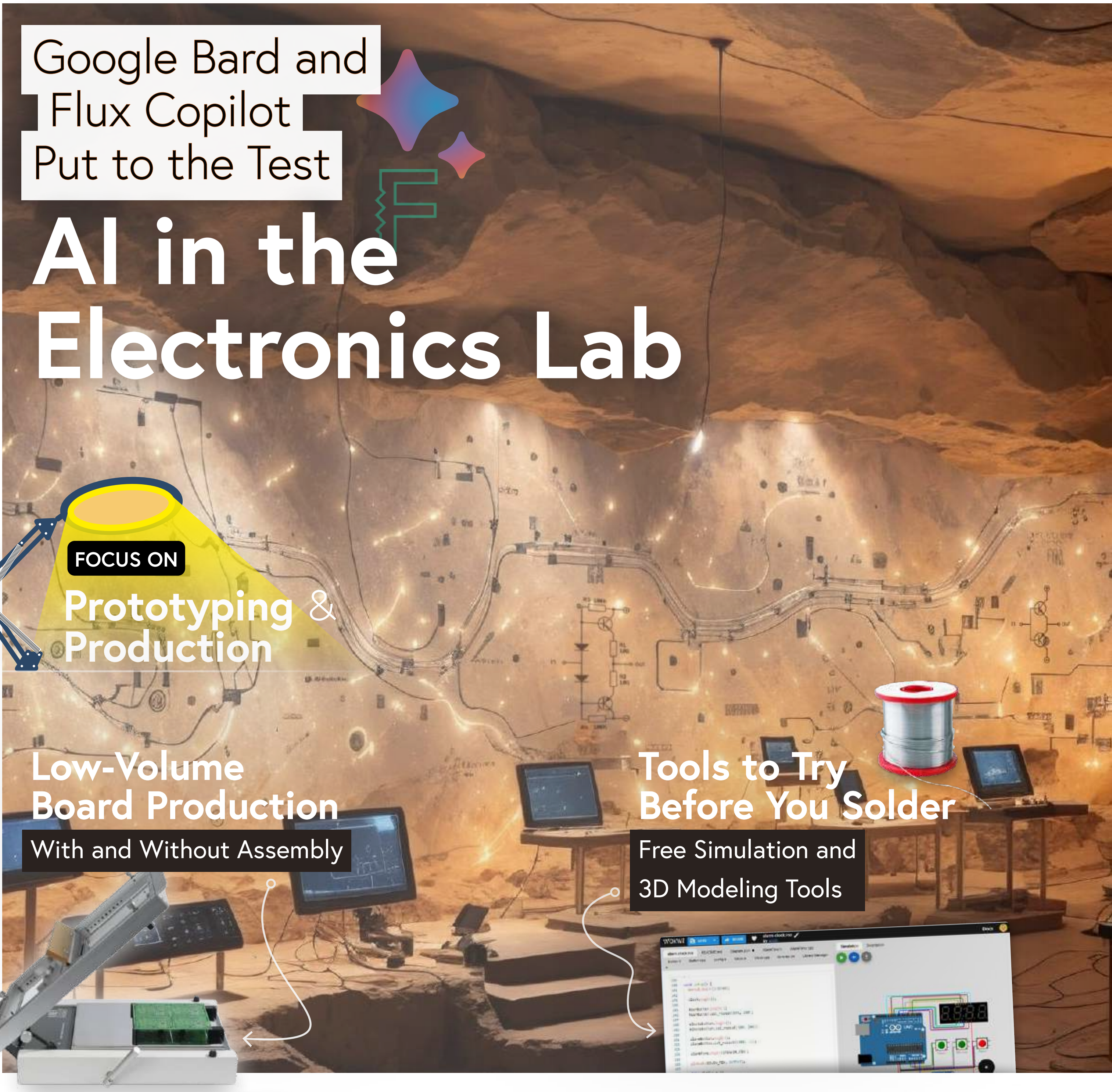
£9.95 | 10.95 € (D) | 13.50 € (other EU-Contries) | $12.95

elektor MAG
design > share > earn

525
NOVEMBER & DECEMBER 2023
ELEKTORMAGAZINE.COM
* SINCE 1961 *



Google Bard and
Flux Copilot
Put to the Test

AI in the Electronics Lab

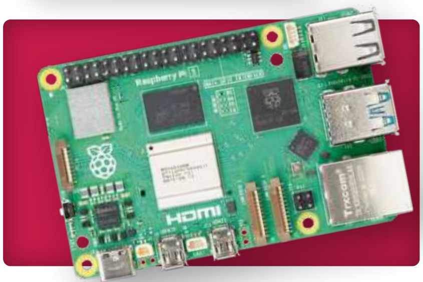
FOCUS ON
Prototyping & Production

Low-Volume Board Production

With and Without Assembly

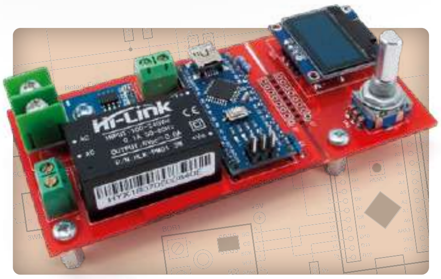
Tools to Try Before You Solder

Free Simulation and 3D Modeling Tools



The Raspberry Pi 5
A Huge Improvement From Its Predecessor
p. 6

MEMS Microphone
Superb Performance at a Low Cost
p. 70



DIY Reflow Plate
For Assembling or Repairing Small SMD Circuits
p. 120

ISSN 1757-0875
9 771757 087002 12

Innovative humans have been pondering the potential of AI for centuries. Now it's time to put the technology to the test.

DEBUGGING
MADE EASY

Discover, configure, develop, debug, qualify and go to market quickly using Microchip's development tools for PIC®, AVR® and SAM microcontrollers, SAM microprocessors and dsPIC® Digital Signal Controllers. We have integrated our entire line of embedded software development tools with our embedded hardware development tools so you can harness the full capabilities of the MPLAB® development ecosystem to get a best-in-class experience.

