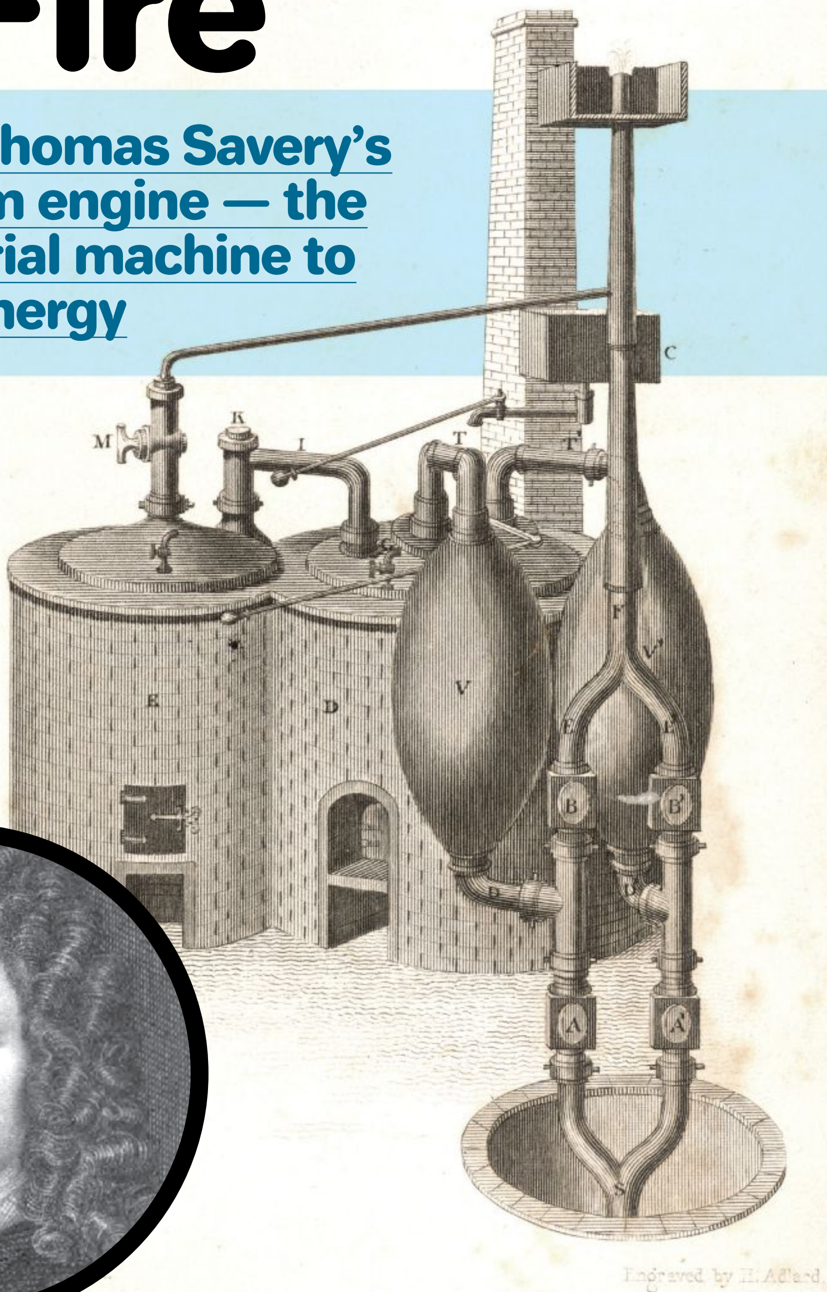
This lathe is a cheap but capable option for someone who is only just starting out, or perhaps has very little space. It works well, but there is plenty of room for improvement. The back of the headstock, for example, has another mounting rod sticking out which could accept perhaps a small disk sander or grinder attachment to add extra function to the machine. I've also thought about using a belt printer to print a longer V-bed; a goal of mine is to get it long enough to turn my own drumsticks on it.

Overall it's a fun weekend project and a great example of how 3D printing can be used for making functional parts.

Raise Water by Fire

Re-create Thomas Savery's proto-steam engine — the first industrial machine to use fossil energy

Written and photographed by William Gurstelle



TIME REQUIRED: A Weekend

DIFFICULTY: Intermediate

COST: $40–$60

MATERIALS

- **Black iron pipe nipple, 2" diameter, 2" long not including threads**
- **Black iron pipe caps, 2" diameter (2)**
- **High temperature plastic tubing, ¼" ID (10 feet)** e.g. McMaster-Carr 5239K19
- **Nylon check valves (2)** for ¼" ID tubing; McMaster 47245K27
- **Steel barbed hose fittings, ¼" ID hose barb × ¼" NPT male (4)**
- **Tee fitting, plastic or brass** for ¼" ID tubing
- **Brass on/off valve with lever handle, ¼" NPT female**
- **Water buckets (2)**
- **Heat source** such as alcohol lamps, a Bunsen burner, or propane torch
- **Scrap lumber**

IF YOU BUILD YOUR OWN STEAM GENERATOR, ADD:

- **Black iron pipe nipple, 2" diameter, 2" long not including threads**
- **Black iron pipe caps, 2" diameter (2)**
- **Pressure relief valve, 15psi or less** or use a 0–50psi adjustable pressure relief valve set to 15psi or less
- **Sight glass** to fit ½" NPT hole
- **Pipe thread sealing compound**

TOOLS

- **Electric drill and 7/16" bit**
- **Cutting oil**
- **Pipe tap, ¼" NPT with tap handle**
- **Hand saw**
- **Adjustable wrench**
- **Clamps**
- **Tap and drill set, ½" NPT (optional)** if you build your own steam generator

WILLIAM GURSTELLE's book series *Remaking History*, based on his *Make:* column of the same name, is available in the Maker Shed, makershed.com.

Wikimedia commons, Adobe Stock-Archivist

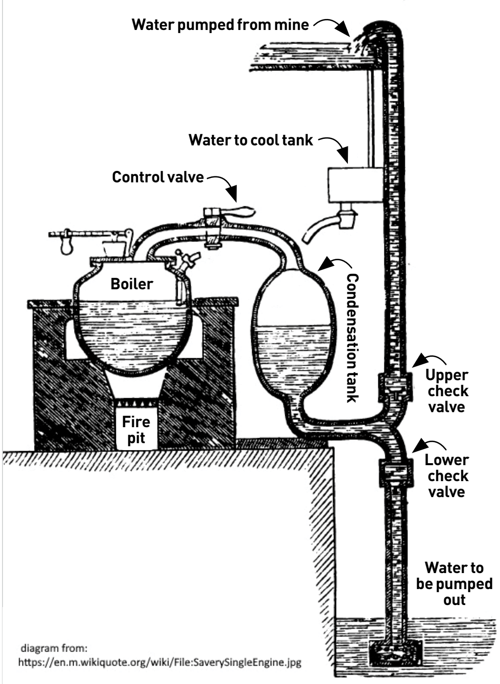
The engines of the Industrial Revolution that ran the machinery in the early textile mills, turned the screws of steamships, and powered the drive wheels of locomotives were made of iron and powered by coal. At the beginning, both the iron ore and the coal came from mines located in England. Mining was hard work, and one of the many difficulties associated with mining was the seeping water that constantly flooded the mines. "Dewatering" was a necessity in nearly all mines, and still is today.

In 1698, Thomas Savery came up with an idea for solving this problem. Savery was a former military engineer from a tin- and copper-mining county in southwest England. His military rank was "trench master," meaning he was in charge of building underground constructions. He must have been skilled at it because he rose to the title of Captain Savery.

After leaving the military, Savery continued to think about life below ground, and he judged the old ways of pumping water inadequate. "For more than an hundred years," he wrote in 1702, "men and horses would raise ... as much water as they have ever done, or I believe ever will." He had long thought about ways to remove water from places underground, and he had "happily found a new and much stronger and cheaper force for powering pumps." That force, he explained, was fire.

In retrospect, Savery's invention, which he christened "the Miner's Friend," was more than just a pump. It was in fact the first machine that used the energy within fossil fuels to do useful work on an industrial scale. The Miner's Friend was neither efficient nor elegant, but it was important. With it, Savery unhitched industry from the limitations of animal, wind, and water power.

Savery's machine consisted of a large, sealed iron tank. The tank was filled with steam, piped in from a large boiler. The machine operator closed the steam valve and then sprayed the outside of the tank with cool water. That caused the steam inside to condense back into water. Since liquid water occupies only a small fraction of the volume of steam, a strong vacuum was created inside the tank. The tank, connected by another pipe controlled by two check valves to the mine water that needed to be pumped out, drew up the water. When the tank was full of water, the steam



valve was reopened and the steam forced the water out of the tank, out the pipe, and out of the mine. The process was repeated over and over.

In this Remaking History column, we will build a model of Savery's Miner's Friend. Although it was a cumbersome and inefficient machine — today it's remembered mostly as the precursor to the far more successful Newcomen and Watt steam engines (which had pistons) — it is fun to observe in action and teaches important historical and engineering lessons.

Note carefully that it is a steam-operated machine. Steam, if not given respect, can be dangerous. As always with DIY projects of this nature, proceed at your own risk.

BUILD THE MINER'S FRIEND

1. BUILD THE STEAM GENERATOR (OPTIONAL)

Drill and tap two ¼" NPT (pipe threaded) holes in one of the 2" steel pipe caps as shown in Figure A. (Note that a hole for a ¼" NPT thread is a much different size than for a ¼"-diameter bolt. You need a ¼" NPT tap and the appropriate 7/16" drill bit.) In one hole, screw in the NPT-to-barbed fitting connector; in the other hole, screw in the pressure relief valve, applying pipe compound to

⚠CAUTION: SAFETY WITH STEAM

You are working with steam which is inherently dangerous. If the tubing breaks or comes loose, the escaping steam can cause scalds. Wear gloves, heavy clothing, and face protection when building or operating the Miner's Friend.

I used iron pipe and fittings, and an adjustable pressure relief valve for a makeshift steam generator. I had no problem with this setup, but a better, safer, and more expensive solution would be to buy a model steam engine boiler to serve as a steam generator.

- Install a 15psi (or less) pressure relief valve and a sight glass on the steam generator. The sight glass allows you to check water levels inside the steam generator. *Never let the steam generator run dry.*
- There is no piston in the Miner's Friend, and thus no need for pressurized steam. Therefore, leave the control valve open at all times except when actually pumping water and even then, never leave the control valve closed for more than 5 to 10 seconds.
- Make certain the check valves are oriented correctly.
- The tube from the condensing tank to the upper reservoir must be open to the atmosphere at all times, so always keep the tube's opening *above the waterline* in the upper reservoir. To do so, use clamps to keep the components in position. Further, be sure that there are no kinks in the tubing, and that the hose barbs on the fittings are fully inserted into the tube.
- All "hot side" components (tubing, valves, and fittings between the inlet of the condensation tank and the steam generator) must have an upper working limit of at least 250°F.
- Use pipe compound on all pipe threads to prevent leaks.

the threads of both.

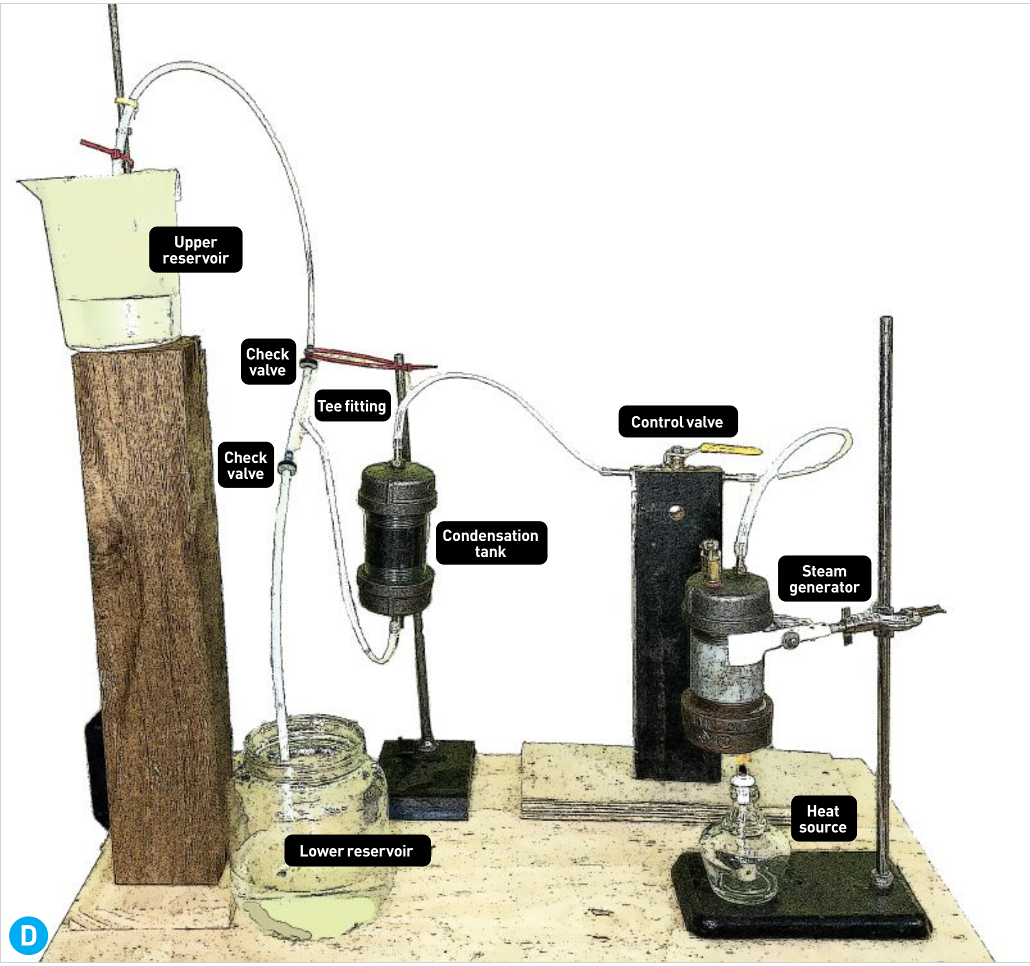
Next, install the sight glass. The sight glass allows you to see that there is water inside the steam generator. To install, drill a hole in the side of the iron nipple using the drill that came with the ½" NPT tap set. Drilling a hole that big in a pipe is not at all easy, so clamp the work securely, use cutting oil, and go slowly. Tap the hole with the ½" NPT tap. Apply pipe compound and twist in the sight glass.

2. PREPARE THE CONDENSATION TANK

Drill a 7⁄16" hole, then tap threads using the ¼" NPT tap, in each of two 2" steel pipe caps as shown in Figure B. Insert an NPT-to-barbed fitting connector in each hole.

3. MOUNT THE STEAM CONTROL VALVE

Attach the on-off valve to a short length of 2×4 lumber to hold it steady while you turn it on or off. Use screws to affix the valve if the valve body accommodates them; otherwise, cut a slot in the 2×4 and press the valve body into the slot (Figure C).



4. CONNECT IT ALL
Use the high-temperature tubing to connect the components. Figure D shows the plumbing arrangement for the reservoirs, condensation tank, check valves, control valve, and steam generator. To support the reservoirs, condenser tank, and steam generator in the correct position relative to one another, you can use lab stands such as those you may have seen in chemistry class, or fabricate your own out of wood or iron.

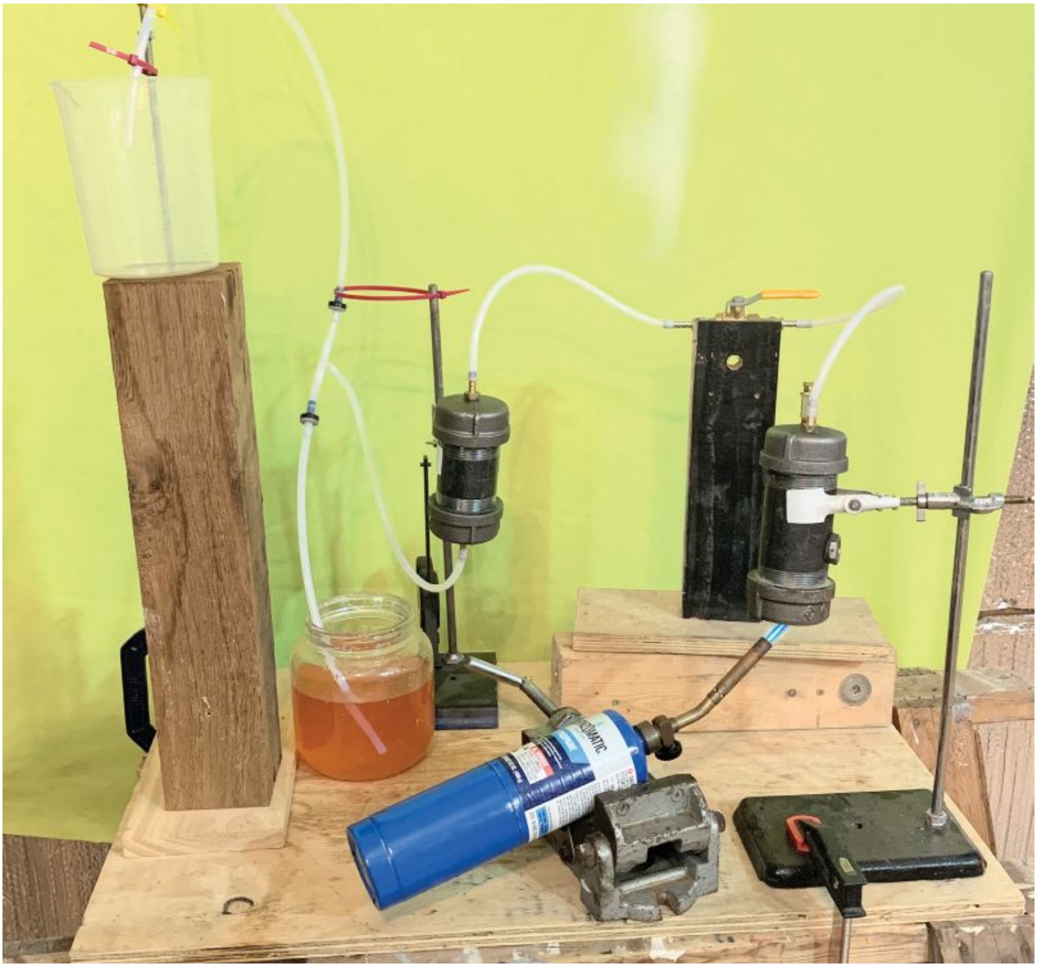
OPERATING THE MINER'S FRIEND

Steam is channeled into the condensation tank via the control valve. When the operator closes the valve, the steam in the tank, now deprived of heat from the flame source, quickly condenses and causes a vacuum. The vacuum sucks up water from the lower reservoir. When the operator reopens the control valve, steam pushes the water out of the tank and into the upper reservoir. The pair of one-way check valves constrain the water to move only from the lower reservoir to the higher one. When the control valve is then closed again, the steam in the tank recondenses and the process repeats. While Savery sprayed the tank with cold water, in this model setup, there is no real need to do so.

1. Remove the pressure relief valve. Fill the steam generator with water to the top of the sight glass. Replace the pressure relief valve.