PICkit™ 5

The MPLAB® PICkit 5 in-circuit debugger/programmer is a powerful tool that provides quick prototyping and production-ready programming for all Microchip devices. This versatile device is equipped with a range of features that make it the perfect companion for your programming needs.

www.microchip.com/pickit5

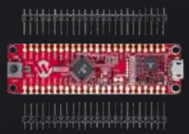
MPLAB® ICD 5

The MPLAB ICD 5 is designed to build on the strengths of our mid-range debugger and offers instrumented trace (ARM SWO) capabilities. With support for all PIC®, dsPIC®, SAM, and AVR® devices, our latest product provides comprehensive support for a wide range of devices.

www.microchip.com/icd5

Works with these boards and many more:

PIC32MK MCA
Curiosity Pro
Development
Board



SAM D21
Curiosity Nano
Evaluation Kit

AM E54
Curiosity Ultra
Development Board

PIC32CM JH01
Curiosity Pro
Development
Board

PIC16F15244
Curiosity Nano
Evaluation Kit

EV72Y42A - AVR64DD32
Curiosity Nano
Evaluation Kit

PIC32CX SG61
Curiosity Ultra
Evaluation Board

PIC32CM MC00
Curiosity Pro
Evaluation Kit

More info: www.microchip.com/mplab

THIS EDITION

Volume 49, No. 525
November & December 2023

AI in the Electronics Lab
Google Bard and Flux Copilot Put to the Test 10

Arduino Nano Waveform Generator
Nano + Code = Function Generator 20

Regulars

4 **Colophon**

116 **Starting Out in Electronics...**
...Amplifying Differences

130 **2023: An AI Odyssey**
Getting Help Designing a Physical Project

134 **Ethics in Action**
Brussels Is Innovating

138 **Hexadoku**
The Original Elektorized Sudoku

46 **IoT Simulation Simplified with Wokwi**
Developer Uri Shaked on Design, Software, and More

54 **A Bare-Metal Programming Guide (Part 3)**
CMSIS Headers, Automatic Testing, and a Web Server

127 **Don't Start with a Prototype — Start with a Pretotype!**
Check That a Market Exists for Your Product

Features

6 **The Raspberry Pi 5**
A Huge Improvement From Its Predecessor

10 **AI in the Electronics Lab**
Google Bard and Flux Copilot Put to the Test

34 **A Simple CNCed Enclosure**
With Autodesk Fusion 360 for Personal Use

40 **Low-Volume Board Production**
With and Without Assembly

Projects

20 **Arduino Nano Waveform Generator**
Nano + Code = Function Generator

28 **Solar-Powered Christmas Garland**
An Eco-Friendly Solution for Garnishing Your Balcony

32 **USB Killer Detector**
Better Safe Than Sorry

64 **LoRa, a Swiss Army Knife (2)**
The Hardware and Software

70 **MEMS Microphone Design and Construction**

120 **Mini Reflow Plate**
For Assembling or Repairing Small SMD Circuits

COLOPHON

Volume 49, No. 525
November & December 2023
ISSN 1757-0875

Elektor Magazine is published 8 times a year by
Elektor International Media b.v.
PO Box 11, 6114 ZG Susteren, The Netherlands
Phone: +31 46 4389444

elektor.com | elektormagazine.com

For all your questions
service@elektor.com

Become a Member
elektormagazine.com/membership

Advertising & Sponsoring
Büsra Kas
Tel. +49 (0)241 95509178
busra.kas@elektor.com
elektormagazine.com/advertising

Copyright Notice
© Elektor International Media b.v. 2023

The circuits described in this magazine are for domestic and educational use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, digital data carriers, and article texts published in our books and magazines (other than third-party advertisements) are copyright Elektor International Media b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning and recording, in whole or in part without prior written permission from the Publisher. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The Publisher disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from schematics, descriptions or information published in or in relation with Elektor magazine.

Print
Senefelder Misset, Mercuriusstraat 35,
7006 RK Doetinchem, The Netherlands

Distribution
IPS Group, Carl-Zeiss-Straße 5
53340 Meckenheim, Germany
Phone: +49 2225 88010

PEFC Certified
This product is from sustainably managed forests and controlled sources
PEFC
PEFC/30-31-151
www.pefc.org

USB Killer Detector
Better Safe Than Sorry
32

Simple CNCed Enclosure
With Autodesk Fusion 360
34

Industry

76 **Tools to Try Before You Solder**
Simulation and 3D Modeling Tools That Can Be Used for Free

80 **New Tools from Microchip!**
PICkit 5 and MPLAB ICD 5 Available Now!

86 **Rapid Prototyping of Flexible, Stretchable Electronics**

90 **Galvanic Isolation**
Using Phototransistor Optocouplers Successfully

93 **The Complex Solution or the Anybus Solution?**
Embedded Industrial Ethernet in 2 Days Rather Than Many Months

96 **Your Essential DFM Checklist**

100 **3D Printing Filaments**
Types, Features and Use in Prototyping

104 **Specialists for Effective Signal Analysis from ELF to EHF Band**

106 **Challenges of DFM Analysis for Flex and Rigid-Flex Design**

108 **Setting Up an SMT Line**

110 **The Right Combination for a Reliable Assembly**

112 **Revolutionizing Industries**
The Rise of Autonomous Mobile Robots (AMRs)

114 **Evolved for More Challenges**
The 8-Channel R&S MXO 5 by Rohde & Schwarz

Next Editions

Elektor & Espressif Guest Edition 2023
For the next edition, Elektor editors are teaming up with engineers from Espressif, the creator of the popular ESP8266 and ESP32 microcontrollers. As you would expect from Elektor, DIY projects are the heart of this edition, but, you'll also find in-depth tech tutorials and background articles about many topics, such as IoT protocols and cloud solutions, Rust, speech and music recognition, AI frameworks, prototyping, and many more.

Elektor & Espressif Guest Edition 2023 will be published around December 13th, 2023. Arrival of printed copies for Elektor Gold members is subject to transport.

Elektor Magazine January & February 2024
As usual, we'll have an exciting mix of projects, circuits, fundamentals and tips and tricks for electronics engineers and makers. We'll focus on Power & Energy.

Elektor Magazine January / February 2024 edition will be published around January 10th. Arrival of printed copies for Elektor Gold members is subject to transport.

The Team

International Editor-in-Chief: Jens Nickel | **Content Director:** C. J. Abate | **International Editorial Staff:** Asma Adhimi, Roberto Armani, Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf (RG), Ton Giesberts, Hedwig Hennekens, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-Francois Simon, Clemens Valens, Brian Tristam Williams | **Regular Contributors:** David Ashton, Tam Hanna, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Graphic Design & Prepress:** Harmen Heida, Sylvia Sopamena, Patrick Wielders | **Publisher:** Erik Jansen | **Technical questions:** editor@elektor.com

Jens Nickel
International Editor-in-Chief, Elektor Magazine

Even more ideas!

You know how it goes: You put in the hard work, and now your little project is up and running. You show it off to your electronics friends and they're impressed. Great stuff, right? It might even be a hit with other makers grappling with a similar problem, or have an application for people who aren't even into electronics. But, you'd need to invest in a more attractive enclosure, a user guide, and a more intuitive interface. Not to mention a way to roll out remote firmware updates.

Some very successful inventions in electronics started out just like this. But if you're looking to turn your project into a moneymaker, you'll need to consider a different approach. Check out the article on page 127, in which our interviewee, Alberto Savoia, explains why even the best ideas can be doomed from the start, and what you should sort out before building your first prototype.

When you're ready to dive into development, there's no shortage of — exactly what we expect today — AI tools ready to help. Starting on page 10, Tam Hanna reviews how well these tools assist in component selection and incorporating them into your schematic. By the time you read this, these tools will certainly have improved even further. Try them yourself now and email us with your feedback (editor@elektor.com)!

AI has also given our graphic artists some fresh inspiration. We used the Leonardo AI image generator to help create our cover. But believe me, finding the perfect idea didn't come easily; we debated multiple suggestions for two hours during our cover meeting. This just goes to show that AI isn't here to replace our creativity — it simply offers us more options than ever before.

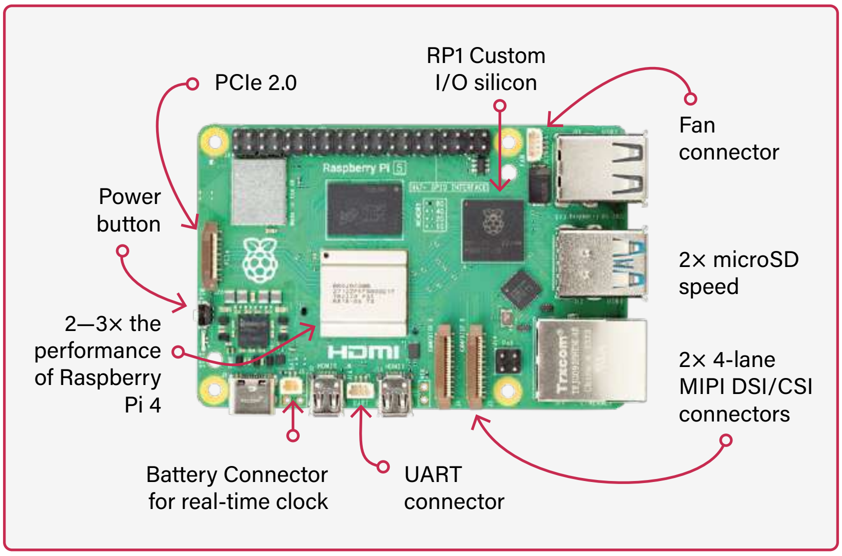
Let's all take advantage of this, shall we?

The Raspberry Pi 5

A Huge Improvement From Its Predecessor

By Saad Imtiaz (Elektor)

The all-new Raspberry Pi 5 is here, and one of the first questions that comes to mind is how much better and faster it is from its predecessor. Well, a lot has changed, and there are many features that have been added and improved. It can be easily said that the new Raspberry Pi 5 is twice as fast as the Raspberry Pi 4. In this article, we will go over some newly added features, benchmark comparisons, thermals and more.


PCIe 2.0
RP1 Custom I/O silicon
Fan connector
Power button
2× microSD speed
2—3× the performance of Raspberry Pi 4
2× 4-lane MIPI DSI/CSI connectors
Battery Connector for real-time clock
UART connector

The Raspberry Pi 5 has the same charming credit card size as previous generations, but a lot has changed while maintaining this size and format. Let's start with the new CPU on board, the Cortex-A76 CPU clocked at 2.4 GHz, and a new Video Core 7 GPU at 1 GHz. Raspberry Pi claimed that this will offer a considerable leap in performance, which it absolutely does: We'll dive into this later with some benchmarks.

RAM and More

Another change is the RAM; now the new Raspberry Pi 5 rocks DDR4X RAM instead of the DDR4 RAM used before. This enables the Raspberry Pi 5 to have improved clock speeds with less power consumption, which is a step in the right direction. When we ran some RAM speed tests on the Raspberry Pi 5, we had some remarkable results. The Raspberry Pi 5 was able to perform close to 30,000 MB writes and reads per second as compared to the Raspberry Pi 4, which could only do 4,000 to 6,000.

Another major improvement is the microSD card slot with support for high-speed SDR104 mode. In our testing, the SD card speeds were twice as fast as the last generation. This is thanks to the new RP1 southbridge developed by Raspberry Pi.

The Raspberry Pi 5 is the first full-size Raspberry Pi computer that uses silicon developed in-house. The RP1 southbridge provides improved performance in I/O capabilities and functionality that includes doubled USB bandwidth, 2×4-lane MIPI interface support, and a single-lane PCI Express 2.0 interface.

PCI, Cameras, Displays...

One major addition to the Raspberry Pi 5 is the single-lane Gen 2 PCIe connection, which makes the Raspberry Pi 5 capable of using an NVMe or an M.2 SSD now, with the potential for much faster read and write speeds as well and much-improved performance. This also allows us to connect high-bandwidth peripherals.

One thing that didn't change is the Wi-Fi and Ethernet: They're the same as on the last version, and it still works as it should.

The differences, however, are what we're interested in, so let's go over everything briefly.

There are two camera/display connectors, so you can have two cameras connected to your Raspberry Pi 5 or a camera and a display, or even two displays.

On board is a new fan connector, which can control the speed of the fan that covers the heat sink on the Raspberry Pi. In addition to the fan connector, we now have a UART connector, a power button to reboot or shutdown, much-improved speeds on the USB 3.0 connectors (reaching up to 5.1 Gbit/s), and an onboard real-time clock (RTC) that can be powered by an external coin cell battery. **Table 1** shows the complete list of specifications that differ from the Raspberry Pi 4.