2. Open the control valve.

3. Using a propane torch, Bunsen burner, or large alcohol lamp, apply heat to the bottom of the



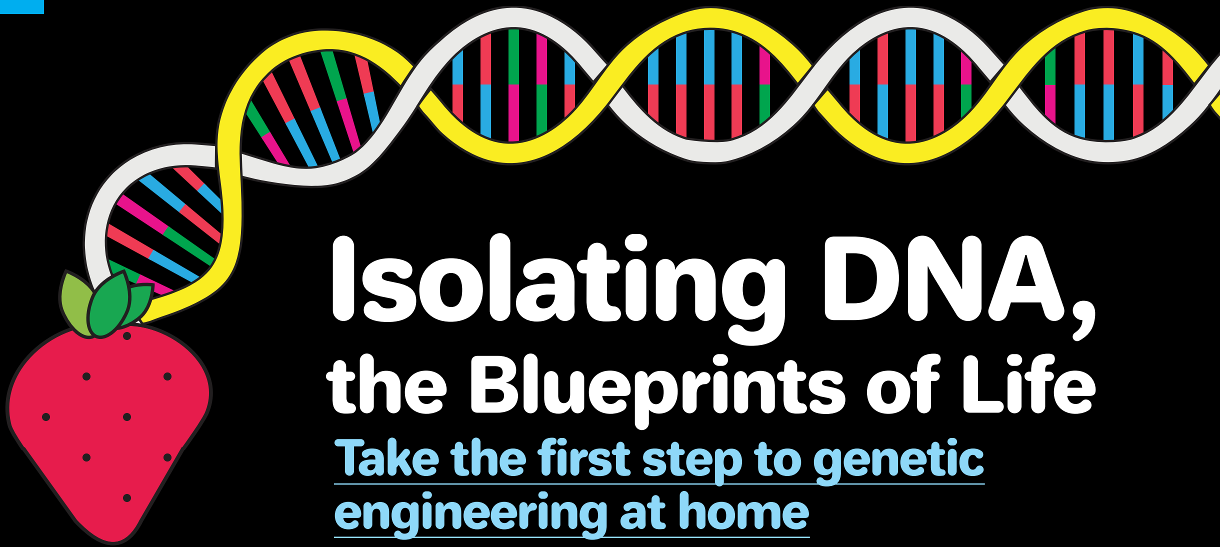
generator. Continue heating until you observe steam exiting the generator through the tube. Turn down the heat to the level in which a small but continuous generation of steam is attained. A fairly vigorous boil is required.

4. Continue heating. When you notice steam exiting the tube that discharges into the upper reservoir, close the control valve for 5 to 10 seconds. Deprived of new steam, the tank will cool, and the steam inside the tank will condense into liquid. Since liquid water takes only a small fraction of the space of steam, a vacuum is produced inside the tank. The vacuum draws the water up from the lower reservoir through the lower check valve and into the tank.

5. Open the valve. Steam reenters the tank, pushing the water out through the upper check valve and into the upper reservoir. Your Miner's Friend is pumping water!

6. Repeat steps 4 through 6 as desired.

NEED A LIFT? Try this mad-scientist project that raises water using a similar principle: John Edgar Park's vacuum siphon coffee brewer. The boiling flask (right) forces hot water down to the grounds in the brewing flask (left), then cools, creating a vacuum that pulls delicious filtered java back up to the boiling flask for serving. Mmm. makezine.com/projects/the-florence-siphon



Isolating DNA, the Blueprints of Life

Take the first step to genetic engineering at home

Written by Justin Pahara and Julie Legault

TIME REQUIRED: 30–60 Minutes

DIFFICULTY: Easy

COST: $10–$20

MATERIALS

- **Soft fruit** A strawberry is best, but ½ a kiwi or 1/5 a banana can work.
- **Distilled water or bottled water, 1 tablespoon**
- **White salt, ¼ teaspoon**
- **Shampoo, hand, or dish soap, translucent (non-creamy), ¼ teaspoon** containing SLES (sodium laureth sulfate) and EDTA (ethylenediaminetetraacetic acid)
- **Paper coffee filter** #4 or #5 are a good size to fit in a cup.
- **Very narrow glass** such as a shot glass
- **Small resealable sandwich bag**
- **Isopropyl alcohol, 91%–99%, 2 tablespoons** aka rubbing alcohol. 70% strength will also work.
- **Small drinking cup**

DR. JUSTIN PAHARA is a Cree scientist-entrepreneur from a Southern Alberta Canadian farm, with a Ph.D. in biotechnology and bioelectronics from the University of Cambridge.

JULIE LEGAULT is a designer-entrepreneur from Montreal, with an M.A. from the Royal College of Art in London, and an M.S. from Massachusetts Institute of Technology (MIT) Media Lab.

During this hands-on activity from our new book *Zero to Genetic Engineering Hero*, you'll mash a piece of fruit so that the cells that make up the fruit become separated. You will then break open these cells to release the inside components into the outer environment. This includes releasing the cell's DNA! Finally, you will use some simple filtration and chemistry to cause the cells' genomic DNA to become visible to the naked eye — the blueprints of life!

THE GENETIC ENGINEER'S PLEDGE:

For the betterment of humanity, I pledge,
With all my DNA, cells, and knowledge,
To never use my genetic engineering mastery
To lay harm on the natural world or anybody.

BREAKING CELLS OPEN & EXTRACTING DNA

1. CREATE SALT WATER IN THE BAG

In a small resealable bag, mix together 1 tablespoon of distilled or bottled water with ¼ teaspoon of white salt. The distilled/bottled water will help you in creating a liquid slurry of mashed-up fruit, ultimately enabling you to create a suspension of separated fruit cells. It's important to use distilled or bottled water because tap water contains lots of salts and other impurities that could ruin the experiment (Figure A).

Julie Legault

2. MASH THE STRAWBERRY INTO INDIVIDUAL CELLS

You will now add one strawberry (or ½ kiwi or ⅕ banana) to the salt-water solution. Mash up the fruit by massaging it in the resealable bag until it is a smooth fruit slurry with no pieces or lumps (Figure B). (You can keep or remove the leaves, it is your choice.) At the end of this step, the fruit cells will have been separated from each other.

These individual cells are now ***suspended*** in the slurry. Most of the cells will still be intact and functioning, but some will have been torn open by the mixing.

The strawberry is made of millions of individual cells that are tightly packed together, each containing genomic DNA. By separating the cells as much as you can (Figure C), it will be easier for the soapy chemicals used in Step 3 to come into contact with each cell and cut them open.

3. BREAK OPEN THE CELLS WITH SOAP

Now we need to break the cells open, a process scientists called ***lysing***. This process will release the DNA from being contained in the cell.

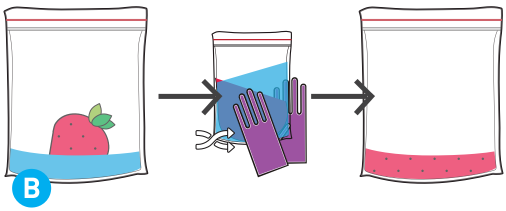
Add ¼ teaspoon of your shampoo/soap with EDTA, also called ***lysis buffer***, into the fruit slurry and mix for 3 minutes (Figure D). If Step 2 was not completed successfully and cells were left in clumps, lots of cells inside the clumps would be protected from coming into contact with the cutting power of the shampoo, resulting in less freed DNA. When the DNA has been released into the salt water environment, it remains dissolved.

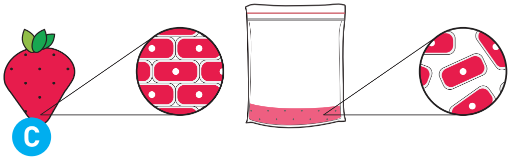
WHY THESE INGREDIENTS?

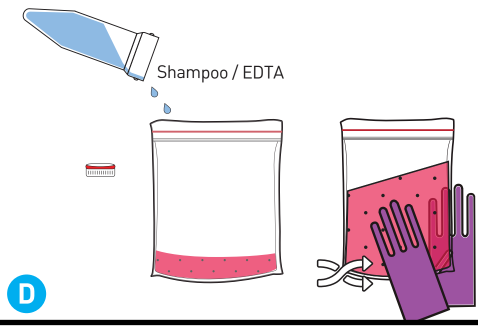
Why distilled or bottled water? Lots of calcite ($CaCO_3$) and dolomite ($CaMg(CO_3)_2$) are often found in tap water and calcium (Ca^{2+}) and magnesium (Mg^{2+}) ***ions*** — charged atoms — play an essential role in how DNA is regulated, driving molecular interactions, and even controlling biochemical reactions in cells. Moreover, tap water often contains chlorine and fluoride. Your DNA extraction will work better with distilled or bottled water which has reduced ions.

Why salt? Salt is added to help the DNA stay separate from the cellular machinery and macromolecules that make up the fruit slurry. When you break open the cells, there will be thousands of different molecules and ions floating together that will want to bind to one another in a jumble. Table salt, which is mostly sodium chloride (NaCl), binds to and creates ***buffers*** or shields around many of the molecules, including the DNA. Like the bumpers on a bumper car, the salt ions help keep the DNA free from binding with (sticking to) other molecules. This enables you to get a larger quantity of 'pure' DNA at the end of the exercise.

Why EDTA? This small molecule is really good at binding to positively charged metal ions like Ca^{2+} and Mg^{2+} — that's why it's added to shampoos. In a living cell, these ions act as a glue that binds DNA to proteins. If we can remove the metal ions, then we remove the proteins and get pure DNA!

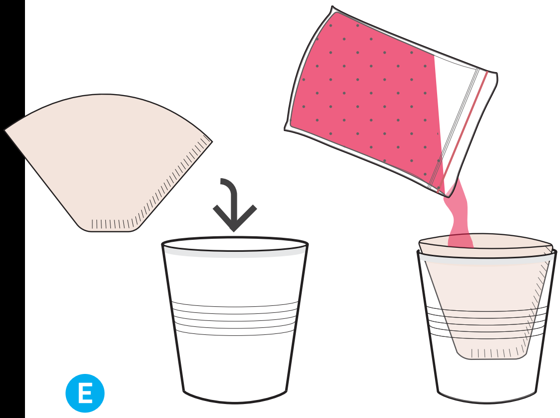




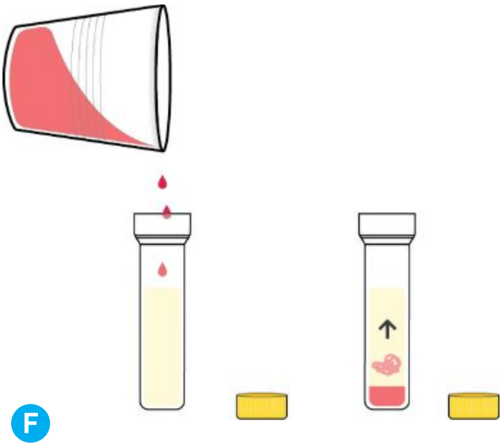


SURFACTANTS PRO-TIP

There are many different kinds of ***surfactants*** — detergents, dispersants, emulsifiers, etc. — used when doing genetic engineering experiments. In this exercise, you're using ***sodium laureth sulfate***, or ***SLES*** — a common surfactant in body soaps — to cut the cell membranes. In laboratories, a very common surfactant is called ***sodium lauryl sulfate (SLS)*** aka ***sodium dodecyl sulfate (SDS)***. Another popular surfactant is called ***Triton X-100***. In general, you can browse the internet to find which surfactant will work for your experiment. Different surfactants may work for the same experiment. For example, we could have used Triton X-100 for this exercise. We didn't because it's much more expensive and harder to get than body soap, which provides the same result in the fruit DNA extraction.



E



F

4. FILTER THE CELL DEBRIS

You now want to separate the DNA that is still dissolved in the salt water from the rest of the cell debris and ***micelles*** — spherical jumbles of surfactant molecules — that you are not interested in for this exercise. This debris includes the carbohydrates and lipids from the membrane of the cells, soapy micelles with captured cell debris, and clumps of proteins. Simple filtration will be sufficient to separate this debris and micelles from the dissolved DNA.

Set a paper coffee filter in a cup and pour your bag's contents into the filter (Figure E). Allow a few minutes for the salt water liquid containing DNA to pass through the filter. As the water, salt, and dissolved DNA pass through the filter, accompanied by some excess shampoo components, a small amount of proteins, and some color pigments, the fruit's remaining molecules will get caught in the filter. At the end of this step, you'll have dissolved DNA in slightly soapy salt water in a cup.

You will notice that the DNA liquid looks like red water. How do you know that DNA is actually in there? Let's make it visible!

5. PRECIPITATE THE DNA

In this last step, you are going to use chemistry to cause the DNA to "fall" out of the solution so that it can be seen with the naked eye. This ***precipitation*** is one of the most commonly used techniques in chemistry. Add 2 tablespoons of

Julie Legault

91% to 99% isopropyl alcohol to a narrow glass and set it in front of you. You're about to see some chemical wizardry!

If you look at the liquid that passed through the filter, which scientists call the ***filtrate***, you will notice it is clear and tinted red thanks to some red pigment from the strawberry that remained in the salt water. You cannot see any DNA, can you?

To make the DNA visible, pour a small amount of the filtrate (~½ teaspoon) containing the DNA into the isopropyl alcohol tube or glass. There is plenty of DNA in that volume of liquid. (Note that if you add too much, the precipitation may not work as well.) As you pour, you will begin to see the DNA precipitate out of solution. After a minute you'll see a white stringy glob of DNA. It may even end up floating to the surface (Figure F)!

ON COMPLETING THE FIRST EXPERIMENT!

Were you surprised at how much DNA you found in a single strawberry? You added only a small amount of DNA filtrate to the alcohol, and yet there was a lot of DNA. Now consider all the other food you eat. If the food was at some point living, then it will have DNA in it — you eat a lot of DNA on any given day!

Now that this experiment is complete, you won't be using your isolated DNA further. You can keep it or throw it away. Since there are no dangerous or living materials involved, you can flush the liquid part down the toilet or sink.

Remember that this experiment was significant for several reasons:

- It helped you begin your journey learning about cells and molecule bonding.
- You saw DNA for the first time!
- It helped you learn about how to break open cells.

Genetic engineers use this technique to isolate DNA from the environment to use in their projects. You'll learn more about this as you become a Genetic Engineering Hero! In the rest of the book, you will take a more in-depth look at the fundamentals of DNA, and how DNA relates to cell survival. You'll also be exploring how DNA can change naturally or through human intervention. This will set the stage for you to even start manipulating the genome of cells.

This project is excerpted from ***Zero to Genetic Engineering Hero*** by Justin Pahara and Julie Legault. Get your copy — and the companion kits with materials for doing all the experiments in the book — from Maker Shed at makershed.com/products/zero-to-genetic-engineering-hero.

New From Make: Projects

The latest cool stuff from makeprojects.com

Written by Dan Schneiderman

1 FUNCTIONAL SNOWSPEEDER FROM CARDBOARD

Vivian Thomas @thevivthomas

No need to be on Hoth to ride this snowspeeder. What started with a set of swivel wheels from the bottom of an old toolbox, scavenged cardboard, and a hodgepodge of recyclables, grew to a light-up, sit-inside pull toy fit for a rebel. makeprojects.com/project/functional-snow-speeder-from-cardboard

DAN SCHNEIDERMAN is a maker, space enthusiast, co-chair of Maker Faire Rochester, and Community Manager at Make:. Share your cool projects at makeprojects.com!

2 BMO: ADVENTURE TIME

Bob Herzberg @bob1524588514

Not only is this BMO fully functional with built-in voice recognition, the ability to play video games that were originally featured in the Cartoon Network show, and customized phrases thanks to the original voice actress, but it has even gone where no BMO has gone before: the International Space Station. makeprojects.com/project/bmo-adventure-time

3 CHICKEN ROBOT VOICE ASSISTANT

Mitchell Malpartida @mmalpartida

Have you ever wanted to have Heihei from *Moana* as your personal voice assistant? Now you can, thanks to a Raspberry Pi, Mycroft voice assistant software, and a few modifications to a previous robot sidekick project. Warning: this robot assistant only responds in "chicken speak." makeprojects.com/project/chicken-robot-voice-assistant

4 CHRISTMAS GNOME WITH CHRISTMAS TREE BRANCHES

JZ @JustMightDIY

With a little bit of holiday maker magic, you can adorn your front steps with this Christmas Gnome. Built from the extra branches from a Christmas tree, some extra mittens, and that tomato cage that no longer cages tomatoes, this little guy will make you feel at *gnome* for the holidays. makeprojects.com/project/holiday-show-tell-contest-make-a-christmas-gnome-with-christmas-tree-branches

5 DIWA ADVICE MACHINE

@DiWa

Out on a walk in Munich you may find this old U.S. Army Multimeter ME9 H/U deployed in the wild, dispensing advice. At the drop of a coin, an ESP8266 pulls info from various APIs via Wi-Fi and prints out tips, lucky lotto numbers, the current weather at your location, and even a joke on a thermal printer. makeprojects.com/project/diwa-advice-machine

6 WORLD'S LARGEST 3D-PRINTED RASPBERRY PI LEGO COMPUTER

Chris Bensen @chrisbensen

This Lego computer might not fit with the rest of your bricks. At 15x scale, the 3D-printed giant brick is big enough to fit a real Raspberry Pi 4 computer, battery, and touchscreen to make a working replica of its namesake. Ready to install in your real spaceship! makeprojects.com/project/worlds-largest-3d-printed-raspberry-pi-lego-computer

Redacted surveillance photo of FBC "Agent K." Note working green LED on shoulder pack.

Fabricate a Costume

Tips on building cosplay props from your favorite show or game

Written and photographed by Bob Knetzger

TIME REQUIRED: A Weekend

DIFFICULTY: Intermediate

COST: $40–$60

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.

Cosplayers are eager to get back to in-person video game events and comic cons. Here's a fun and easy costume my son Reed came up with, based on the video game *Control*: a Federal Bureau of Control agent. This costume also works great for a Zoom meeting as it's a "waist up" design!

Even if you're making a completely different costume, here are some basic fabrication tips and techniques I helped him with that you can use, too.

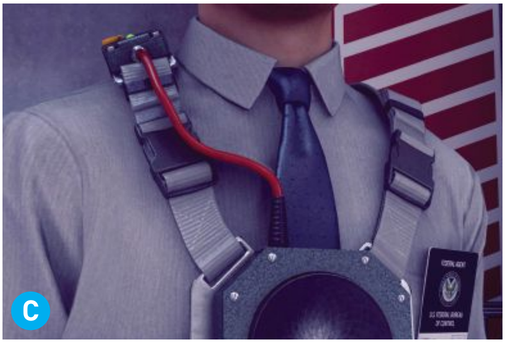
REFERENCE

For this costume there was plenty of good reference imagery, direct from the game and website. This basic design looks easy enough: web straps and buckles holding a breastplate and a shoulder strap-mounted module.

TIP: Find front, side, and top views so you can size things (Figures A, B, and C).


A


B


C

SOURCING

Buy versus build: Many parts can simply be purchased but to get it looking just right some elements will have to be custom fabricated.

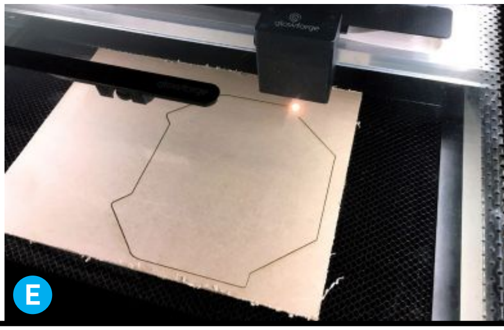
TIP: If you are going to buy components, you may find it is easiest to shop online, but some things are best sourced in person. You can quickly feel materials, see real colors, and check compatibility of sizes, like these straps and buckles gathered at the fabric store (Figure D).


D

FABRICATION

Weight, durability, and comfort are important in any costume. The two housings are hollow and made of light plastic with thicker, ¼" acrylic back panels for strength.

TIP: You could cut and drill the backing shapes and the strap slots, but that would also mean lots of handwork finishing all the edges. Laser cutting with the Glowforge (Figure E) gives a perfect "flame polished" edge to the acrylic parts — done!


E


F

DRY FITTING

These straps fit fine but the flat backing panel can be improved (Figure F).

TIP: Always check your work as you go, by trying the fit and function first before painting or finishing.


G

THERMOFORMING

To fit better, the acrylic shoulder piece is given a slight bend at each end. Remove the paper backing from both sides and clamp the acrylic where the bend will be. Carefully warm the plastic at the *outside* corner of the bend, using a heat gun (Figure G). When softened, bend and hold the plastic until cooled to make the new shape (Figure H).

TIP: The clamp protects the end part of the acrylic from bending or warping.

I vacuum-formed some styrene to make a hollow shoulder housing. To create a form, I cut and shaped some rigid foam and milled corner recesses for the screws (Figure I), then drilled holes for the LED and screws. The screws are just for appearance and don't do anything — they'll be glued on after the shell is painted (Figure J).

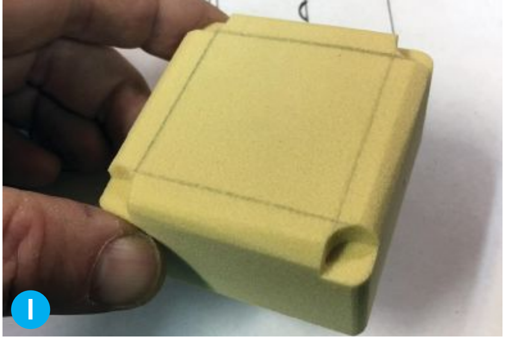
TIP: Vacuum forming is really great for costume making. You can make multiples fast and it's perfect for creating masks. For more on vacuum forming and bending plastic, check out my book *Make: Fun!* (makershed.com/products/make-fun) and build an easy Kitchen Floor Vacuum Former (makezine.com/projects/kitchen-floor-vacuum-former).


H

To make the breastplate housing, I fabricated the octagonal shape from glued-up styrene sheet (Figure K). Triangular gussets inside make it strong and light (Figure L).

The "lens" of the breastplate is a screw–on plastic jar lid with a disc of styrene glued onto the bottom (Figure M).

TIP: Be resourceful and find household items or trash that can be used for parts.



DETAILS AND FINISHING

The housings get a "hammered" finish paint, and the backing plates are painted silver. A green LED, battery, and switch go inside the shoulder housing, which is velcro-ed to the backing for easy access. The cosmetic screw heads are glued in place. The yellow knob is a dummy made from solid plastic parts shaped, painted, and then glued into place. The red cord and its black stress relief came from the junk parts bin and luckily were a perfect match!

Loop and sew the straps onto the slots in the backings. The rest is just assembly and final graphics to add the lens and badge.

Add your own shirt, pants, and black tie. The end result is picture perfect: Agent K is ready for the con!

1+2+3

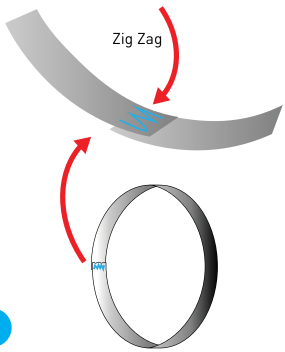
DIY Air Filter

Fight fire with filtration!

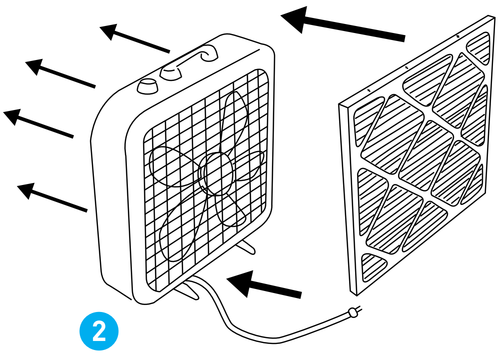
Written by Ace Makerspace

YOU WILL NEED:

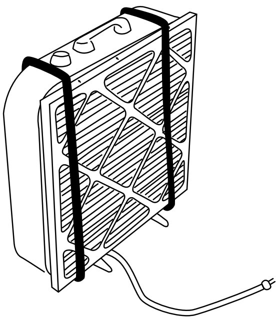
- **Box fan, 20"**
- **Air filter, MERV 13 grade, nominal size 20"×20"×1"** actual size 19½"×19½"×¾"
- **Elastic straps, 50% stretch, 1" wide, 40" long (2)** such as Amazon B07TBP5589
- **Sewing machine and thread**



1



2



3

Rachel "McCrafty" Sadd

Air getting nasty? Use this DIY air filter during a wildfire or other event when the Air Quality Index (AQI) is hazardous to create a room in your home with better air quality.

1 SEW THE STRAPS.

Loop the elastic and sew the ends together with a zig-zag stitch for strength.

2 PUT THE FILTER ON THE FAN.

Place the MERV 13 filter on the back of the fan. Make sure the arrow markings on the filter are pointed toward the fan.

3 STRAP IT DOWN.

Use the two elastic straps to hold the filter securely to the fan, one on each side. That's it!

TIPS FOR YOUR CLEAN AIR ROOM

Use painter's tape to seal around windows and door frames, and use a rolled-up towel or sheet at the bottom of doors. Inside your "clean air" room, don't smoke or use candles, stoves, aerosol sprays, or vacuum cleaners.

DIY KITS

Wildfire smoke has had serious impacts in California and the West for the past several years, with under-resourced folks being the most severely affected. The Ace Makerspace community has partnered with Homies Empowerment to distribute 500 DIY air filter kits in Oakland to help manage poor air quality indoors. Learn more or contribute at acemakerspace.org/500-filters.

ACE MAKERSPACE, formerly Ace Monster Toys, brings people together to create, learn, and solve life's challenges with an amazing array of tools, technology, and project-based education, all in inclusive spaces serving everybody in Oakland, California, and beyond.

Calling All Space Farmers!

The Growing Beyond Earth® Maker Challenge is kicking off Year 3: Robotic Planting + Harvesting

Fairchild Tropical Botanic Garden, in Miami, Florida, has partnered with Moonlighter Fab Lab, Nation of Makers, and researchers and engineers at NASA to improve food production technologies for long duration space travel. Together, we're calling on makers across America to submit new designs for gardening systems to be used aboard spacecraft. This year-long contest is open to high school, collegiate, and professional entries.

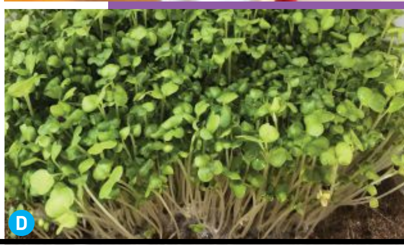
Help NASA grow plants in space!

Learn more at bit.ly/gbe-contest


A


B


C


D

YEAR 2 WINNERS

(A) HIGH SCHOOL WINNER | **Team C.A.M.I.**
Noah Kay, Noah Berlin, Chris Regini

(B) COLLEGIATE WINNER | **Team S.I.M.O.N.**
Sarah Humphrey, Emerick Larkin, Stephen Lantin, Seann Romero, Jacob Mass, Jacob York, Oren Anderson

(C) PROFESSIONAL WINNER | **Team Harvest Drawers**
Danielle + Karl Wagner

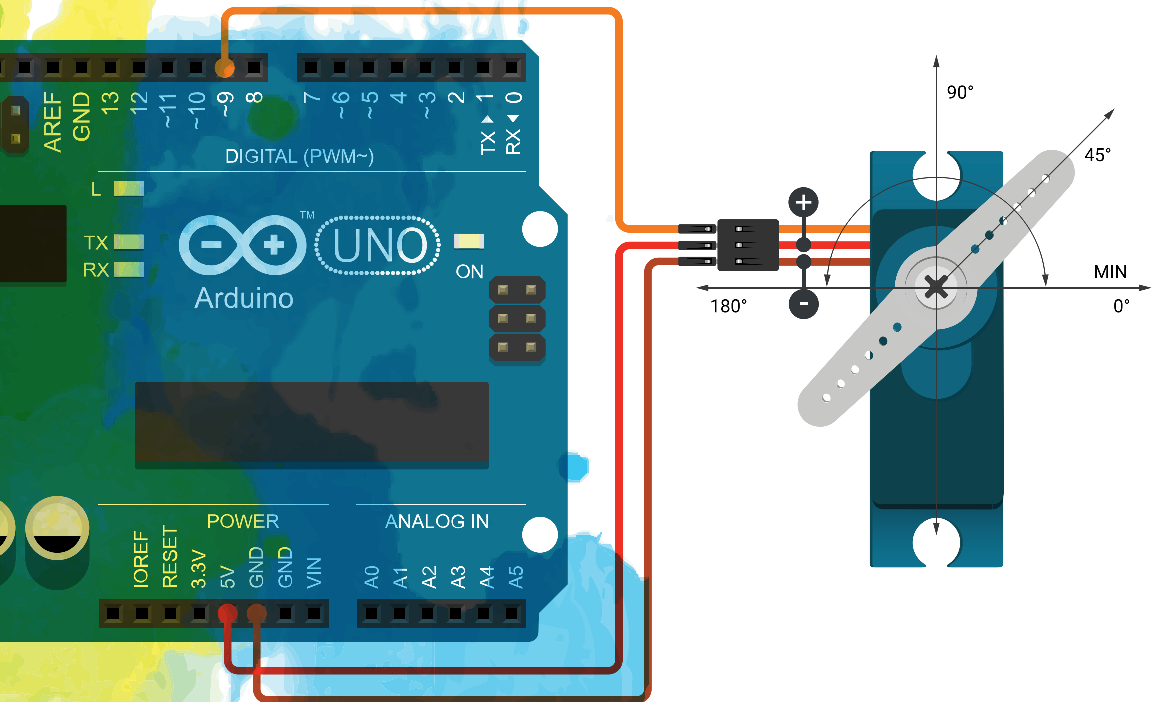
(D) DISTRUPTIVE TECHNOLOGY | **Team Microgreens**
Megan + Jeff Hughes

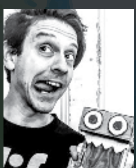
This project is based upon work supported by NASA under award No 80NSSC18K1225. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration.

MAKE A MOTOR MOVE!

The easiest way to animate your robotic creations

Written by Mario Marchese





MARIO "THE MAKER MAGICIAN" MARCHESE is recognized as an innovator integrating DIY robotics with homemade magic. He has appeared on *Sesame Street*, NBC's Universal Kids, and live on tour with David Blaine, who calls him "the best kids' magician in the world!" Mario is the author of *The Maker Magician's Handbook* and the new book *Robot Magic*. Learn more about him at mariothemagician.com.

Adobe Stock-Eva Kali and Ashish, Mario and Katie Marchese

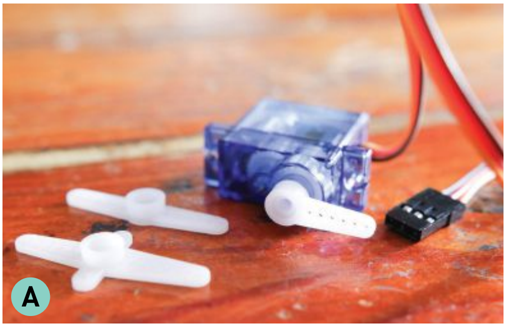
To create robotic movement, we need *motors!* But before we get into this, let's first remember that at the core, everything we're doing with the Arduino microcontroller is just about switching things on and off, off and on. Whether it's a small LED that blinks, or a relay that allows us to power something bigger, it all comes back to simply turning things on and off. Working with the Arduino is just about commanding a bunch of digital pins to go on and off. Keep this in mind, and it's much less overwhelming!

What's the easiest way to make a motor move with the Arduino? My favorite motor to use is a ***micro servomotor***. They are cheap, accessible, and very easy to code. The best part is that they work well within the 5V that the Arduino's digital pins create. This is huge! This means you can separate the board from your computer and power both the Arduino and the micro servo with just a 9V battery! Think about how portable that is.

I built one project from a small coffee canister and lid. Every time I looked away, the lid would pop open and two eyeballs would appear. Then, just as fast, the eyeballs would disappear back into the can and the lid would close. Every time I looked away, it would happen again. It was so simple, but it elicited great reactions when I performed with it. That project was created with just a single micro servo and an Arduino, powered with a 9V battery. This is just one small example, but believe it or not, the foundation of almost all of my robot magic comes down to programming a micro servomotor with an Arduino. We will start with one motor and work up to programming multiple motors and understanding how to power them all. Through this process, we will learn the fundamentals to *animate what we make!*

WHAT IS A MICRO SERVO?

Don't be fooled by its size ... this little thumb-sized motor can accomplish a lot. A micro servo has a small arm that can move from 0–180 degrees (Figure A). With the Arduino, you can program that arm to move to a particular position, have it stay in that position, and move again whenever you want. The servo arm has small holes in it. This is so you can attach things to it.



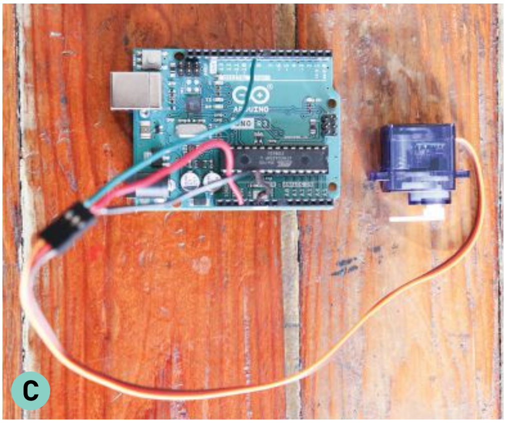
A

B

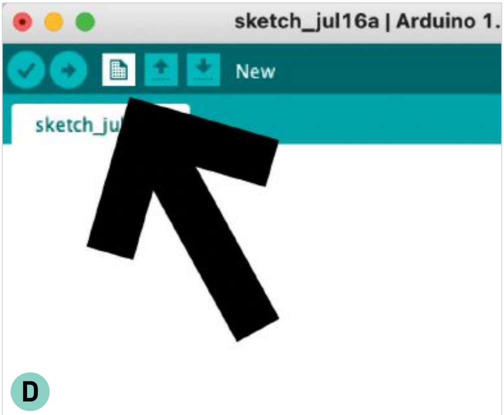
There are thousands of applications for servomotors, and infinity ways to adapt them for performing magic and comedy. One Halloween, I built a servo-powered silly string shooter (instructables.com/Arduino-controlled-Silly-String-shooter) and hid it inside a carved pumpkin. That's right, we made a puking pumpkin silly string shooter! All with a servo and an Arduino. Trick-or-treaters had a blast!

Inside a servo, we have four main components: a ***gearbox***, a ***potentiometer***, a ***DC motor***, and a ***control circuit***. The DC motor inside is very small and can power up at a high speed but is fairly low on torque, meaning it spins really fast but can't lift much. This is where the gearbox comes in. Just as the name describes, the gearbox inside is made up of little gears (Figure B). The spinning gears take the motor's speed and turn it into higher torque. Think of a bicycle going up a hill. At a low gear, you are able to pedal faster, putting less pressure on the pedals to get to the top of the hill easier. It's just like that with this.

The main shaft comes out of the gearbox and sticks out of the servo's plastic housing. That main shaft is where we attach the servo arm. On the underside of the shaft lies the potentiometer. A potentiometer is a ***variable resistor***. It acts as



C



D

```
#include <Servo.h>

void setup(){

Servo servo;              // Change the name from Servo
                          // to whatever you want!

servo.attach(8);          // Here we tell the Arduino
                          // which pin "servo" is
                          // attached to

}

void loop() {

delay(1000);              // Delay just means WAIT! Remember
                          // every 1000 = 1 second

servo.write(180);         // move servo to 180 degrees

delay(1000);              // Wait 1 second

servo.write(0);           // Move servo to 0 degrees

delay(1000);

}
```

E

Mario and Katie Marchese

a feedback sensor and also helps the servo stay in the position we tell it to. Try gently moving a servo arm that is programmed to stay in a certain position. It will fight you! It will always push or pull back to the position it's programmed to stay in.

But how does the servo arm know where to go? And how does it know how long to stay there? That's where the control circuit and Arduino come in. The control circuit inside the servo reads a ***PWM (pulse width modulation)*** signal from the Arduino, and with this communication, all the various components of the servo work together with the Arduino to turn on a motor, control the speed, stop it to an exact position, and hold it at that position until it's told to do otherwise. It's pretty amazing, really!

CONNECTING THE SERVO TO THE ARDUINO

You'll see three wires coming out of your servomotor. Let's connect them to the Arduino using ***male-to-male jumper wires***. Plug the red wire to positive, 3.3V or 5V on the board. Plug the black wire to GND on the board, and plug the orange/yellow wire to digital pin 8 on the board (Figure C).

Now that the servo is connected to the board, let's connect the board to your computer via USB. Then, open up the Arduino application and open a new ***sketch*** (an Arduino program) by pressing the square button that looks like a piece of paper (Figure D).

Replace the code in the window with the code shown in Figure E (also found at mariothemagician.com/robotmagicchapter3).

Now go to Tools → Board and select Arduino/ Genuino Uno. Then, click Tools → Port and select the correlating port (usually the last one listed.) Press the Compile button (the circular checkmark button.) When it's done compiling, press Upload (the circular arrow button.) As always, if you receive an error code, double-check your sketch! The code must be copied exactly.

UNDERSTANDING THE CODE

Let's break it down, starting with **`#include <Servo.h>`** — we must include the *Servo.h* library here for the board to communicate with the servomotor.

Next, you'll see **`Servo servo; //Change the name from «servo» to whatever you want!`** As the note within the code suggests — remember, text that comes after the **`//`** marks is a *comment* for your reference only and will not be read by the Arduino — this is where we name our servo. Call it whatever you want! Say, you'd like to name your servo Meatball. That's fine! Just make sure you replace **`servo`** with **`Meatball`** in all parts of the code.

Next up, we see **`servo.attach(8); //Here we tell the Arduino which pin # «servo» is attached to`** — we need to make sure the orange/yellow servo wire is connected to the same pin referenced in the code. Here, it's pin #8. If you have the wire plugged into a different digital pin, either change the number in the code to match, or move the wire to pin #8.

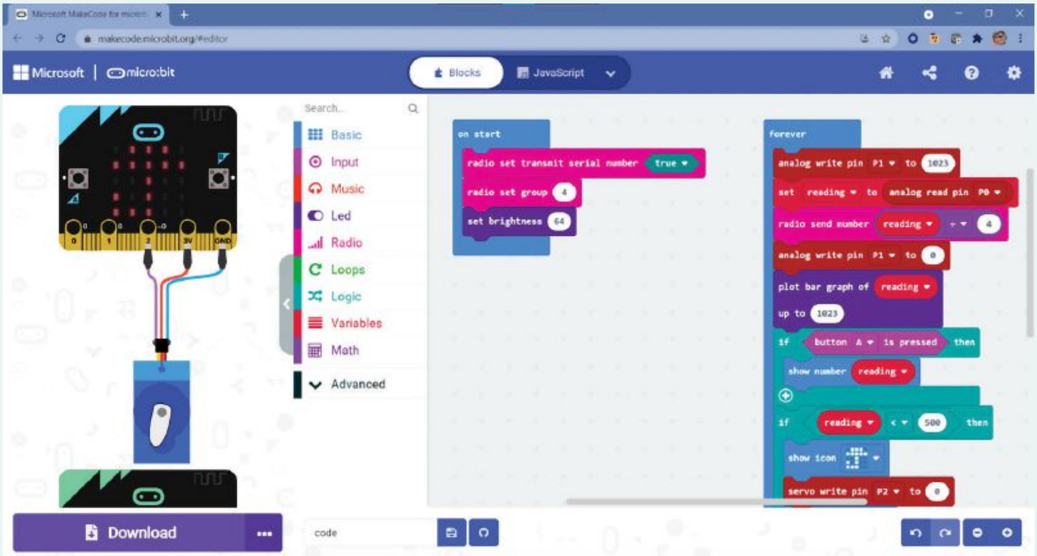
Next we see **`delay(1000);`** — this tells our routine to wait! **`1000`** equals one second.

In the next line, **`servo.write(180); // Move servo to 180 degrees!`**, the command **`servo.write(180);`** tells the servo arm to move to 180 degrees. Change this number to change the position the arm lands on. If it said **`servo.write(78);`** that would move the servo arm to 78 degrees, for example.

And then **`delay(1000);`** tells it to wait 1 second again. Do you know that between those two commands, you hold all you need to animate what you create? That's it! Once you understand these two commands, you can create all kinds of amazing magic with servomotors. You can bounce between multiple servos, too! This is a simple formula that will weed out all of the confusion that comes with coding. Again, take some time to play, attach things to the servo arm and see for yourself what the motions look like and how changing the numbers in the code affects the movements.

I use **`servo.write();`** for all my coding. There are other ways to make a servo move with code! A simple Google search will show you that. But this is the way I love doing it. In my stage show, each of my robotic elements perform several-minute-long routines in front of live audiences. Through a lot of trial and error in creating all those various routines, the thing that became most apparent is that creating a clear storyline with code is the most important thing. Overly complicated code can easily overwhelm us and prevent us from making edits and tweaks and going the extra mile to hone our routine to perfection. Having an easy-to-follow storyline in your code, through which you can program movements piece by piece, is key. I have found that using **`servo.write();`** is the simplest way, and that will become more clear as we work through the projects featured in my book. The goal of our code is to replicate human emotion with robotic movements. Those movements can demand a lot of code, sometimes 16–17 lines for just one subtle gesture. Keeping it all confined to a simple **`servo.write();`** and **`delay();`** will help you create those complicated movements with ease.