Table 1: Comparison: Raspberry Pi 4 vs Raspberry Pi 5

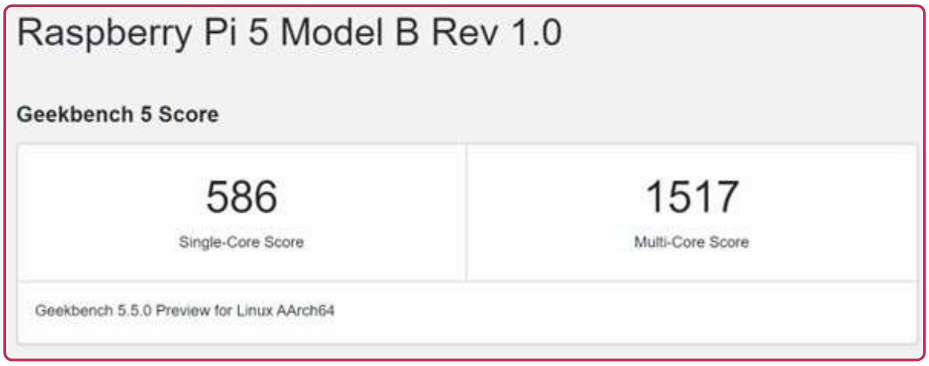
	Raspberry Pi 4	**Raspberry Pi 5**	
CPU	Broadcom BCM2711, Cortex-A72 (ARM v8) 64-bit SOC @ 1.8 GHz	**Broadcom BCN12712, quad-core Cortex-A76 (ARM v8), 64-bit SOC @ 2.4 GHz**	**2–3× performance**
RAM	1 GB, 2 GB, 4 GB, 8 GB	**1 GB, 2 GB, 4 GB, 8 GB**	
Connectivity	2.4 GHz and 5.0 GHz 802.11 ac wireless Bluetooth 5.0, BLE Gigabit Ethemet n/a 2× USB 3.0, 2× USB 2.0 ports Standard 40-pin GPIO header 2× Micro HMDI Ports (up to 4K 60p) 2-lane MIPI DSI, 2-lane MIPI CSI 4-pole stereo audio and composite video	2.4 GHz and 5.0GHz 802.11 ac wireless Bluetooth 5.0, BLE Gigabit Ethemet **1× PCIe 2.0 interface** **2× USB 3.0 (5 Gbit/s),** **2× USB 2.0 ports** Standard 40-pin GPIO header 2× Micro HDMI ports (up to 4K 60p) **2× 4-lane MIPI (DSI/CSI) n/a**	**High-speed peripheral interface (for SSDs, etc.)**
OS and data storage	microSD card slot	**microSD card slot with support for high-speed SDR104 mode**	**2× interface speed**
Input power	5 V DC @ 3 A (via USB-C connector or GPIO)	**5 V DC @ 5 A DC (PD-enabled)**	**New Raspberry Pi power supply**
PoE	Via separate PoE HAT	**Via (new) separate PoE HAT**	**Fully PoE 802.3at-compliant**
Real-time clock (RTC)	n/a	**RTC and RTC battery connector**	

Benchmarks

We did some benchmarking on the Raspberry Pi 5 [1] and the Raspberry Pi 4 [2] keeping the s setup and hardware the same — so, same build on the SD card, same power supply, and with and without a cooling solution. With the links below, you can take a deep dive into all of the benchmark scores from Geekbench 5. But, for the sake of easy, in **Figure 1** and **2**, you see the overall benchmark scores of the Raspberry Pi 5 and Raspberry Pi 4, which clearl indicate that the Raspberry Pi 5 is two to three times faster than its predecessor in all types of tests.

Thermals

As you know, in computing, the faster a CPU becomes in terms of performance, the more power it utilizes, and more power means more heat! We did some thermal imaging of the Raspberry Pi 5 and 4 under both load and idle conditions. We also tested the new active cooler provided by Raspberry Pi, seen in place in **Figure 3**.



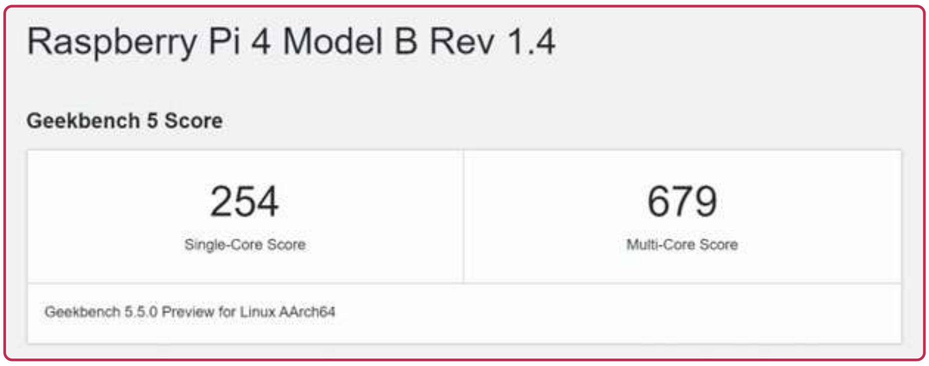
Raspberry Pi 5 Model B Rev 1.0

Geekbench 5 Score

586	1517
Single-Core Score	Multi-Core Score

Geekbench 5.5.0 Preview for Linux AArch64

Figure 1: Geekbench 5 benchmark scores for the Raspberry Pi 5, 8 GB.



Raspberry Pi 4 Model B Rev 1.4

Geekbench 5 Score

254	679
Single-Core Score	Multi-Core Score

Geekbench 5.5.0 Preview for Linux AArch64

Figure 2: Geekbench 5 benchmark scores for the Raspberry Pi 4, 8 GB.

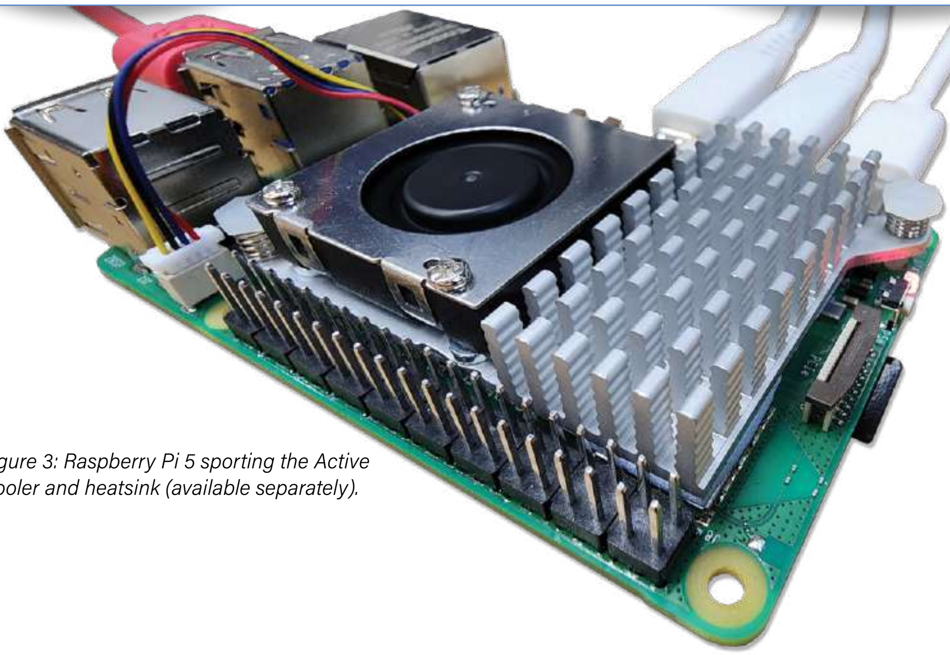


Figure 3: Raspberry Pi 5 sporting the Active Cooler and heatsink (available separately).

While running benchmarks and some CPU- and GPU-intensive tasks, the Raspberry Pi 5 almost reached 78°C (at the chip level), without cooling, in the test lab ambient temperature of 23°C. On the other hand, the Raspberry Pi 4 reached a maximum of 71°C (chip-level), without cooling, under the same load — but not with the same performance, of course. In **Figure 4** and **5**, thermal images of the Raspberry Pi 5 and Raspberry Pi 4 are shown. These temperatures are quite high, and it is recommended to use the cooling solution provided by Raspberry Pi. In **Figure 6**, we see the thermal image of the Raspberry Pi 5 with the active cooler installed, and it dropped the chip-level temperature to 58°C — a 20-degree difference with just the cooler! In future, we're really looking forward to any new thermal solutions and cases provided by third parties to improve the thermals even more.

Is It for You?

The Raspberry Pi 5 is a game-changer in the single-board computer world. Its impressive hardware upgrades, from its new CPU and GPU to its DDR4X RAM, make it a powerhouse in a credit-card-sized package. You'll have plenty of peripheral options, from multiple screens to dual cameras to M.2 SSDs. Unfortunately, the USB sockets are not USB-C, but we can hope for that in a future iteration.

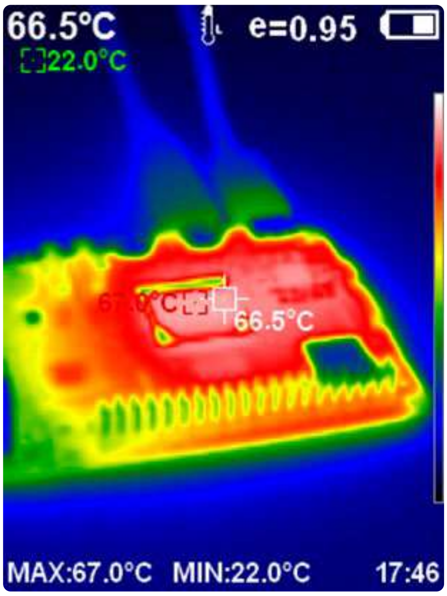


Figure 4: Raspberry Pi 5 thermal image — no active cooling.

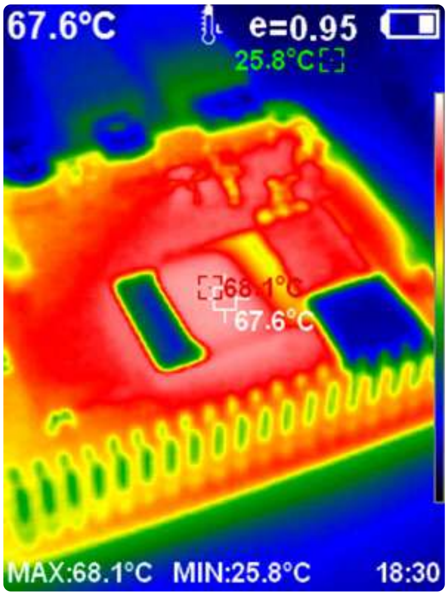


Figure 5: Raspberry Pi 4 thermal image — with active cooling.

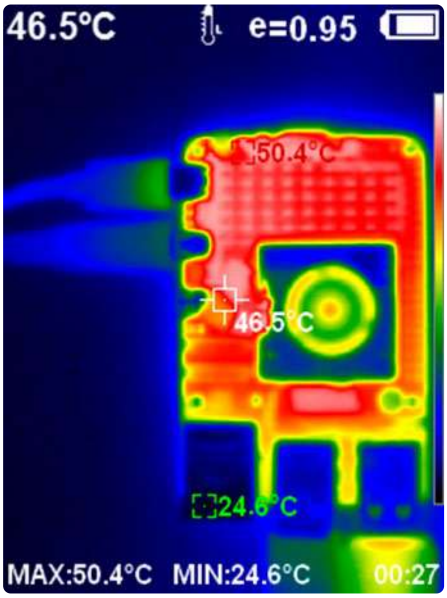


Figure 6: Raspberry Pi 5 thermal image — active cooling.