This tutorial is excerpted from the new book *Robot Magic: Beginner Robotics for the Maker and Magician*, available at makershed.com. It shows how to program motors, relays, R/C transmitters, and more, to create 7 different robot magic tricks, including the Chomper Bot (kit available at makershed.com/products/the-chomper-bot).



A The BBC micro:bit Plant Watering project in MakeCode.

BLOCK-BASED PROGRAMMING

Don't write code? Try drag-and-drop, snap-together, *visual programming languages*

Written by Brian Jepson

BRIAN JEPSON is a content manager at LinkedIn Learning, co-organizer of Providence Geeks, founding member of the National Maker Faire planning and production team, and co-producer of the Rhode Island Mini Maker Faire (makerfaireri.com).

There must be something fundamental that makes building with blocks so universally compelling. Toy blocks and bricks have been with us for hundreds of years, if not more. When kids grow out of alphabet blocks, they graduate to Lego or Duplo bricks, or some other kind of construction toy. (For me, it was Lincoln Logs — I didn't get into Lego until I was older.) Today, there are many construction experiences that take advantage of modern technology, from electronic blocks like littleBits to sandbox construction games like *Minecraft* and *Fortnite* Creative mode.

Early learning is catalyzed by the simplicity and tactile satisfaction of snapping things together, and can create an appetite for similar experiences later in life. Even if you don't opt for a career assembling things from modular components, the block paradigm may prove inescapable. Computer programmers who use object-oriented or object-based programming methods are no strangers to encapsulating code and data into modules that can be snapped together like so many blocks.

Even when programmers are working in a modular way, the blocks are merely conceptual. In most cases, they write code in text, connecting their blocks by typing out the name of the module and any data they want to send to that module. Yet, many programmers will use visual diagramming tools to plan and to document the code that they write as so many strings of text. The logical extension of this is ***block-based programming*** — creating code entirely by using visual, snap-together code blocks, without actually writing code at all.

COLOR-BLOCK CODING

There have been many successful attempts to turn programming into an entirely visual process. These approaches have largely been siloed: one approach for early education, another for architectural design, yet another for industrial engineering, and another for music and entertainment.

But in recent years, a colorful, drag-and-drop style of ***visual programming language (VPL)*** has broken out of its silo and been applied to microcontrollers (Figure A), to 3D design, the Internet of Things (IoT), and even mobile programming. This approach traces its roots to a decades-long initiative that employs block-based construction kits (both physical and virtual) to help young people learn computer science, engineering, and even user experience (UX) concepts.

This style of block-based coding grew out of the wildly popular **Scratch** programming language. Scratch itself is rooted in a community that has been working out ways to teach computing concepts to kids since the early days of computing. It really began in 1966 with the development of the **Logo** programming language by Seymour Papert at MIT and Wally Feurzeig, Daniel Bobrow, and Cynthia Solomon at Bolt, Beranek, and Newman (BBN).

TURTLES IN MOTION

Logo itself was not a block-based visual language. However, there are many concepts in Logo that made their way into Scratch. The most enduring of these is the idea of turtle graphics, in which an onscreen sprite was made to move in various directions on the screen. In fact, an introductory Scratch project today looks a lot like an introductory Logo program from 40 years ago, as shown in Figure B. In Logo, you control an onscreen "turtle" which drags a virtual pen and draws lines wherever it goes. In Scratch, you control an onscreen sprite (a cat), who can also drag a virtual pen and draw lines.

Shortly after the first Logo implementation, the virtual turtle was joined by a robotic turtle that crawled on a floor as the Logo code was executed. In the 1980s, Papert and Lego Group began a collaboration that would expand the definition of a turtle to anything you can make with Lego, and would eventually lead to the Lego Mindstorms robotics kit.

LOGO, MEET LEGO

One of Papert's students, Mitchel Resnick, working with Stephen Ocko, integrated Lego and Logo in a way that was reminiscent of the early turtles. **Lego TC (Technic Control) Logo**,

Logo and Scratch doing essentially the same thing nearly 40 years apart.

or **Lego/Logo**, was a combination of software and hardware that allowed students to create Lego robots running under the control of a Logo interpreter running on an Apple II computer or IBM PC. This was a tethered affair, in that you had to install an interface card that connected to a Technic Control Center module, which would control the Lego motors.

In the late 1980s and into the mid-90s, working under the guidance of Papert and now-Professor Resnick, Fred Martin, Randy Sargent, and Brian Silverman developed the Programmable Brick which removed the need for students' Lego creations to be tethered to a computer.

(As a Rhode Islander, I was proud to learn that in 1994 the Programmable Brick was used by teachers outside of an MIT lab for the first time, at a workshop held by the Rhode Island School of the Future, led by a friend of mine, Janice Kowalczyk!)

Resnick would go on to succeed Papert in the Lego Papert Professorship of Learning Research at the MIT Media Lab, and the Programmable Brick research led to the development of Lego's Robotics Invention System at the heart of their **Lego Mindstorms** educational product.

BLOCKY BEGINNINGS

In 1996, one of Resnick's undergraduate students, Andrew Begel, developed the **Logo Blocks** visual programming environment, which would bundle Logo code into visual blocks that students could drag around and connect to create behaviors. In Begel's writeup of the project (andrewbegel.com/mit/begel-aup.html), he asked a very important question: "Is it more fun for kids to program graphically?"

Logo Blocks was by no means the first graphical programming language, nor was it the first graphical programming environment for kids (Alexander Repenning's **AgentSheets** is likely to hold that honor). However, the development of Logo Blocks took place in an environment (the MIT Media Lab's Lifelong Kindergarten Group) where a massive amount of experimentation and innovation was focused on engaging kids in creative learning. So there was much, much more to come.

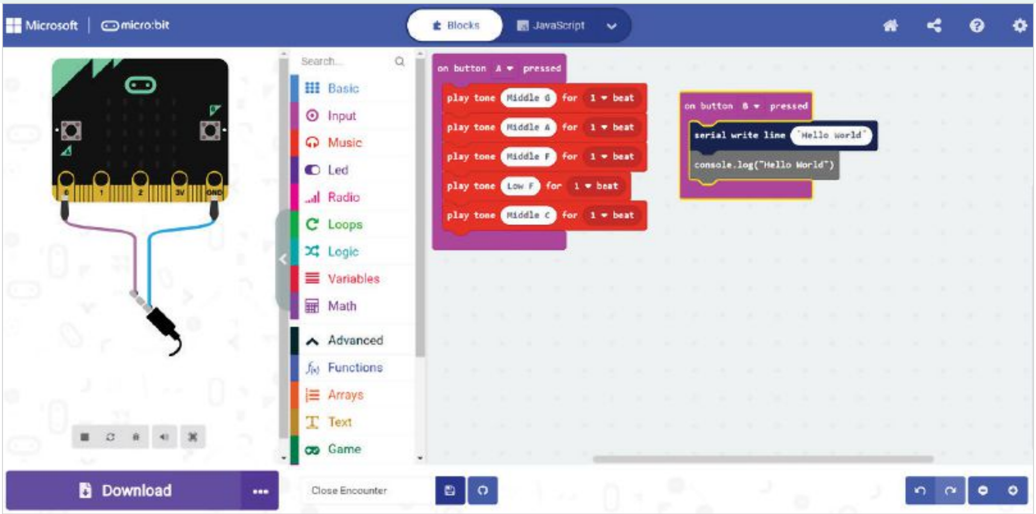
SCRATCHING AN ITCH

Starting in 2003, Resnick, working with John Maeda (MIT) and Yasmin Kafai (UCLA), took all the lessons from years of Lego/Logo work, inspiration from other visual programming languages, and with funding from the National Science Foundation, developed the visual programming language **Scratch**, which itself has inspired so many other educational languages. Since its initial release in 2007 as a desktop application, Scratch has evolved into a web-based JavaScript application.

The current, web-based version of Scratch is itself built on **Blockly**, a JavaScript library for adding block-based coding to any application. Blockly is the result of a collaboration between MIT and Google, and came out of the App Inventor project (another MIT/Google collaboration for using block-based programming to create Android apps). In addition to powering Scratch, Blockly is widely used in other visual programming platforms, including Microsoft's **MakeCode**, Code.org's introductory programming curriculum, and programming environments for various robotic kits and toys.

MakeCode itself is an open source platform for bringing block-based programming to all kinds of devices and environments. The BBC micro:bit uses MakeCode to power its programming environment. You write your code using either the Blockly-based programming environment or JavaScript. MakeCode allows you to switch between visual programming or a text-based programming language on the fly. It's not limited to JavaScript. You can switch the text-based editor to Python as well. What's more, so long as you are writing valid code, MakeCode will convert between JavaScript, Python, or visual blocks. If you write Python or JavaScript code that doesn't have a representation in the block-based syntax, MakeCode will show a `<python code>` or JavaScript block to represent it.

MakeCode will also simulate your target device to whatever extent possible. So if you add a code block that plays a tone, MakeCode will update the image of the micro:bit to include a wiring diagram for a headphone connection. And if you have an action associated with a button press



C Making music with MakeCode and micro:bit.

(Figure **C**), you can click that button in MakeCode to trigger it.

There are many other block-based programming environments for makers. For example, **Ardublock**, for programming Arduino-compatible microcontrollers (Figure **D**), is based on a customized version of **OpenBlocks**, the visual programming framework originally used in App Inventor.

BlocksCAD is a block-based 3D modeling environment based on Blockly that uses OpenSCAD (a text-based programming language for 3D modeling) as its 3D modeling engine (Figure **E** on the following page).

And coming full circle, there are variants of MakeCode that support programming with Lego Mindstorms and even with *Minecraft*.

VISUAL PROGRAMMING IN ART AND ENGINEERING

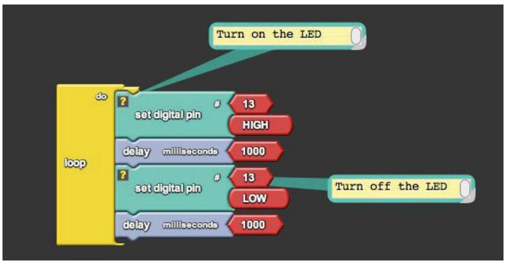
Visual programming is also well-established in professional disciplines, from music to engineering. Once you step outside of the realm of Scratch and Blockly, visual programming languages get a little more siloed. The languages all look quite different from one another, and function quite differently.

Max (also known as Max/MSP/Jitter) and **Pure Data** are two closely related visual programming tools. Both are powerful environments for creating music-centered

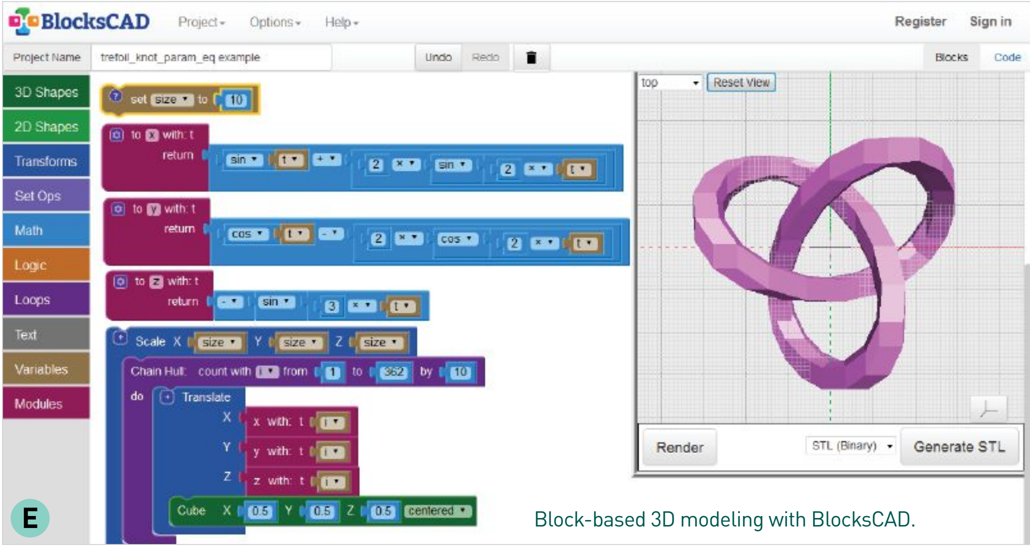
VOICE-CONTROLLED ROBOT

HomeMadeGarbage's Voice Controlled Robot Car on Hackster.io uses not one but two visual programming environments to create a robot vehicle that you control using Amazon Alexa. It uses MakeCode to program a micro:bit to control the car and Node-RED (a visual programming language for the Internet of Things) on a Raspberry Pi Zero to communicate with the Alexa API.

hackster.io/H0meMadeGarbage/voice-controlled-robot-car-54faef



D Snap-together Arduino code in Ardublock (example from *Make: Getting Started with Arduino* book.)



Block-based 3D modeling with BlocksCAD.

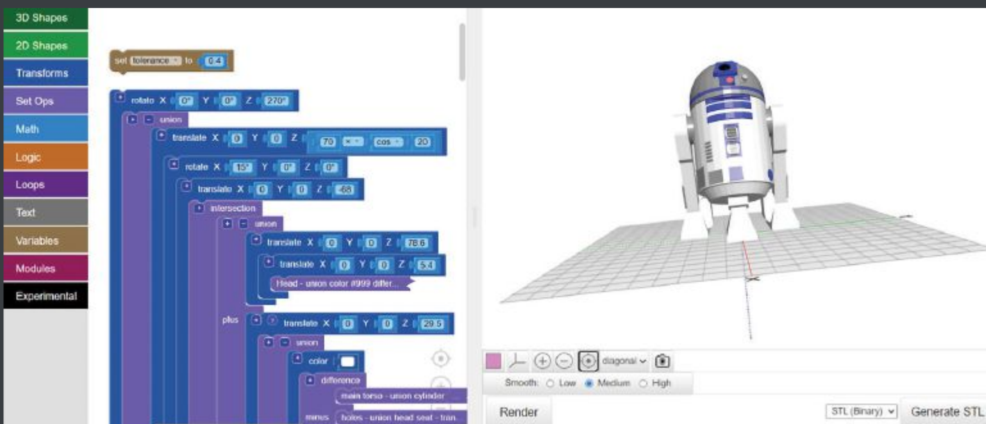
interactive experiences. Max is a proprietary, commercial DSP (digital signal processing) product created by Miller Puckette and now maintained and published by David Zicarelli's company, Cycling '74. Pure Data is an open source package, also created by Puckette. Both are quite similar, and use a patch metaphor for programming. Chunks of code are called ***patches***, and you link the patches together to create new behaviors (Figure F).

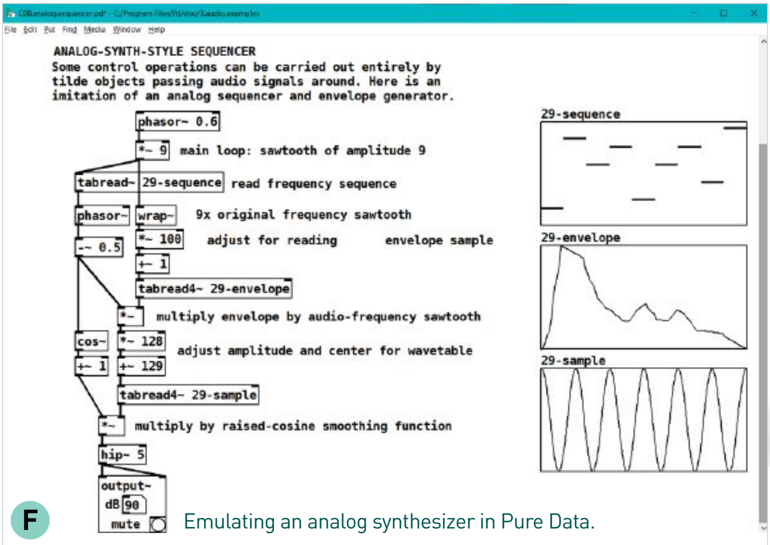
One of the benefits of advanced manufacturing, such as 3D printing or CNC machining, is that products and structures can be customized in amazing ways. Repeated patterns can follow the contours of a curved surface in a consumer product or even on a building. Structures can borrow ideas from organic shapes, creating interesting and compelling designs. But these sorts of structures aren't easy for industrial designers or architects to compose in CAD software. This is where visual programming environments like **Grasshopper** (a visual

R2-D2 MODEL

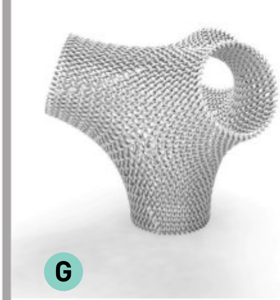
BlocksCAD user <aaron> has been working on an R2-D2 model for four years in a project that's been remixed more than a dozen times. This incredibly detailed droid involves some pretty complicated blocks. It's a great example of the kinds of complex creations that visual programming supports.

blockscad3d.com/community/projects/43443

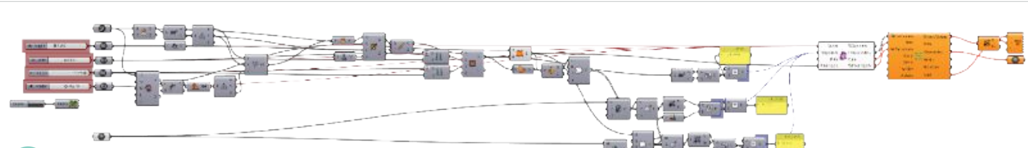




F Emulating an analog synthesizer in Pure Data.



G Woven node design created using Grasshopper (courtesy Walter Zesk).

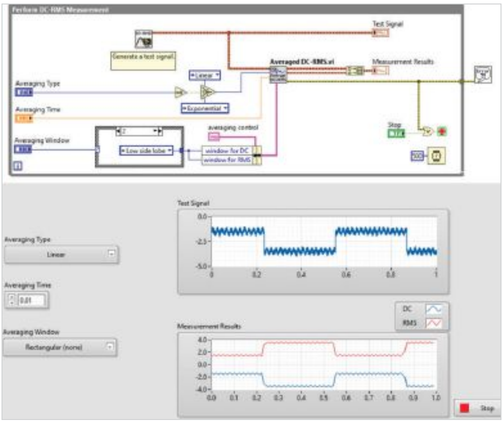


H Grasshopper script for the Woven node (courtesy Walter Zesk).

programming environment that's part of Rhino 3D) and **Dynamo** (a visual programming language that works with Autodesk software such as Revit) come in. Using a node-based programming environment, designers can manipulate geometry in ways that would be impractical to do manually or with traditional scripting environments. Grasshopper and Dynamo open up opportunities for enhancing models with a generative, algorithmic approach. Figures **G** and **H** show a Grasshopper design from Walter Zesk of Conform Lab LLC, a computational design consultancy.

LabVIEW (Figure **I**) is a visual programming environment from National Instruments used in industrial control. It allows you to create systems for test, measurement, and control. It will work with hardware from National Instruments as well as many other kinds of sensing and control hardware, including Arduino and Raspberry Pi. For a time, the Lego-supplied Mindstorms development environment was based on LabVIEW (the latest Mindstorms is based on Scratch).

Visual programming languages have been with us since at least the 1960s. Although the best-known and most recognizable of their ilk are the Scratch and Scratch-inspired languages that appeal to makers, educators, and kids, visual programming can be found in nearly every discipline. It's tempting to write off visual languages as a toy, something kids will grow out of, but visual programming is something that can stay with you well beyond your youth!