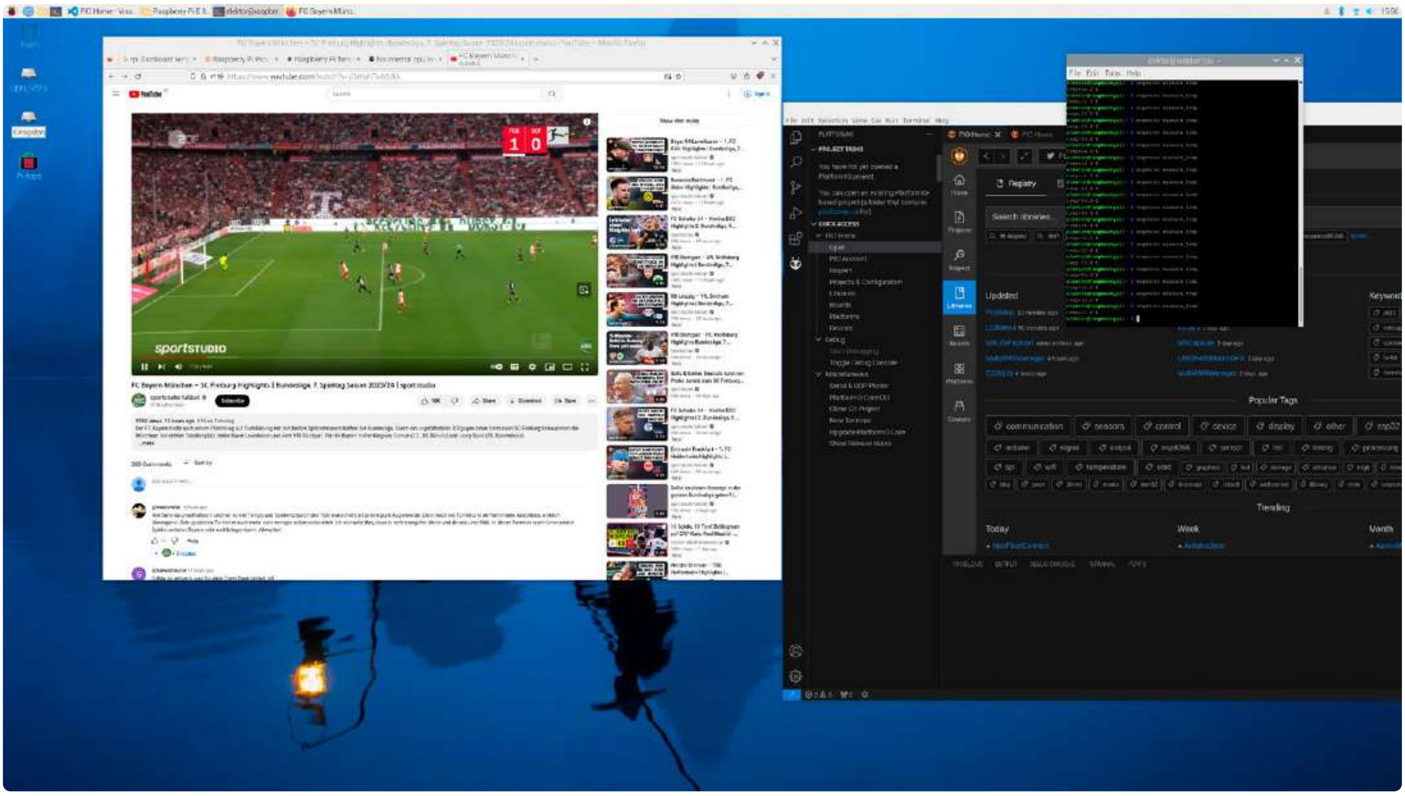


Figure 7: Raspberry Pi 5 running a normal workload for testing as a daily driver.

While it retains much of the charm and form factor that fans love, it doubles down on performance, making it a compelling offer for both newcomers and Raspberry Pi veterans. We tried to use it as an actual computer, with some normal daily workload, having a 1080p video playing on YouTube alongside some programming in VS Code for the ESP32 (**Figure 7**). It turns out it performs decently, and it can be used in a scenario where a laptop or computer is not available (use a power bank to power it up). Overall, it is a great addition to the Raspberry Pi family and the price, while higher, is still attractive.

If you're looking to get your hands on one, the Raspberry Pi 5 promises a whole new level of tinkering, development, and fun. ◀

230635-01

Questions or Comments?

If you have questions about this interview, feel free to e-mail the author at saad.imtiaz@elektor.com or the editorial team at editor@elektor.com.

Elektor's Raspberry Pi 5 Resources

The speedy Raspberry Pi 5, a must-have for pro engineers, makers, and students alike, serves as an ideal foundation. Whether your goal is constructing a desktop computer or deploying a smart robot, Elektor offers a wealth of articles and videos on this versatile board.

https://elektormagazine.com/rpi5

Related Products

- **Raspberry Pi 5 (4 GB RAM)**
  www.elektor.com/20598
- **Raspberry Pi 5 (8 GB RAM)**
  www.elektor.com/20599
- **Official Case for Raspberry Pi 5 (White)**
  www.elektor.com/20602
- **Active Cooler for Raspberry Pi 5**
  www.elektor.com/20604

WEB LINKS

[1] Raspberry Pi 4 Model B Rev 1.4 — Geekbench: https://browser.geekbench.com/v5/cpu/21819994
[2] Raspberry Pi 5 Model B Rev 1.0 — Geekbench: https://browser.geekbench.com/v5/cpu/21819518

AI in the Electronics Lab

Google Bard and Flux Copilot Put to the Test

By Tam Hanna (Hungary)

The capabilities of ChatGPT and similar AI systems have created considerable interest in popular media. It seems logical that these AI systems, which are designed to process general languages, should also be able to handle the relatively simple syntax of program code and circuit functions, and this syntax should be easier to formalize. Here we put two widely used systems — Google Bard and Flux Copilot — to a practical test.

For this test, I used Google Bard and Flux Copilot, and I would like to say that they do not pose any sort of threat to anyone working in electronics. Much like the advanced analysis functions introduced a while ago in digital storage oscilloscopes, the so-called Gatling paradox described by Chivers applies equally well to artificial intelligence. When a new technology (such as the Gatling gun) increases worker efficiency, a worker will be able to do more work, which means the colleagues can also do more. In the end, this leads to higher productivity.

As I consider it important to distinguish between those who do electronics and those who deal with electronics ethics, at this point I want to leave these thoughts aside and focus on how AI can help you do your job.

Google Bard

Particularly in today's highly charged political world, miscalculations by AI systems can potentially cause considerable damage to the reputations of the companies providing these systems. To counter this risk, Google has released its AI service, Bard, for general use, but has restricted its availability to a relatively small number of countries. Only a few days before this article was published, the list of allowed locations [1] was expanded to include Austria, Hungary, and Slovakia. It should also be noted that the Bard AI system is now able to speak German, but despite this, I worked in English in the examples described below. From practical experience, it can be seen that the results are better with English.

If you live in a country where Bard is not available, you needn't give up hope. Instead of using the account location, Google checks use authorization by continuously validating the geolocation of the IP address used for access. It's therefore advisable to use a VPN, which basically amounts to a scheme that presents a different IP address to fool the other party, as shown in **Figure 1**.

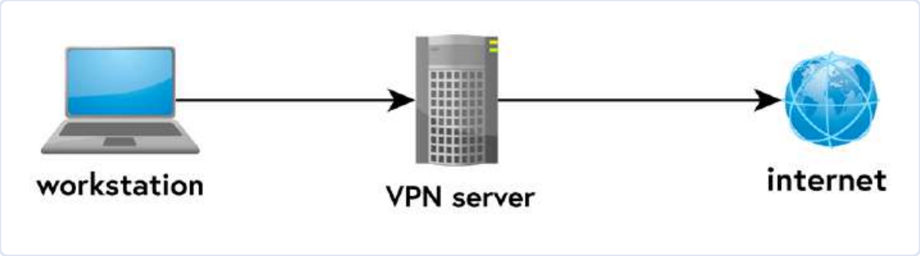
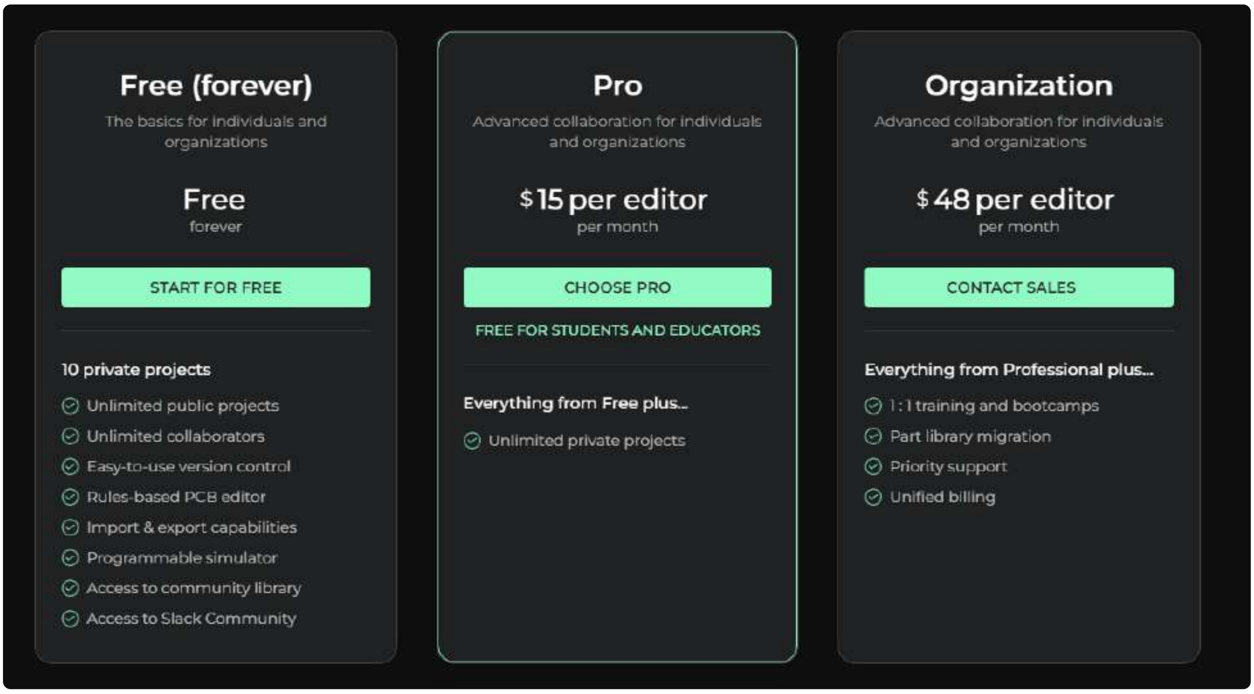


Figure 1: VPN services are now commercially available.

A difficulty with using a VPN service in conjunction with Google servers is that Google is an attractive target for bot exploiters and the like, so Google blocks many VPNs at the IP level. In my tests, Proton VPN generally worked without any problems, although in some cases some additional Captchas had to be clicked through to confirm that a human was trying to access Google.

The next thing you have to do is access the URL [2]. At the I/O Developers Conference, Google announced that, from now on, the



Free (forever)

The basics for individuals and organizations

Free

forever

10 private projects

- Unlimited public projects
- Unlimited collaborators
- Easy-to-use version control
- Rules-based PCB editor
- Import & export capabilities
- Programmable simulator
- Access to community library
- Access to Slack Community

Pro

Advanced collaboration for individuals and organizations

$15 per editor

per month

FREE FOR STUDENTS AND EDUCATORS

Everything from Free plus...

- Unlimited private projects

Organization

Advanced collaboration for individuals and organizations

$48 per editor

per month

Everything from Professional plus...

- 1 : 1 training and bootcamps
- Part library migration
- Priority support
- Unified billing

Figure 2: You only have to pay to use Flux if you want to use it in Private Mode.

Bard service would be available without a waitlist. When I logged in for the first time a few weeks ago, I spent only a few hours on the waitlist, but it should be noted that wait times of several days are being reported.

Flux Copilot

While Google's Bard presents itself as a general-purpose AI service that can answer a wide range of questions, Flux Copilot [3] is designed as an AI service optimized for the needs of people working with electronics.

Actually, Flux is a complete browser-based EDA package, including circuit design, simulation, and PCB routing. The AI functions have simply been integrated on the side. It should be noted that the open-source version of Flux is free. As can be seen from **Figure 2**, you only have to pay when you want to keep projects that are managed in Flux largely confidential.

Ten Free Projects

At the time of publication, the completely free version of Flux includes 10 private projects. However, in practice, we see that free perks like this tend to disappear faster than we like.

In the examples described below, I used the free, basic version. Along with a fully fledged account, you can also use Gmail or GitHub to verify your identity with respect to the Flux servers. An interesting detail in this regard is that if you use Gmail, you have to specify a local username.