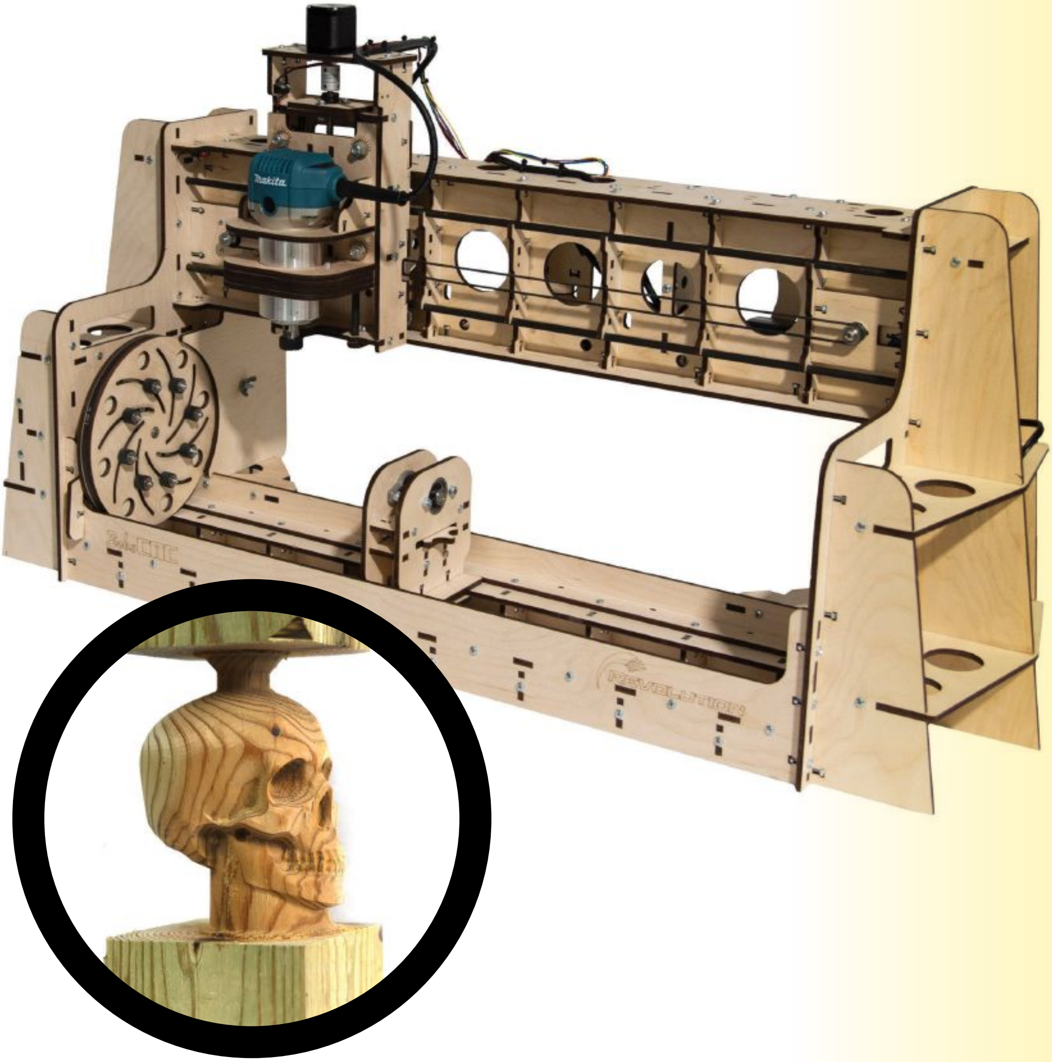
I Waveform measurement in LabVIEW.

TOOLBOX

GADGETS AND GEAR FOR MAKERS

Tell us about your faves: editor@makezine.com

Written by Caleb Kraft and Mike Senese



Bob's Revolution Rotary Mill

$1,130 bobscnc.com

Bob's CNC is back with another interesting purpose-built CNC router. This group produces kits that are laser-cut wooden frames, allowing them to be cheap and easy to construct. Their new kit is a rotary CNC router, meaning that it spins the material on one axis, while the cutting head moves around to cut into it. While this might sound like a lathe, it is more like a dedicated 4th axis, meaning that it can carve a full 360 degrees around a piece of material. Usually this type of thing is an upgrade to an existing CNC router, but in this case you can purchase it as a stand-alone device.

See an in-depth look and video of it in action at youtu.be/3zOas0xITTM

Wallwerx Organizer System

$9.99 wallwerx.com

Pegboard organizing systems can really be life savers when it comes to keeping your workbench as clutter-free as possible. Wallwerx has come up with an innovative jar-based solution. While they may not be the first to use the concept of a jar (I recall seeing baby food jars nailed through the lid onto a board in my grandpa's shop), they've considerably improved the usability and durability of the concept.

Instead of a flimsy peg grafted onto a cheaply cast lid, as is commonly found, the Wallwerx system has a clamp that fastens onto the peg board, allowing you to remove the whole jar, lid included. This is great, as I often need to grab a full jar of screws and take them to the job; the cheaper systems would mean I did this sans-lid.

Hear an interview with the creator of Wallwerx in our podcast, the Make:Cast makezine.com/2021/04/28/everything-in-its-own-jar

Sizzix Big Shot Pro Die Cutter

$310 amzn.to/3uHz7ol

When I got on a kick for intaglio printmaking, I found myself surprised by the cost of an etching press. A friend of mine mentioned that they had seen die cutters used as presses, so I looked into it. They were right — turns out buying a die cutter is a viable solution to getting a press on a budget.

I picked up the Sizzix Big Shot Pro, as it was large enough to run an entire letter-sized sheet (and then some!) through. It was cheaper than a comparable etching press, and could double as a die cutter should I choose to explore that. The only downside is that you can't adjust the clamping pressure, you just have to stack more material if you want more pressure.

20-Piece Cleaning-Brush Drill Attachment Kit

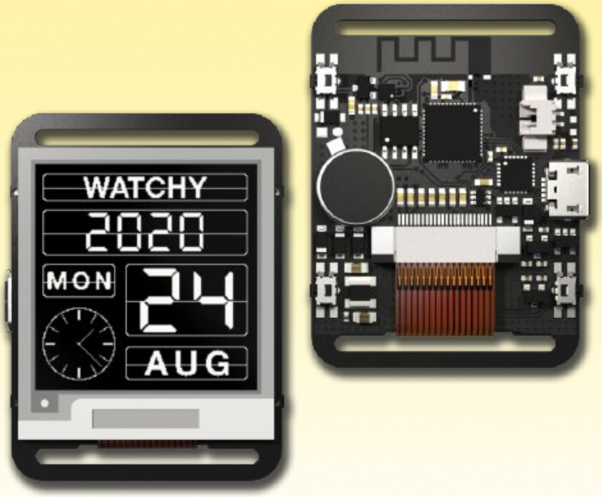
$17 amzn.to/2Wv1okY

Makes shower and tub cleanup a lot easier; useful for carpet and automotive too. Comes with enough variety for most jobs we can think of. Extension adapter is a little cheap, however.

Watchy — Open Source, E-Paper Wristwatch

$59 sqfmi.com

Build and program your own smartwatch with this tidy kit. ESP32-powered and Arduino compatible, too.



BESTSELLING Make: ELECTRONICS BY CHARLES PLATT NOW IN THIRD EDITION

More than 250,000 SOLD!

Back in 2009, Charles Platt found that most of the existing books on electronics were not hands-on guides. "I saw a big opportunity to do things differently and innovate," says Platt, who created *Make: Electronics*. He has now revised the book for a Third Edition.

"Other books tell you a theory, and then — maybe — you can test it experimentally," he says. "It's much more interesting, and memorable, to put some components together to *find out* what will happen, and reach a conclusion." Platt's book tells the reader how to put some resistors together to discover the relationship between voltage, current, and resistance — Ohm's Law. "Georg Ohm discovered it, but a newcomer can still RE-discover it!"

In electronics "the fundamentals don't change," Platt says. "Everything is still made out of transistors; the only difference since 1975 is that the transistors have become smaller! The information will still be important as a foundation for understanding electronics 10 or 20 years from now."

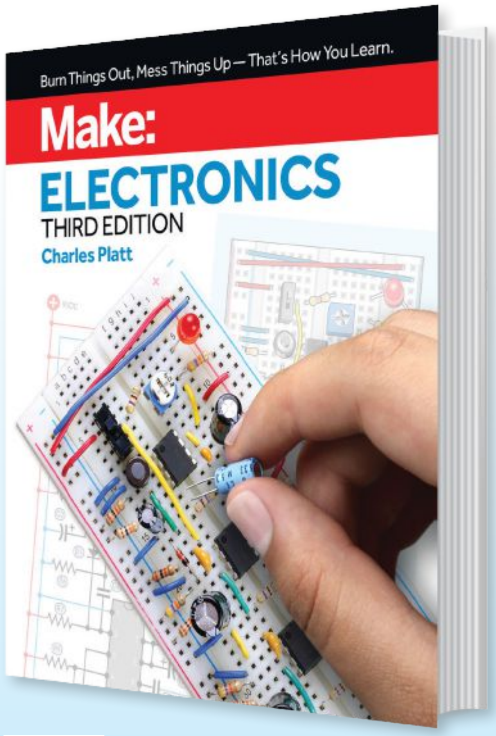
Platt notes that improvement is always possible, and that the new Third Edition is the product of his own learning processes, especially interacting with readers.

NEW IN THE THIRD EDITION, YOU'LL FIND:

- 80% new or rewritten text
- More than 100 all-new, more realistic color diagrams to guide the reader in building and understanding circuits
- All schematics now use the same orientation as components on a breadboard
- Full revisions of various circuits
- New, expanded chapters on microcontrollers

Make: Electronics provides a broad path into the topic, starting with just a 9V battery and ending up with logic diagrams. The Third Edition has more than 330 pages and almost 300 photographs and diagrams. Platt continues to make everything as simple as possible. "I wanted to reduce the necessary number of parts, so that people on a limited budget could still do the hands-on work," he says.

Make: Electronics Third Edition is the ideal starting point for anyone wanting to get started with the subject. Platt summarizes it succinctly: "Books are still the best way for you to learn electronics."



Make: Electronics Third Edition
By Charles Platt
ISBN: 978-1-68045-712-4

TOOLBOX

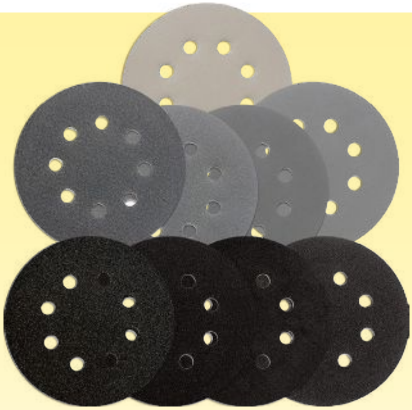
GADGETS AND GEAR FOR MAKERS

Tell us about your faves: editor@makezine.com

USB Endoscope

$20 amzn.to/3kqoP7E

Skinny, 5.5mm-wide camera with embedded LEDs fits into tiny spaces — peek inside walls, under appliances, into engine cylinders and ear canals. 720p resolution is adequate for rough inspection work.



5" Sanding Disc Assortment Pack

$16 amzn.to/2WwZ7WS

It's surprisingly hard to find a full range of grits in one packet. From 80 to 3000, this 54-piece bundle covers almost everything we need.

Aukey PowerZeus 500 Power Station

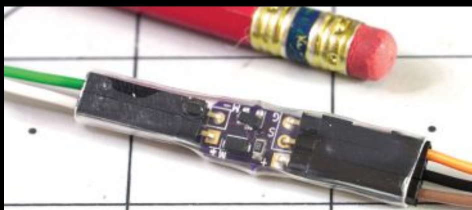
$459 aukey.com/products/powerzeus-500-portable-power-station-518wh

28,000Ah capacity, 500W max output, and sockets for most plug types. Useful for powering lights and doing small repair work in the field. In our testing, one hour of 100W AC continuous output consumed about 25% of its charge.

Large Reusable Lint Rollers

$14 FOR TWO-PACK amzn.to/2U05Zyi

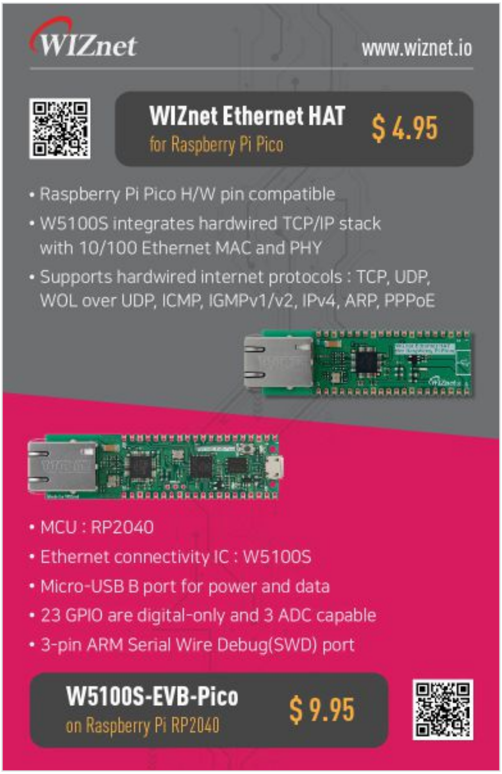
These things are a little flimsy, but otherwise work great, and can be reused over and over. For best performance, simply clean with water and a drop of liquid soap, then air-dry.



EZ Fan2 Tiny Raspberry Pi Fan Controller

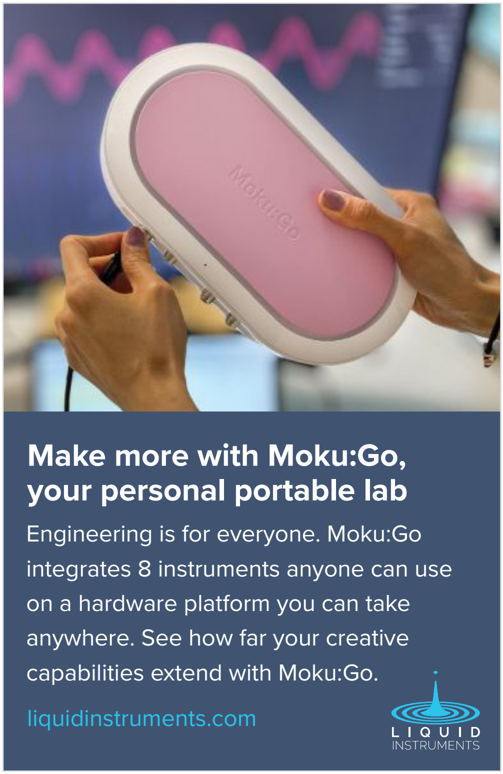
$8 tindie.com/products/jeremycook/ez-fan2-tiny-raspberry-pi-fan-controller

This sleek in-line device lets you set up and control a small DC fan (or other low-amperage motor) with your Raspberry Pi or Arduino.


WIZnet
www.wiznet.io
WIZnet Ethernet HAT
for Raspberry Pi Pico
$ 4.95
• Raspberry Pi Pico H/W pin compatible
• W5100S integrates hardwired TCP/IP stack with 10/100 Ethernet MAC and PHY
• Supports hardwired internet protocols : TCP, UDP, WOL over UDP, ICMP, IGMPv1/v2, IPv4, ARP, PPPoE
• MCU : RP2040
• Ethernet connectivity IC : W5100S
• Micro-USB B port for power and data
• 23 GPIO are digital-only and 3 ADC capable
• 3-pin ARM Serial Wire Debug(SWD) port
W5100S-EVB-Pico
on Raspberry Pi RP2040
$ 9.95


steamoji™
HELP TRAIN THE NEXT GENERATION OF INNOVATORS!
Franchise a maker academy for kids and bring our 400-hour curriculum to your community.
steamojifranchise.com


The World's, best value, easiest to use desktop router platform is now available a fantastic price.
Our desktop CNC machines are designed specifically for use by hobbyists, small business, and STEM Programs.
Mill RIGHT
CNC Creativity and Value
Our machines have long been recognized as an industry leader in versatile,easy-to-use synergistic CNC machines and software.
Contact:
1-229-439-6569
Office Hours 9AM-4PM EST M-F
CUSTOMER SUPPORT
sales@millrightcnc.com
Here's a 10% off coupon on your order. Use Coupon Code MAKE
(Offer expires 12/31/2021)


Make more with Moku:Go, your personal portable lab
Engineering is for everyone. Moku:Go integrates 8 instruments anyone can use on a hardware platform you can take anywhere. See how far your creative capabilities extend with Moku:Go.
liquidinstruments.com
LIQUID INSTRUMENTS

Farewell, Dear Friend

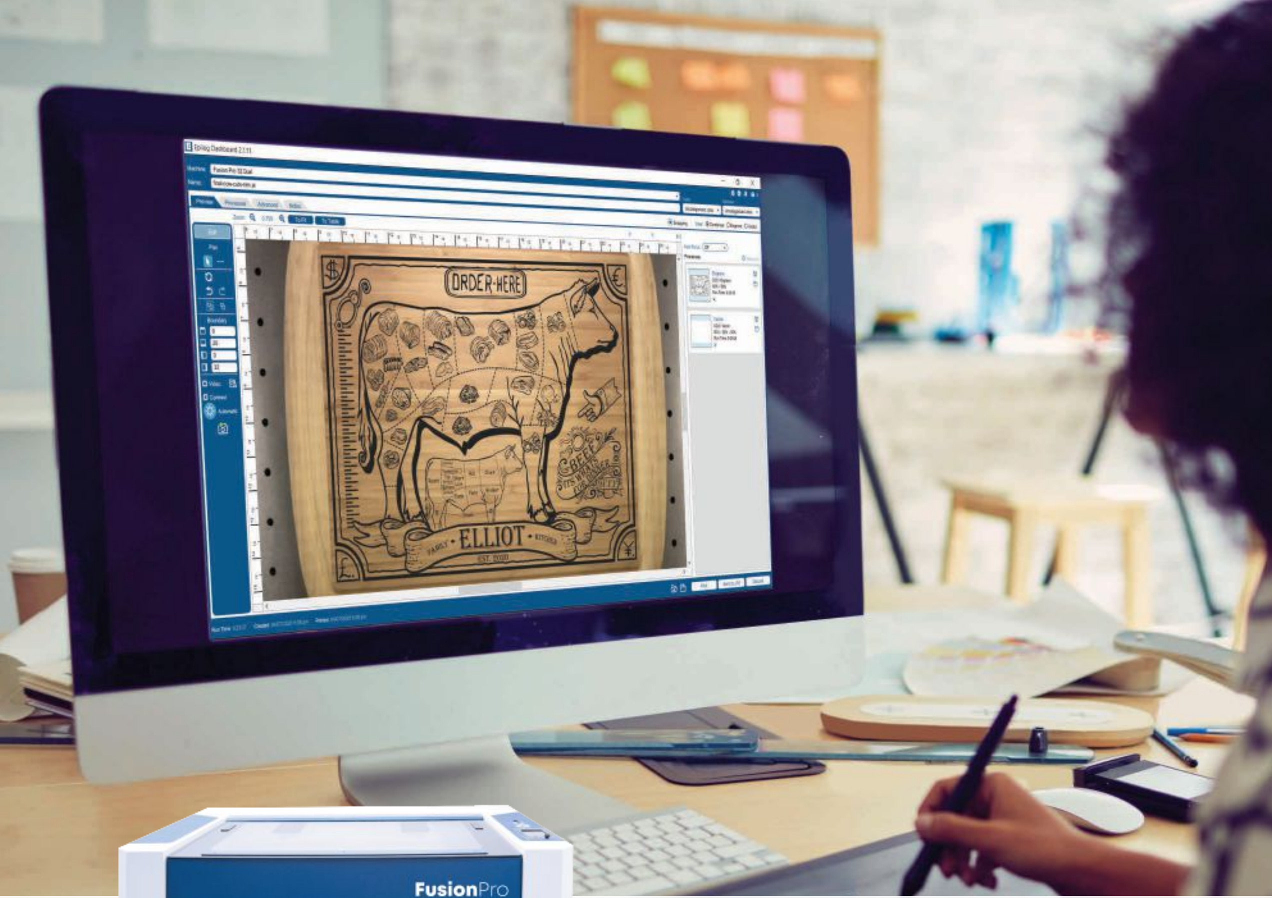
At *Make:*, we are mourning the loss of our colleague, **Patrick DiJusto**, our Books editor, who passed away unexpectedly this September. We are remembering Patrick's life as an author, editor, tech enthusiast and New Yorker.

From Patrick's bio: **"He has sworn to defend mankind against the eventual rise of the killer bots."** Patrick had a strong moral sense of how technology could be used for good and how it can be misused. An article he wrote for us in June, "MOSI-MISO and 140 Years of Wrong," advocated eliminating the "master/slave" terminology and markings from microcontrollers and other hardware components. You can hear Patrick's very nice voice on a Make:cast podcast with the authors of the *Make: Geometry* book (makezine.com/go/geometry-makecast). Our book authors valued his insights and help in shaping their books.

As a journalist, Patrick wrote for *Wired*, *The New Yorker*, *The Atlantic*, and many other publications. A *Washington Post* (2015) review of Patrick's own book said: "Patrick DiJusto has written a lively coffee-table book for the modern consumer. We could describe it here, but you'll get the point from its mouthful of a title: "This Is What You Just Put in Your Mouth? From Eggnog to Beef Jerky, the Surprising Secrets of What's Inside Everyday Products."

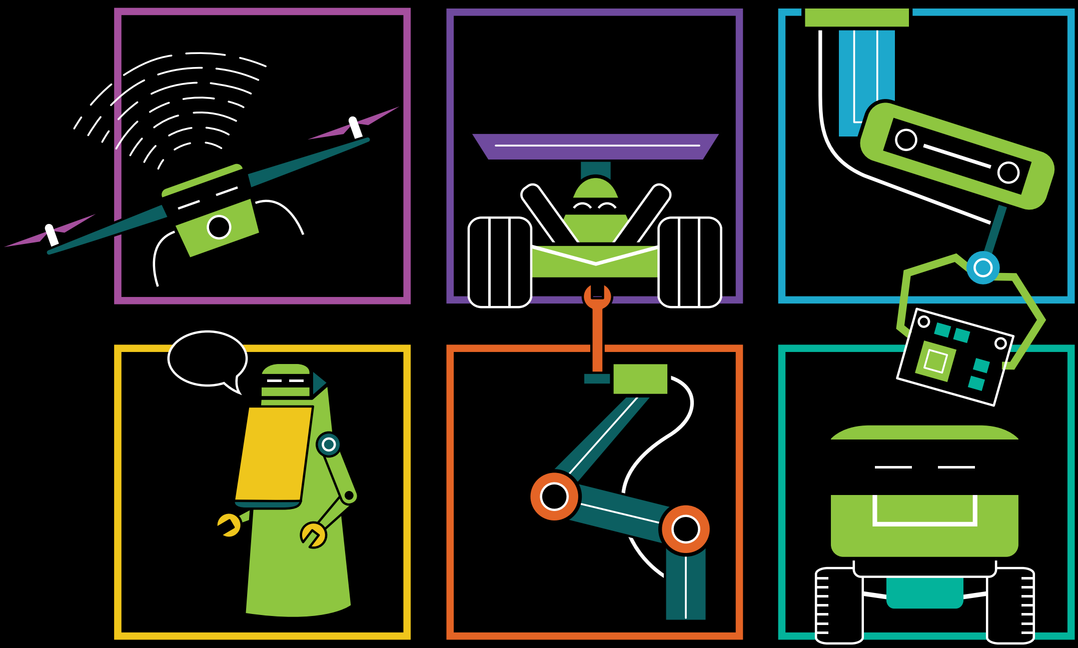
We are deeply saddened by the loss of Patrick as a colleague. He was smart, funny and enthusiastic. Our thoughts and love go out to his family, and his partner, Emily Gertz. Patrick and Emily were co-authors of several handbooks for us and like-minded partners in life. We can't easily accept losing Patrick and his sensibility. All of us at *Make:* and those in the maker community who knew him will miss him.

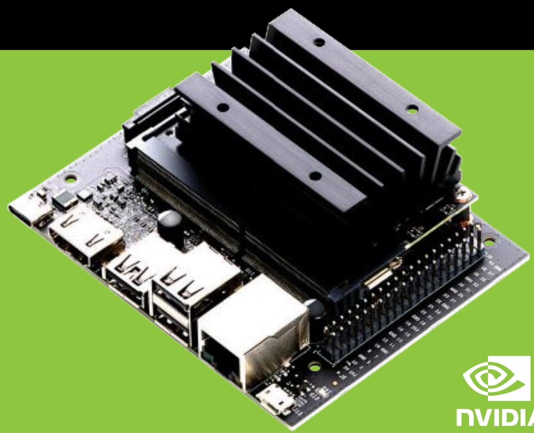
Share a memory of Patrick at
makezine.com/2021/09/22/remembering-patrick-dijusto


ORDER·HERE
ELLIOT


FusionPro
EpilogLaser.com

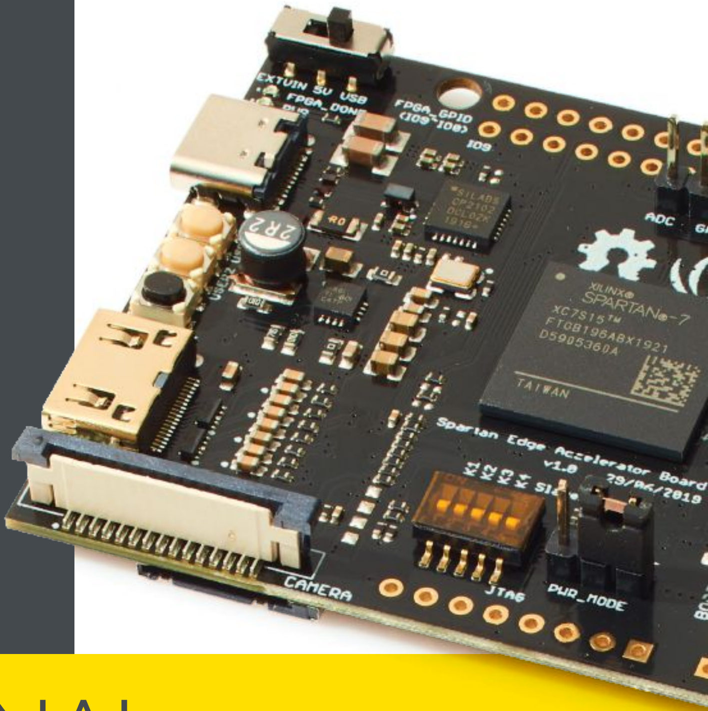

ORDER·HERE
FAMILY
ELLIOT
KITCHEN
EST. 2020





Make:

THE ORIGINAL GUIDE TO BOARDS



2021

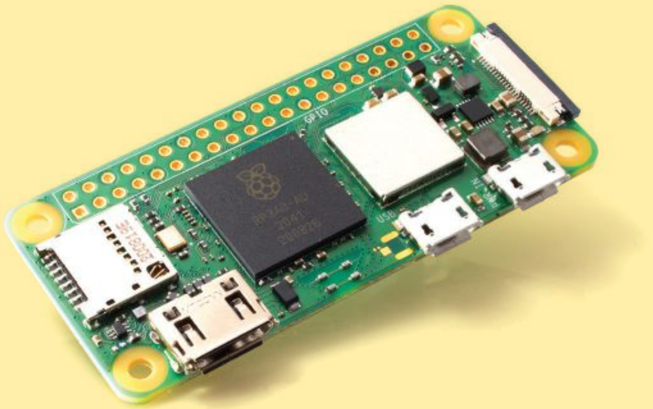
Boardspotting

The world of development boards has grown tremendously since the education-led microcontroller revolution of the mid-aughts. We've watched the emergence of easy-to-use learning tools from Arduino, Raspberry Pi, micro:bit, and others, and followed along as they've evolved into serious machines suitable even for professional applications. Makers and enthusiasts can now choose from a dizzying assortment of practical microcontroller, single board computer, and even FPGA options. Fear not, however, this guide will help you narrow down the options for your project. What are you working on next?

Scan the QR code to get the Boards Guide app and see the guide come to life in dynamic AUGMENTED REALITY!

STOP THE PRESSES!!

As we finalized this year's *Guide to Boards*, we got our hands on another surprise from Raspberry Pi — a much-desired update to the original Raspberry Pi Zero. This version, aptly named the Zero 2 W (yep, that means it includes wireless capabilities) crams the same processor under the hood as the Raspberry Pi 3, albeit slightly underclocked at 1GHz. Get all the specs on page 12.



LATEST TRENDS: THE PROS WEIGH IN

No one knows the board world better than those that live and breathe it on a daily basis. We've asked the leading board makers to chime in on 2021: what they've been working on, developments they've been watching, surprises that have come up, and what are some of their favorite new boards from their peers.

Scan this image with the **Boards Guide augmented reality app** to watch the video and hear from them all.

A CHIP IS BORN

RP2040-BASED BOARDS GO FROM 0–60+ WITH IMPRESSIVE SPEED

The microcontroller market — where chips designed for real-time embedded use are placed into development boards typically referred to as "microcontrollers" themselves — is rarely shaken up. If you want something friendly to beginners, you pick an Arduino with a Microchip ATmega328; if you want connectivity, an Espressif ESP32-based board; for performance, STMicro's STM32 series.

The Raspberry Pi Pico, a $4 development board with a powerful, custom dual-core Arm Cortex-M0+ microcontroller and flexible programmable input/output (PIO) blocks, shook things up with its launch this year — but it was the news that these new RP2040 chips would be sold individually at just $1 each that caused the biggest aftershocks.

Accompanying the Pico's launch, a number of manufacturers released boards based on the RP2040: Adafruit, Arduino, Pimoroni, and SparkFun to start. In the months since, nearly 70 announcements have followed with designs spanning the gamut from domain-specific gadgets like the Raspberry Fish synth from TINRS to general-purpose tinkering boards like Invector Labs' Challenger RP2040.

"We're pretty happy with how things are going," Raspberry Pi co-founder Eben Upton told us. "Lots of enthusiasm for the community as people really start to understand what the chip can do. We're still on track to clear our Pico backlog, and get RP2040 into high-volume, over 1 million unit availability before the end of the year."

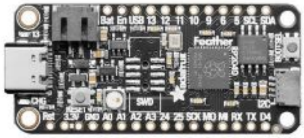
Adafruit's Limor Fried amplifies the excitement, saying "It's one of the best things to ever happen for electronics."

At time of writing, we've identified over 60 distinct boards that utilize this new chip, with more coming continuously. Here are images of just a few; see more at makezine.com/go/rp2040boards.

— Gareth Halfacree

Raspberry Pi Pico

Arducam Pico4ML



Adafruit Feather RP2040

Pimoroni PGA2040

Sparkfun Thing Plus RP2040

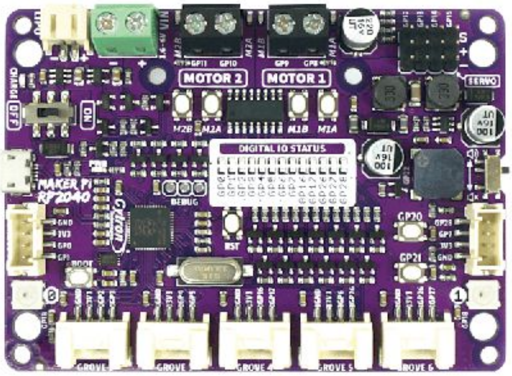
Pimoroni Tiny 2040

Adafruit Trinkey QT2040

Arturo 152 RP2040 Stamp

Arduino Nano RP2040 Connect

Seeed Xiao RP2040



Maker Pi RP2040

Pimoroni PicoSystem

Make: GUIDE TO BOARDS 2021

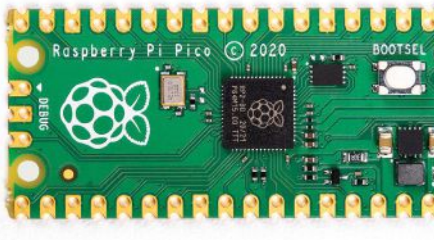
MICROCONTROLLERS (MCU)

* Price field reflects current market prices, which have been affected by 2021 component shortages

Board Name	Price*	Dimensions	Software	Clock Speed	Processor	Memory
Adafruit Circuit Playground Bluefruit	$25	2.0" dia.	Arduino IDE, CircuitPython	64MHz	32-bit Nordic nRF52840 (single-core Cortex-M4F)	1MB flash, 256 kB RAM, 2 MB QSPI flash
Adafruit CLUE	$45	2.0"×1.6"	Arduino IDE, CircuitPython	64MHz	32-bit Nordic nRF52840 (single-core Cortex-M4F)	1MB flash, 256kB RAM, 2MB QSPI flash
Adafruit Feather M0 with RFM95 LoRa Radio - 900MHz - RadioFruit	$35	2.0"×0.9"×0.3"	CircuitPython, Arduino IDE	48MHz	ATSAMD21G18 (single-core Cortex-M0)	256kB flash, 32kB RAM
Adafruit Feather RP2040	$12	2.0"×0.9"×0.3"	CircuitPython, MicroPython, Arduino IDE, C/C++	125MHz	32-bit RP2040 (dual-core Cortex M0+)	264kB SRAM, 8MB SPI flash
Adafruit Flora	$15	1.75" dia.	Arduino IDE	8MHz	8-bit ATmega32u4	32kB flash, 2.5kB SRAM
Adafruit FunHouse	$35	3.35"×2.2"×0.43"	CircuitPython, Arduino IDE	240MHz	32-bit ESP32-S2 (single-core Xtensa LX7)	4MB flash, 2MB PSRAM
Adafruit Gemma M0	$10	1.1" dia.	Arduino IDE, CircuitPython, MakeCode	48MHz	32-bit ATSAMD21 (single-core Cortex-M0+)	256kB flash, 32kB RAM
Adafruit ItsyBitsy RP2040	$10	1.4"×0.7"×0.2"	CircuitPython, MicroPython, Arduino IDE, C/C++	125MHz	32-bit RP2040 (dual-core Cortex M0+)	264kB RAM, 8MB SPI flash
Adafruit Metro ESP32-S2	$20	2"×2.8"×0.6"	CircuitPython, Arduino IDE	240MHz	32-bit ESP32-S2 (single-core Xtensa LX7)	4MB flash, 2MB PSRAM
Adafruit Neo Trinkey	$7	1.2"×0.5"×0.1"	CircuitPython, Arduino IDE	48MHz	32-bit ATSAMD21E18 (single-core Cortex M0+)	256kB flash, 32kB RAM
Adafruit QT Py RP2040	$10	0.9"×0.7"×0.2"	CircuitPython, MicroPython, C/C++	125MHz	32-bit RP2040 (dual-core Cortex M0+)	264kB RAM, 8MB SPI flash
Adafruit Trinket M0	$9	1.07"×0.6"	Arduino IDE, CircuitPython	48MHz	32-bit ATSAMD21E18 (single-core Cortex-M0+)	256kB flash, 32kB RAM
Adafruit Trinkey QT2040	$8	1.5"×0.7"×0.2"	CircuitPython, MicroPython, Arduino IDE, C/C++ SDK	125MHz	32-bit RP2040 (dual-core Cortex M0+)	264kB RAM, 8MB SPI flash
Arducam Pico4ML	$50	0.9"×2.0"	MicroPython	133MHz	32-bit RP2040 (dual-core Cortex M0+)	2MB flash 264kB RAM
Arduino Mega	$40	4.0"×2.1"	Arduino IDE	16MHz	8-bit ATmega2560	256kB flash, 8kB SRAM, 4kB EEPROM
Arduino MKR GSM 1400	$69	2.6"×1.0"	Arduino IDE	48MHz	32-bit ATSAMD21 (single-core Cortex-M0+)	256 kB flash, 32kB SRAM
Arduino MKR WAN 1310	$38	2.7"×1.0"	Arduino IDE	48MHz	32-bit ATSAMD21 (single-core Cortex-M0+)	256kB flash, 32kB SRAM, 2MB QSPI flash
Arduino MKR WiFi 1010	$35	2.4"×1.0"	Arduino IDE	48MHz	32-bit ATSAMD21 (single-core Cortex-M0+)	256kB flash, 32kB RAM
Arduino Nano 33 BLE Sense	$33	1.8"×0.7"	Arduino IDE	64MHz	32-bit Nordic nRF52840 (single-core Cortex-M4F)	1MB flash, 256kB RAM

Digital Pins	Analog Pins	Radio	Video	Input Voltage	Battery Connection	Operating Voltage
8	8 PWM, 6 ADC	Bluetooth	–	3.7V–5V	✓	3.3V
18	18 PWM, 8 ADC	Bluetooth	1.3" 240×240 Color IPS TFT LCD display	3V–6V	✓	3.3V
20	8 PWM, 10 ADC, 1 DAC	LoRa	–	3.3V–5V	✓	3.3V
21	16 PWM, 4 ADC	–	–	3.3V–5V	✓	3.3V
8	4 PWM, 4 ADC	–	–	3.5V–16V	✓	3.3V
5	5 PWM, 3 ADC	Wi-Fi	1.54" 240×240 Color TFT Display	5V	–	3.3V
3	2 PWM, 3 ADC, 1 DAC	–	–	3V–6V	✓	3.3V
23	16 PWM, 4 ADC	–	–	3.3V–5V	–	3.3V
25	8 PWM, 18 ADC, 2 DAC	Wi-Fi	–	3.7V–12V	✓	3.3V
0	0	–	–	5V	–	3.3V
13	13 PWM, 4 ADC	–	–	3.3V–5V	–	3.3V
5	5 PWM, 3 ADC, 1 DAC	–	–	3.3V/5V	–	3.3V
0	0	–	–	5V	–	3.3V
26	16 PWM, 3 ADC	–	0.96" 160×80 Color LCD	5V–5.5V	–	3.3V
54	15 PWM, 16 ADC	–	–	6V–20V	✓	5V
22	13 PWM, 7 ADC, 1 DAC	GSM 1400	–	3.7V–5V	✓	3.3V
22	13 PWM, 7 ADC, 1 DAC	LoRa	–	3.7V–5V	✓	3.3V
22	13 PWM, 7 ADC, 1 DAC	Bluetooth	–	3.7V–5V	✓	3.3V
22	5 PWM, 8 ADC	Bluetooth	–	5V–21V	–	3.3V

NEW & NOTABLE



RASPBERRY PI PICO

The speedy Pico is the first microcontroller developed by the Raspberry Pi Foundation and is powered by their new RP2040 chip. Pico also provides something that other boards don't: eight Programmable I/O (PIO) state machines to interface with unsupported devices or additional peripherals. And at $4? It's a steal.

BBC MICRO:BIT V2

The updated micro:bit V2 keeps its familiar form factor and education-friendly design, but features a big performance boost that lets it handle more complex and heavy tasks (like AI and machine learning!) than its predecessor. And it's still cheap enough that it won't break the bank outfitting a classroom full of students.

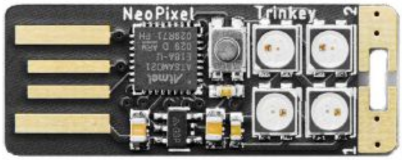
Make: GUIDE TO BOARDS 2021

MICROCONTROLLERS (MCU)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
Arduino Nano Every	$13	1.8"×0.7"	Arduino IDE	20MHz	8-bit ATMega4809	48kB flash, 6kB RAM, 256B EEPROM
Arduino Nano RP2040 Connect	$25	0.7"×1.8"	Arduino IDE, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	16MB flash, 264 kB RAM
Arduino Portenta H7	$103	2.6"×1.0"	Arduino IDE, MicroPython, JavaScript, TensorFlow Lite, Mbed OS	480MHz Cortex-M7, 240MHz Cortex-M4	32-bit STMicro STM32H747XI (dual-core Cortex-M7, M4 coprocessor)	2MB/16MB Int/Ext flash, 1MB/8MB Int/Ext RAM
Arduino Portenta H7 Lite	$72	2.6"×1.0"	Arduino IDE, MicroPython, JavaScript, TensorFlow Lite, Mbed OS	480MHz Cortex-M7, 240MHz Cortex-M4	32-bit STM32H747XI dual-core Cortex-M7+M4	2MB/16MB Int/Ext flash, 1MB/8MB Int/Ext RAM
Arduino Uno/Uno WiFi Rev2	$23 / $45	2.7"×2.1"	Arduino IDE	16MHz	8-bit ATmega328PU/ATmega4809	32kB flash, 2kB RAM, 1kB EEPROM/48kB flash, 6kB RAM, 256B EEPROM
BBC micro:bit V2	$15	2"×1.6"	Javascript, MicroPython, CircuitPython, C++	64MHz	32-bit Nordic nRF52833 (single-core Cortex-M4F)	512kB flash, 128kB RAM
Cypress PSoC 62S2 Pioneer Kit	$113	5.08"×3.62"	ModusToolbox, Mbed OS	150MHz Cortex-M4, 100MHz Cortex-M0+	32-bit Cypress PSOC 62 (Cortex M4, Cortex M0+ coprocessor)	2MB/64MB Int/Ext flash, 1MB SRAM, 512kB FRAM
Espressif ESP32-S2 Saola-1	$8	2.22"×1.1"	Arduino IDE, CircuitPython, ESP IDF	240MHz	32-bit ESP32-S2 (single-core Xtensa LX7)	128kB flash, 320kB SRAM, 16kB SRAM (RTC)
Espressif ESP8266-DevKitC	$8	1.77"×1"	Arduino IDE, MicroPython, NodeMCU	80MHz	32-bit ESP8266 (single-core Tensilica L106)	2MB flash, 80kB RAM
Espruino Pico	$25	1.3"×0.6"	Espruino JavaScript Interpreter	84MHz	32-bit STMicro STM32F401CDU6 (single-core Cortex-M4)	384kB flash, 96kB RAM
Invector Labs Challenger M0 WiFi Feather	$16	2.0"×0.9"×0.3"	Arduino IDE, Micropython	48MHz	32-bit ATSAMD21G18A (single-core Cortex-M0+)	256kB/2MB Int/Ext flash, 32 kB SRAM, 1MB flash (ESP8285)
LilyPad Arduino USB	$25	2.0" dia.	Arduino IDE	8MHz	8-bit ATmega32u4	32kB flash, 2.5kB SRAM, 1kB EEPROM
M5 Stamp Pico	$5	0.71"×0.94"×0.17"	Arduino IDE, MicroPython, UIFlow	240MHz	32-bit ESP32-PICO-D4 (dual-core Xtensa LX6)	4MB flash, 520kB SRAM, 8kB SRAM (RTC)
M5Stick C+ dev kit	$16	1.89"×0.95"×0.71"	Arduino IDE, MicroPython, UIFlow	240MHz	32-bit ESP32-PICO-D4 (dual-core Xtensa LX6)	4MB flash, 520kB SRAM, 8kB SRAM (RTC)
Meadow F7 v2	$45	1.9"×0.9"	Meadow.OS	216MHz Cortex-M7, 240MHz ESP32	32-bit STM32F7 (single-core Cortex-M7), ESP32 coprocessor	64MB flash, 32MB RAM
Microchip AVR128DA48 Curiosity Nano	$19	3.3"×0.77"	MPLAB X / Microchip Studio	24MHz	8-bit AVR128DA48	128kB flash, 16kB SRAM, 512B EEPROM
Nordic Thingy:52	$37	2.4"×2.4"	Nordic Thingy, Zephyr OS	64MHz	32-bit Nordic nRF52832 (single-core Cortex-M4F)	512kB flash, 64kB RAM
Nordic Thingy:91	$121	2.4"×2.4"	Nordic Thingy, Zephyr OS	64MHz	32-bit Nordic nRF9160 (single-core Cortex-M33)	1MB flash, 256kB RAM
OpenMV Cam H7 R2	$65	1.77"×1.41"	MicroPython	480MHz	32-bit STMicro STM32H743VI (single-core Cortex-M7)	2MB flash, 1MB RAM, µSD Card Slot

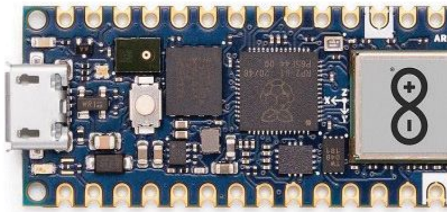
Digital Pins	Analog Pins	Radio	Video	Input Voltage	Battery Connection	Operating Voltage
22	5 PWM, 8 ADC	–	–	7V–21V	–	5V
22	20 PWM, 8 ADC	Wi-Fi, Bluetooth	–	5V–21V	–	3.3V
22	10 PWM, 8 ADC, 2 DAC	Wi-Fi, Bluetooth	MIPI DSI Host & MIPI D-PHY	3.7V–5V	✓	3.3V
22	10 PWM, 8 ADC, 2 DAC	–	–	3.7V–5V	✓	3.3V
14	6 PWM, 6 ADC/ 5 PWM, 6 ADC	Uno WiFi: Wi-Fi, Bluetooth	–	6V–20V/ 7V–12V	–	5V
19	3 PWM, 6 ADC	Bluetooth	–	3V–5V	✓	3V–3.3V
48	32 PWM, 15 ADC	Wi-Fi, Bluetooth	–	7V–12V	–	1.8V/3.3V
43	8 PWM, 20 ADC, 2 DAC	Wi-Fi	Serial LCD, Parallel LCD	3.3V–5V	–	3.3V
11	4 PWM, 1 ADC	Wi-Fi	–	3.3V–5V	–	3.3V
22	21 PWM, 9 ADC	–	–	3.5V–16V	–	3.3V
20	20 PWM, 6 ADC, 1 DAC	Wi-Fi	–	3.3V–5V	✓	3.3V
9	4 PWM, 4 ADC	–	–	2.7V–5.5V	–	3.3V
12	2 PWM, 3 ADC, 2 DAC	Wi-Fi, Bluetooth	–	5V	–	5V
3 (2 more via Grove)	1 ADC	Wi-Fi, Bluetooth	0.96" 80×160 RGB LCD	5V	✓	3.3V
24	12 PWM, 6 ADC, 2 DAC	Wi-Fi, Bluetooth	–	3.3V–12V	✓	3.3V (5V-tolerant digital IO)
40, 8 digital-only	3 PWM, 8 ADC, 1 DAC, 3 Analog Comparator	–	–	5V	–	1.6V–5V
14	4 PWM, 6 ADC	Bluetooth	–	5V	✓	3.3V
10	4 PWM, 3 ADC	LTE-M, NB-IoT, Bluetooth LE, NFC	–	5V	✓	3.3V
10	10 PWM, 1 ADC, 1 DAC	–	680x480 RGB Camera	5V	✓	3.3V

NEW & NOTABLE



ADAFRUIT NEO TRINKEY

This tiny little board is part USB key, part microcontroller, emphasis on the micro. Coming in just slightly longer than a quarter, but even thinner, this board is perfect for simple tasks such as notifications or hot keys. The M0 processor can more than handle the four built-in RGB NeoPixel LEDs and two capacitive touch pads. The Neo Trinkey is one of many Trinkeys in a new line from Adafruit.



ARDUINO NANO RP2040 CONNECT

The diminutive Nano Connect RP2040's wireless connectivity (both Wi-Fi and Bluetooth/BLE) provide compatibility to the Arduino Cloud for IoT endeavors. A built-in microphone input and the 6-axis inertial sensor allows for onboard environmental awareness. Its 133MHz RP2040 chip offers speeds beyond those of the other Arduino Nano variants, and the 16MB of processor-external memory is enough for advanced tasks.

MICROCONTROLLERS (MCU)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
Particle Argon	$28	2.0"×0.8"	Particle Device OS	64MHz Cortex-M4F, 240MHz ESP32	32-bit Nordic nRF52840 (single-core Cortex-M4F), ESP32 coprocessor	1MB flash, 256kB RAM, 4MB SPI flash, 4MB flash (ESP32)
Particle Boron	$59	2.0"×0.9"	Particle Device OS	64MHz	32-bit Nordic nRF52840 (single-core Cortex-M4F)	1MB flash, 256kB RAM, 4MB SPI flash
Particle Electron Development Kit	$72	2.0"×0.8"×0.3"	Particle Device OS	120MHz	32-bit STMicro STM32F205 (single-core Cortex-M3)	1MB flash, 128kB RAM
Pimoroni PGA2040	$8	0.8"×0.8"	Arduino IDE, MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	8MB QSPI flash, 264kB RAM
Pimoroni Pico LiPo	$15 (16MB), $11 (4MB)	2.09"×0.83"	Arduino IDE, MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	4MB/16MB QSPI flash, 264kB RAM
Pimoroni Plasma 2040	$15	1.9"×1.1"×0.5"	MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	2MB QSPI flash, 264kB RAM
Pimoroni Tiny 2040	$10	0.90"×0.72"	Arduino IDE, MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	8MB QSPI flash, 264kB RAM
PJRC Teensy 4.0	$20	1.4"×0.7"	Arduino IDE with Teensyduino extension, CircuitPython	600MHz	32-bit NXP iMX RT1062 (single-core Cortex-M7)	2MB flash, 1MB RAM, 1kB EEPROM (emulated)
PJRC Teensy 4.1	$27	2.4"×0.7"	Arduino IDE with Teensyduino extension, CircuitPython	600MHz	32-bit NXP iMX RT1062 (single-core Cortex-M7)	8MB flash, 1MB RAM, 4kB EEPROM (emulated)
Pycom FiPy	$61	2.1"×0.7"	MicroPython	160MHz	32-bit ESP32 (dual-core Xtensa LX6)	8MB flash, 4MB RAM
Pycom Lopy4	$40	2.1"×0.7"	MicroPython	160MHz	32-bit ESP32 (dual-core Xtensa LX6)	8MB flash, 4MB RAM
Raspberry Pi Pico	$4	2"×0.827"	Arduino IDE, MicroPython, CircuitPython, FreeRTOS, RT-Thread, Rust, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	2MB flash, 264kB RAM
Seeed Wio Terminal	$37	2.83"×2.24"× 0.47"	Arduino IDE, MicroPython, CircuitPython, ArduPy	120MHz	32-bit ATSAMD51 (single-core Cortex-M4F)	4MB SPI flash, 192kB RAM
Seeed Xiao RP2040	$5	0.8"×0.7"	Arduino IDE, MicroPython, CircuitPython	133MHz	32-bit RP2040 (dual-core Cortex M0+)	2MB flash, 264kB RAM
Sipeed Maixduino	$28	2.7"×2.1"	MaixPy IDE, Arduino IDE, MicroPython, OpenMV IDE, PlatformIO IDE, FreeRTOS	400MHz	64-bit Sipeed M1 (dual-core RISC-V), KPU coprocessor	16MB flash, 8MB SRAM
Sony Spresense	$65	1.96"×0.81"	NuttX emulating Arduino IDE, CircuitPython	156MHz	32-bit Sony CXD5602 (six-core Cortex-M4F)	8MB flash, 1.5MB SRAM
SparkFun ESP32 Thing	$22	2.35"×1.0"	ESP-IDF, Arduino IDE	240MHz	32-bit ESP32 (dual-core Tensilica LX6)	4MB flash, 520kB SRAM
SparkFun MicroMod Artemis	$15	0.86"×0.86"	Arduino IDE, TensorFlow Lite, C++ with the Ambiq SDK	48MHz/ 96MHz Turbo	32-bit Ambiq Apollo3 (single-core Cortex-M4F)	1M flash, 384kB RAM
SparkFun MicroMod RP2040	$12	0.86"×0.86"	Arduino IDE, MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	16MB flash, 264kB SRAM

Digital Pins	Analog Pins	Radio	Video	Input Voltage	Battery Connection	Operating Voltage
20	8 PWM, 6 ADC	Wi-Fi, Bluetooth	–	4.5V–5.5V	✓	3.3V
20	8 PWM, 6 ADC	LTE-M1, Bluetooth, NFC	–	USB 4.5V–5V, Battery 3.6V–4.2V	✓	3.3V
28	11 PWM, 14 ADC	Cellular 2G/3G	–	3.9V–12V	✓	3.3V
30	16 PWM, 4 ADC	–	–	3V–5.5V	–	3.3V
23	16 PWM, 3 ADC	–	–	3V–5.5V	✓	3.3V
3	3 PWM, 3 ADC	–	–	5V	–	3.3V
12	12 PWM, 4 ADC	–	–	3V–5.5V	–	3.3V
40	31 PWM, 14 ADC	–	–	3.6V–5.5V	–	3.3V
55	35 PWM, 18 ADC	–	–	3.6V–5.5V	–	3.3V
22	18 PWM, 8 ADC, 2 DAC	Wi-Fi, Bluetooth, CAT–M1/NB–IOT, LoRa, Sigfox	–	3.3V–5V	–	3.3V
24	18 PWM, 8 ADC, 2 DAC	Wi-Fi, Bluetooth, LoRa, Sigfox	–	3.3V–5V	–	3.3V
26	16 PWM, 3 ADC	–	–	1.8V–5.5V	–	3.3V
26	5 PWM, 9 ADC	Wi-Fi, Bluetooth	2.4" 320x240 Color LCD	5V	–	3.3V
11	11 PWM, 4 ADC	–	–	5V	–	3.3V
48	6 ADC	Wi-Fi, Bluetooth	8-bit LCD interface	5V–12V	–	3.3V
17; extension:14	2 ADC; extension: 6 PWM, 6 ADC	–	–	5V	–	1.8V
28	16 PWM, 18 ADC, 2 DAC	Wi-Fi, Bluetooth	–	2.2V–5V	✓	3.3V
8; 48 via shared MIcroMod pins	2 PWM, 2 ADC, 31 PWM, 10 ADC (via pins)	Bluetooth	–	3.3V	–	1.755V–3.63V
30	16 PWM, 3 ADC	–	–	3.3V	–	3.3V

NEW & NOTABLE

SPARKFUN MICROMOD TEENSY

The MicroMod system puts interchangeable processors onto M.2-socket-laden boards, allowing users to jump between platforms by simply swapping the modules on the carrier board, of which there are many options. The Teensy MicroMod puts the ridiculously fast (up to 600MHz!) Teensy processor into numerous scenarios. If you need speed, this is worth a look.

PIMORONI TINY2040

This postage stamp-sized board really brings the micro to microcontroller. It's tiny enough to fit anywhere, but, powered by the RP2040 chip, still offers 4 ADCs, a debug port, an RGB LED, and 8 Megs of storage. Pimoroni even set it up to allow the boot button to double as a user input — a surprisingly handy feature.

Make: GUIDE TO BOARDS 2021

MICROCONTROLLERS (MCU)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
SparkFun MicroMod Teensy	$20	0.86"×0.86"	Arduino IDE, C/C++	600MHz/ 1GHz Turbo	32-bit NXP iMX RT1062 (single-core Cortex-M7)	16MB flash, 1MB RAM
SparkFun Pro Micro - RP2040	$10	1.3"×0.7"	Arduino IDE, MicroPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	16MB flash, 264kB RAM
SparkFun RedBoard Edge	$23	4.0"×1.5"	Arduino IDE	16MHz	8-bit ATmega328	32kB flash, 2kB SRAM, 1kB EEPROM
SparkFun Thing Plus RP2040	$18	0.9"×2.3"	Arduino IDE, MicroPython, CircuitPython, C/C++	133MHz	32-bit RP2040 (dual-core Cortex M0+)	16MB QSPI flash, 264kB RAM
Texas Instruments TM 4C1294XL	$24	4.9"×2.2"×0.43	Energia, Code Composer, others	120MHz	32-bit TM4C1294NCPDT (single-core Cortex-M4F)	1MB flash, 256kB SRAM, 6kB EEPROM
TinyLily Mini	$10	0.55" dia.	Arduino IDE	8MHz	8-bit ATmega328P	32kB flash, 2kB SRAM, 1kB EEPROM
Unexpected Maker FeatherS2	$22	0.9"×2.0"	Arduino IDE, MicroPython, CircuitPython, ESP-IDF	240MHz	32-bit ESP32-S2 (single-core Xtensa LX7)	16MB SPI flash, 320kB SRAM, 8MB PSRAM
Unexpected Maker FeatherS2 NEO	$20	0.9"×2.0"	Arduino IDE, MicroPython, CircuitPython, ESP-IDF	240MHz	32-bit ESP32-S2 (single-core Xtensa LX7)	4MB SPI flash, 320kB SRAM, 2MB PSRAM
Unexpected Maker TinyPICO V2	$20	0.71"×1.26"	Arduino IDE, MicroPython, CircuitPython, ESP-IDF	240MHz	32-bit ESP32 (dual-core Xtensa LX6)	4MB SPI flash, 520kB SRAM, 4MB PSRAM, 8kB SRAM (RTC)

SINGLE-BOARD COMPUTERS (SBC)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
Asus Tinker Edge R	$235	3.9"×2.8"	Debian 9, Android 10	1.8GHz Cortex-A72, 1.4GHz Cortex-A53	64-bit Rockchip RK3399Pro (dual-core Cortex-A72, quad-core Cortex-A53)	4GB dual-channel LPDDR4 (System) + LPDDR3 2GB (NPU) RAM, 16GB eMMC
Asus Tinker Board 2S	$129	3.37"×2.13"	Debian 9, Android 10	2.0GHz Cortex-A72, 1.5GHz Cortex-A53	64-bit Rockchip RK3399 (dual-core Cortex-A72, quad-core Cortex-A53)	2GB/4GB dual-channel LPDDR4 RAM, 16GB eMMC
Banana Pi M2 Berry	$32	3.35"×2.2"	Ubuntu 16.04, Debian 9, Raspbian Stretch, Armbian Bionic, Tina Linux, Android 6	1GHz	32-bit Allwinner R40 (quad-core Cortex-A7)	1GB DDR3 RAM
BeagleBoard PocketBeagle	$39	2.2"×1.4"	Debian 10, Debian 9, Cloud 9 IDE	1GHz	32-bit Sitara AM3357 (Cortex-A8), 2 PRU coprocessors	512MB DDR3 RAM
BeagleBone AI	$116	3.4"×2.1"	Debian 10, Debian 9, Cloud 9 IDE	1.5GHz	32-bit Sitara AM5729 (dual-core Cortex-A15), 2 32-bit Cortex-M4, 2 C66x, 4 Vision Engine, 4 PRU, SGX544 GPU coprocessors	1GB DDR3L RAM, 16GB eMMC
BeagleBone Black Wireless	$55	3.4"×2.1"	Debian 10, Debian 9, Cloud 9 IDE	1GHz	32-bit Sitara AM335X (Cortex-A8), 2 PRU, SGX530 coprocessors	512MB DDR3 RAM, 4GB eMMC
BeagleBone Blue	$70	3.4"×2.1"	Debian 10, Debian 9, Cloud9 IDE, libroboticscape	1GHz	32-bit Sitara AM335x (Cortex-A8), Cortex-M3, 4 PRU coprocessors	512MB DDR3 RAM, 4GB eMMC
DFRobot LattePanda v1	$99/$159	2.8"×3.47"	Windows 10, Ubuntu 16.04, OpenSuSE 15	1.92GHz	64-bit Intel Z8350 (quad-core x86-64), ATmega32u4 coprocessor	2GB DDR3L RAM 32GB eMMC/4GB DDR3L RAM, 64GB eMMC

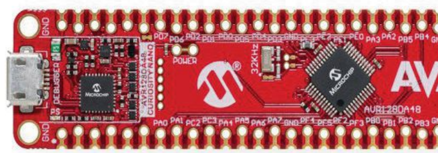
Digital Pins	Analog Pins	Radio	Video	Input Voltage	Battery Connection	Operating Voltage
12; 35 via shared MicroMod pins	16 PWM, 2 ADC (14 with signal sharing)	–	–	3.3V	–	3.3V
20–18 on the board edge and 2 through the Qwiic connector	10 PWM, 3 ADC	–	–	5V	–	3.3V
14	4 PWM, 8 ADC	–	–	7V–15V	–	5V
18	16 PWM, 3 ADC	–	–	3.7V–5V	✓	3.3V
84	8 PWM, 20 ADC, 3 Analog Comparators	–	–	4.75V–5.25V	–	3.3V
8	1 PWM, 4 ADC	–	–	2.7V–5.5V	–	3V
21	21 PWM, 13 ADC, 2 DAC	Wi-Fi	–	3.7V–5V	✓	3.3V
22	22 PWM, 13 ADC, 2 DAC	Wi-Fi	–	3.7V–5V	✓	3.3V
14	14 ADC, 2 DAC	Wi-Fi, Bluetooth	–	5V	✓	3V

Digital Pins	Analog Pins	Radio	Video	Ethernet On Board	Input Voltage	Operating Voltage
28	3 PWM	Wi-Fi, Bluetooth, Mini PCIe slot (for for 4G/LTE)	HDMI, USB-C, MIPI DSI	✓	12V–19V	5V
28	3 PWM	Wi-Fi, Bluetooth	HDMI, USB-C, MIPI DSI	✓	12V–19V	3.3V and 5V
28	4 PWM	Wi-Fi, Bluetooth	HDMI, MIPI DSI, LVDS	✓	5V	3.3V
44	4 PWM, 8 ADC (6 at 1.8V, 2 at 3.3V)	–	–	–	5V	3.3V
69	7 PWM, 7 ADC	Wi-Fi, Bluetooth	Micro-HDMI	✓	5V	1.8V and 3.3V
69	7 PWM, 7 ADC	Wi-Fi	Micro-HDMI	✓	5V	1.8V and 3.3V
8	4 PWM, 4 ADC	Wi-Fi, Bluetooth	–	–	9V–18V	1.8V–7.4V
6, 20 via ATmega32u4	6 PWM, 12 ADC	Wi-Fi, Bluetooth	HDMI, MIPI DSI	✓	5V	5V

NEW & NOTABLE

M5 STAMP PICO

The Stamp Pico is an incredibly compact controller, even with its heat-resistant plastic front cover installed. Based on the ESP32-PICO-D4, it offers wireless integration with the UIFlow programming software, allowing for simple and fast application development. Compatibility with other M5Stack components will let you build almost any project idea, quick and easy.



MICROCHIP CURIOSITY NANO

Debugging a microcontroller project can be a specialized skill. The new AVR-based Curiosity Nano board from Microchip makes that a lot less painful by including a hardware debugger directly on the board itself. Program in Atmel Studio 7 or MPLAB X IDE. Compatible with growing list of adapters and sensors for evaluation and full project making.

Make: GUIDE TO BOARDS 2021

SINGLE-BOARD COMPUTERS (SBC)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
Google Coral Dev Board Mini	$100	1.9"×2.52"	Mendel Linux (Debian based)	1.5GHz	64-bit MediaTek 8167s (quad-core Cortex-A35), GE8300, Edge TPU coprocessors	2GB LPDDR3 RAM, 8GB eMMC
Google Coral Edge TPU Dev Board	$130/$170	3.5"×2.4"	Mendel Linux (Debian based)	1.3GHz	64-bit NXP i.MX8MQ (quad-core Cortex-A53), Cortex-M4F, GC7000 Lite, Edge TPU coprocessors	1GB/4GB LPDDR4 RAM, 8GB eMMC
Hackboard 2	$199/$249	4.72"×3.15"	Microsoft Windows 10 Pro, Debian 9 (subtract $24)	2.8GHz	64-bit Intel Celeron N4020 (dual-core x86-64)	4GB/8GB LPDDR4, 64GB eMMC
Mangoh Yellow	$140	1.65"×2.56"	Legato Linux	1.2GHz	32-bit Sierra WP7702 (single-core Cortex-A7)	2GB DDR RAM, 4GB eMMC
Myir MYS-8MMX	$99/$119	3.74"×2.72"	Ubuntu 18.04, Yocto 3.0	1.8GHz/ 1.6GHz	64-bit NXP i.MX 8M Mini (quad-core Cortex-A53), Vivante GC320, Cortex-M4F coprocessors	2GB DDR4 RAM, 8GB eMMC, 32MB QSPI flash
Nvidia Jetson Nano Dev Kit	$54/$99	3.95"×3.15"× 1.14"	Ubuntu-based JetPack	1.43GHz CPU, 921MHz GPU	64-bit Nvidia CPU (quad-core Cortex-A57), 128-CUDA-core Maxwell GPU coprocessor	2GB/4GB LPDDR4 RAM
Nvidia Jetson Xavier NX Dev Kit	$399	4.06"×3.56"× 1.22"	Ubuntu-based JetPack SDK	1.9GHz CPU, 1.1GHz GPU	64-bit Carmel ARM CPU (six-core Cortex-A57), 384-CUDA-core 48-Tensor-core Volta GPU, 2 NVDLA, 7-way Vision Processor coprocessors	8GB 128-bit LPDDR4x RAM
Odroid C4	$54	3.35"×2.2"	Ubuntu 20.04, Android 9	2.0GHz CPU, 640MHz GPU	64-bit Amlogic S905X3 (quad-core Cortex-A55), Mali-G31 GPU	4GB 32-bit DDR4 RAM
Odroid N2+	$66/$83	3.54"×3.54"	Ubuntu 20.04, Android 9	2.4GHz Cortex-A73, 2GHz Cortex-A53	64-bit Amlogic S922X (quad-core Cortex-A73, dual-core Cortex-A53); Mali-G52 GPU	2GB/4GB 32-bit DDR4 RAM
Onion Omega2+	$13	1.1"×1.7"	Customized OpenWRT	580MHz	32-bit MT7688 (single-core MIPS)	128MB DDR2 RAM, 32MB Flash
Qualcomm DragonBoard 410c	$75	3.35"×2.12"	Debian 8, Ubuntu Core, Windows 10 IoT Core, Open Embedded, Andriod 5.1	1.2GHz	64-bit Snapdragon 410 (quad-core Cortex-A53) CPU, Adreno 306 GPU, Hexagon QDSP6 V5 coprocessor	1GB LPDDR3 RAM, 8GB eMMC
Raspberry Pi 3, Model A+	$25	2.6"×2.2"	Raspberry Pi OS, Raspbian, Ubuntu 20.04/21.04, RISC OS, Windows 10 IoT, more	1.4GHz CPU, 400MHz GPU	64-bit Broadcom BCM2837 (quad-core Cortex-A53), VideoCore IV GPU	512MB LPDDR2 RAM
Raspberry Pi 4, Model B	$35/$55/ $75	3.4"×2.2"	Raspberry Pi OS, Raspbian, Ubuntu 20.04/21.04, RISC OS, Windows 10 IoT, more	1.5GHz CPU, 500MHz GPU	64-bit Broadcom BCM2711 (quad-core Cortex-A72), VideoCore VI GPU	2GB/4GB/8GB LPDDR4 RAM
Raspberry Pi Zero W	$10	2.56"×1.18"	Raspberry Pi OS, Raspbian, RISC OS, more	1GHz CPU, 400MHz GPU	32-bit Broadcom BCM2835 (single-core ARMv6), VideoCore IV GPU	512MB LPDDR2 RAM
Raspberry Pi Zero W 2	$15	2.56"×1.18"	Raspberry Pi OS, Raspbian, Ubuntu 20.04/21.04, RISC OS, Windows 10 IoT, more	1GHz CPU, 400MHz GPU	64-bit Broadcom BCM2837 (quad-core Cortex-A53), VideoCore IV GPU	512MB LPDDR2 RAM
Rock Pi 4	$39–$75	3.37"×2.22"	Debian 10, Ubuntu 20.04, Android 7/9/10, more	1.8GHz Cortex-A72, 1.4GHz Cortex-A53	64-bit Rockchip RK3399 (dual-core Cortex-A72, quad-core Cortex-A53) CPU, Mali T860MP4 GPU	1GB/2GB/4GB 64-bit Dual-Channel LPDDR4 RAM
Seeed Odyssey STM32MP157C Evaluation Board	$55	2.2"×3.3"	Debian 10	800MHz Cortex-A7, 209MHz Cortex-M4	32-bit STMicro STM32MP157C (dual-core Cortex-A7, Cortex-M4 coprocessor)	512MB DDR3 RAM, 4GB eMMC
Seeed Odyssey X86J4125800	$218	4.33"×4.33"	Windows 10, Ubuntu 16.04, OpenSuSE 15, OpenWRT	2GHz–2.7GHz	64-bit Intel Celeron J4125 (quad-core x86-64), SAMD21 Cortex-M0+ coprocessor	8GB LPDDR4 RAM
UDOO Bolt	$459	4.72"×4.72"	Windows 8.1/10, any Linux distribution	2GHz–3.6GHz	AMD Ryzen Embedded V1000 (quad-core eight-thread x86-64) CPU, Radeon Vega 8 GPU, ATmega32u4 coprocessor	None included, supports up to 32GB DDR4 RAM, 32GB eMMC

Digital Pins	Analog Pins	Radio	Video	Ethernet On Board	Input Voltage	Operating Voltage
28	3 PWM	Wi-Fi, Bluetooth	Micro-HDMI, MIPI DSI	—	5V	5V
28	3 PWM	Wi-Fi, Bluetooth	HDMI, MIPI DSI	✓	5V	5V
28	2 PWM	Wi-Fi, Bluetooth	HDMI, eDP	—	12V	GPIO 5V; 3.3V
6	2 ADC	Wi-Fi, Bluetooth, Cellular (NB-IOT, cat-m1, 2G, 3G, LTE cat-1, cat-4), NFC, GPS	—	—	4.75V–6V	3.3V
30	3 PWM	Wi-Fi, Bluetooth	HDMI, LVDS LCD	✓	5V	5V
28	2 PWM	—	HDMI, DP (4GB only)	✓	5V	5V
28	2 PWM	Wi-Fi, Bluetooth	HDMI, DP	✓	9V–20V	5V
25	6 PWM, 2 ADC	—	HDMI	✓	5.5V–13V	1.8V and 3.3V
25	4 PWM, 2 ADC	—	HDMI, Composite	✓	7.5V–18V	1.8V and 3.3V
18	2 PWM	Wi-Fi	—	✓ Via pins	3.3V	3.3V
12	1 PWM	Wi-Fi, Bluetooth	HDMI, MIPI DSI	—	6.5V–18V	1.8V
26	4 PWM	Wi-Fi, Bluetooth	HDMI, Composite, MIPI DSI	—	5V	3.3V
26	4 PWM	Wi-Fi, Bluetooth	2 micro-HDMI, Composite, MIPI DSI	✓	5V	3.3V
26	4 PWM	Wi-Fi, Bluetooth	Mini-HDMI, Composite	—	5V	3.3V
26	4 PWM	Wi-Fi, Bluetooth	Mini-HDMI	—	5V	3.3V
27	1 PWM, 1 ADC	Wi-Fi, Bluetooth (on Models B and C only)	HDMI, MIPI DSI, mini-DP (Model C only)	✓	6V–28V	3.3V
28, 2 via Grove	4 PWM	Wi-Fi, Bluetooth	HDMI, MIPI DSI	✓	5V, 12v~24V	5V
53	20 PWM, 6 ADC	Wi-Fi, Bluetooth	HDMI	✓	12V–19V	12V
10, 23 via ATmega32u4, 2 via Grove	7 PWM, 12 ADC, 1 ADC via Grove	—	2 HDMI, 2 USB-C DP	✓	19V	3.3V

NEW & NOTABLE

NVIDIA JETSON XAVIER NX

The Jetson Xavier NX offers seriously impressive performance. Where other devices may allow you to handle one AI task at a time, the Xavier NX is powerful enough to run multiple such tasks. This makes it a great device for robotics projects that navigate around a room while also identifying humans and responding to their commands.

HACKBOARD 2

Most maker-focused SBCs run Linux, while only a few offer Windows support. New to the scene, Hackboard 2 goes all in on that with Windows 10 running on a 2.8GHz dual-core Intel Celeron N4020. Configurable options allow for up to 8GB of RAM and 512GB of storage. It's capable of being a desktop computer, but the 40-pin GPIO still let you tinker away. (And don't worry, it'll do Linux too.)

Make: GUIDE TO BOARDS 2021

SINGLE-BOARD COMPUTERS (SBC)

Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
UDOO Neo Full	$71	3.5"×2.32"	UDOObuntu2 14.04, Android 6	1GHz Cortex-A9, 227MHz Cortex-M4	32-bit Freescale i.MX 6SoloX (single-core Cortex-A9 CPU, Cortex-M4 coprocessor), Vivante GC420 GPU	1GB RAM
UDOO X86 II Ultra	$294	4.72"×3.35"	Windows 8.1/10, any Linux distribution, Android-x86	1.6GHz–2.56Hz CPU, 700MHz GPU	64-bit Intel Pentium N3710 (quad-core x86-64, HD Graphics 405), ATmega32u4 coprocessor	8GB DDR3L RAM, 32GB eMMC
VoCore 2	$18	1.0"×1.0"	OpenWRT	580MHz	32-bit Mediatek MT7628AN (single-core MIPS 24K)	128MB DDR2 RAM, 16MB flash

FIELD-PROGRAMMABLE GATE ARRAY BOARDS (FPGA)

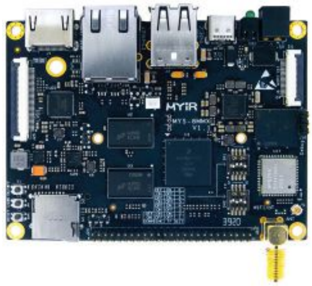
Board Name	Price	Dimensions	Software	Clock Speed	Processor	Memory
Alchitry Au	$100	2.56"×1.77"	Alchitry Labs, Vivado	100MHz oscillator on board	Xilinx Artix 7 XC7A35T-1FTG256C	256MB DDR3 RAM, 32MB configuration flash; FPGA: 225kB block RAM, 33,280 logic cells
Alchitry Cu	$50	2.56"×1.77"	Alchitry Labs, IceCube2, IceStorm	100MHz oscillator on board	Lattice iCE40HX8K-CB132	32MB configuration flash; FPGA: 16kB block RAM, 7,680 logic cells
Arduino MKR Vidor 4000	$72	3.27"×1"	Intel Quartus, Arduino IDE (for Cortex-M0+ only)	48MHz–200MHz FPGA, 48MHz Cortex-M0+	Intel Cyclone 10CL016, 32-bit ATSAMD21 (single-core Cortex-M0+) coprocessor	256kB flash, 32kB SRAM, FPGA 2MB flash, 8MB SDRAM, 16,000 logic cells
Arty Z7-10	$169	3.46"×4.3"	Vivado, Xilinx SDK, Vitis	650MHz Cortex-A9, 125MHz external clock	Xilinx Zynq-7000 XC7Z010-1CLG400C with dual-core Cortex-A9 CPU	16MB QSPI flash, 512 MB DDR3 RAM; FPGA: 270kB block RAM, 28,000 logic cells
Fomu	$50	0.38"×0.5"	Yosys/Nextpnr	12MHz VexRISC-V, 48MHz external oscillator	Lattice iCE40UP5K with VexRISC-V core	2MB flash, 128kB SRAM, 5,280 logic cells
Minized	$89	3"×2.8"	Vivado, Xilinx SDK, Vitis	667MHz, 33.33MHz external oscillator	Xilinx Zynq XC7Z007S with single-core Cortex-A9 CPU	512MB DDR3L RAM, 16MB flash, 8GB eMMC; FPGA: 225kB block RAM, 23,000 logic cells
Seeed Spartan Edge Accelerator	$40	2.09"×2.7"	ivado, Arduino IDE (ESP32 Only)	100MHz	Xilinx Spartan-7 XC7S15-1FTGB196C FPGA, spressif ESP32 coprocessor	4MB flash, FPGA 360kB block RAM, 12,800 logic cells
Snickerdoodle Black	$245	2.0"×3.5"	Vivado, Xilinx SDK, Vitis, PYNQ	866MHz	Xilinx Zynq XC7Z020-3CLG400E with dual-core Cortex-A9 CPU	16MB flash, 256kB SRAM, 1GB LPDDR2 RAM; FPGA :630kB block RAM, 85,000 logic cells
Snickerdoodle One	$115	2.0"×3.5"	Vivado, Xilinx SDK, Vitis, PYNQ	667MHz	Xilinx Zynq XC7Z010-1 with dual-core Cortex-A9 CPU	16MB flash, 256kB SRAM, 512MB LPDDR2 RAM; FPGA: 270kB block RAM, 28,000 logic cells
TinyFPGA AX2	$19	1.2"×0.7"	Lattice Diamond	133MHz	Lattice LCMX02-1200	8kB flash, 8kB RAM, 1.25kB block RAM, 1,280 logic cells
TinyFPGA BX	$39	1.4"×0.7"	IceStorm, APIO-IDE	16MHz external oscillator	Lattice iCE40LP8K	1MB flash, 16kB block RAM, 7,680 logic cells
WebFPGA	$38	2.2"×0.9"	WebFPGA, IceStorm	120MHz, 16MHz external oscillator, 48MHz (STM32F04)	Lattice iCE40UP5k, STMicro STM32F04 coprocessor	32kB flash, 6kB RAM, 16MB ext. flash; FPGA: 128kB SRAM, 15kB block RAM, 5,280 logic cells
Zynqberry	$127	3.4"×2.2"	Vivado	667MHz, 33.33MHz oscillator	Xilinx Zynq XC7Z010-1CLG225C with dual-core Cortex-A9 CPU	16MB flash, 512MB DDR3L RAM; FPGA: 270kB block RAM, 28,000 logic cells
Zynqberry Zero	$115	2.56"×1.18"	Vivado	667MHz, 33.33MHz oscillator	Xilinx Zynq XC7Z010-1CLG225C with dual-core Cortex-A9 CPU	16MB flash, 512MB DDR3L RAM: FPGA: 270kB block RAM, 28,000 logic cells

• Reviews by Mel Ho, Chris Yohe, Cabe Atwell, Paul J. Henley, Whitney Knitter, Kelly Egan, and Mike Senese •

Digital Pins	Analog Pins	Radio	Video	Ethernet On Board	Input Voltage	Operating Voltage
32, 22 via Cortex-M4	8 PWM, 6 ADC	Wi-Fi, Bluetooth	Micro-HDMI, LVDS LCD	✓	5V–15V	3.3V
23	7 PWM, 12 ADC	–	HDMI, 2 mDP++	✓	12V	3.3V
40	4 PWM	Wi-Fi	–	✓ Via pins	3.6V–5.5V	3.3V

Digital Pins	Analog Pins	Radio	Video	Ethernet On Board	Input Voltage	Operating Voltage
111	9 ADC	–	–	–	5V	3.3V and 1.8V
79	–	–	–	–	5V	3.3V
22, 25 via Mini PCI Express header	22 PWM, 25 via mPCIe, 13 via M0+, 7 ADC, 1 DAC	Wi-Fi, Bluetooth	Micro-HDMI, LVDS	–	5V	3.3V
65	6 XADC, 4 Differential XADC	–	HDMI	✓	7V–15V	3.3V
4	–	–	–	–	5V	3.3V
38	6 ADC	Wi-Fi, Bluetooth	–	–	5V	3.3V
20, 10 in Shield Mode	20 PWM, 1 ADC; 10 PWM, 1 ADC in Shield Mode	Wi-Fi, Bluetooth	Mini HDMI	–	5V–17V	5V
180	16 ADC, 2 DAC	Wi-Fi, Bluetooth, BLE	–	✓ Via pins	3.7V–17V	1.8V–3.3V
155	16 ADC, 2 DAC	Wi-Fi	–	✓ Via pins	3.7V–17V	1.8V–3.3V
21	–	–	–	–	3.3V	3.3V
41	–	–	–	–	5V	3.3V
32	–	–	–	–	5V	3.3V
26	–	–	HDMI, MIPI DSI	✓	5V	1.8V–3.3V
26	–	–	Mini HDMI	–	5V	1.8V–3.3V

NEW & NOTABLE



MYIR MYS-8MMX

Looking into embedded projects? This quick, power-efficient Linux machine should be on your list. Built around NXP's powerful new i.MX 8 Mini processor, this 1.8GHz quad-core SBC brings all the goodies: gigabit ethernet; MicroSD and M.2 slots; HDMI; Wi-Fi and Bluetooth; LVDS, QSPI, and MIPI interfaces; and lots more.

SEEED SPARTAN EDGE ACCELERATOR

FPGA development boards can get expensive quickly, especially if matching all of the peripherals that the Spartan Edge Accelerator has for wireless, high-speed image processing, and user I/O. Targeting a wide range of developers from Arduino, to IoT, to FPGA, this is a good tool for any hobbyist, maker or engineer to have on hand.