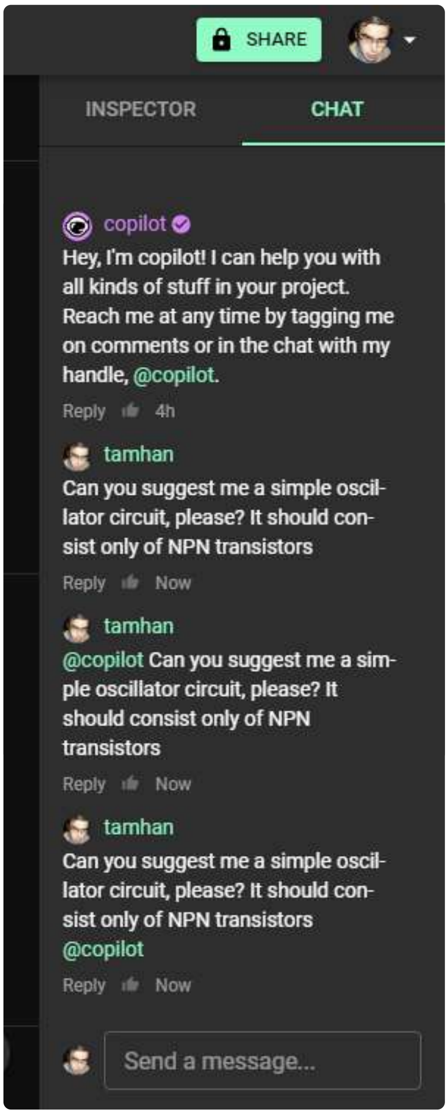
To actually start using Flux, you only have to click the green *Create Project* button. Copilot normally lives in the *Chat* pane, as shown in **Figure 3**, where it waits to be called.

Artificial Intelligence as a Lab Assistant

An old saying goes that knowing where to find information is 80% of the solution. In this regard, getting the document is often only a small part of the job. This applies equally well to working with electronic components. A distributor such as Mouser lists thousands of components in some categories, and hundreds of semiconductor manufacturers are trying to use dedicated special-function ICs to worm their way into the developer community. Logically enough, AI systems designed specifically for data processing could be helpful in this regard, so the first things I tried were to ask both systems to get information and to task them with designing a simple circuit.

Searching for an Astable Multivibrator

The first thing I wanted to try, in the spirit of the *Electronics for Dummies* series of books, was to search for an astable multivibrator. In theory, Google Bard should be perfectly suited for this task, since one of the major announcements at the I/O Developers



copilot

Hey, I'm copilot! I can help you with all kinds of stuff in your project. Reach me at any time by tagging me on comments or in the chat with my handle, @copilot.

Reply 4h

tamhan

Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors

Reply Now

tamhan

@copilot Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors

Reply Now

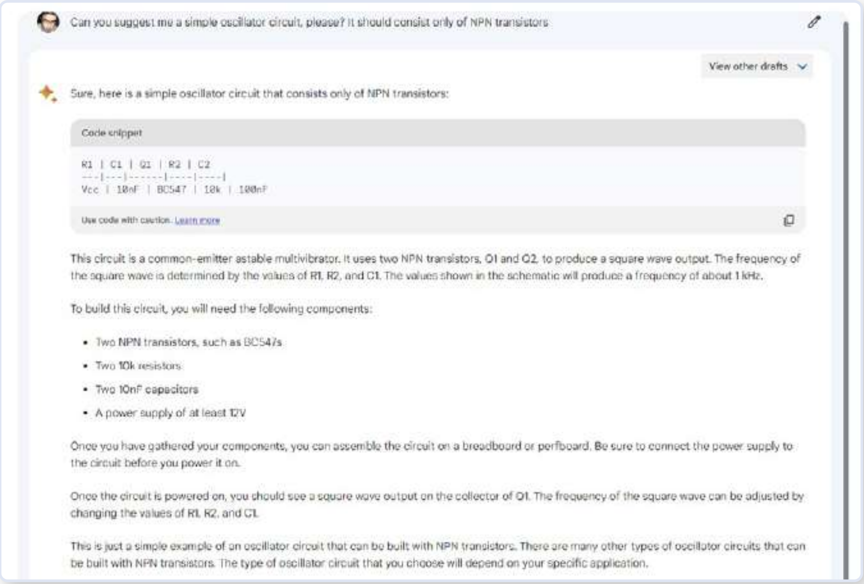
tamhan

Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors @copilot

Reply Now

Send a message...

Figure 3: Flux integrates its in-house AI assistants into a pane of the EDA tool.



Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

```
R1 | C1 | Q1 | R2 | C2
---|---|------|----|----|
Vcc | 10nF | BC547 | 10k | 100nF
```

This circuit is a common-emitter astable multivibrator. It uses two NPN transistors, Q1 and Q2, to produce a square wave output. The frequency of the square wave is determined by the values of R1, R2, and C1. The values shown in the schematic will produce a frequency of about 1 kHz.

To build this circuit, you will need the following components:

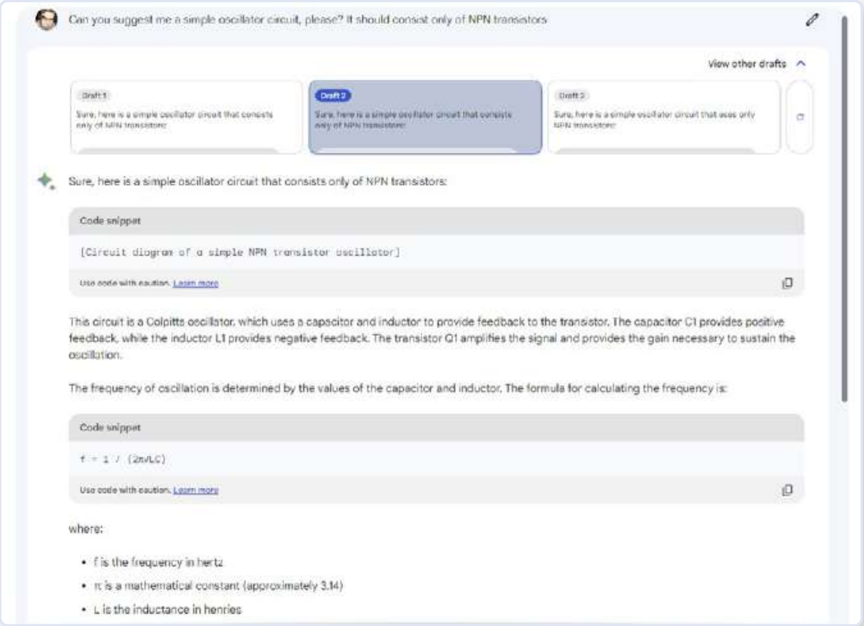
- Two NPN transistors, such as BC547s
- Two 10k resistors
- Two 10nF capacitors
- A power supply of at least 12V

Once you have gathered your components, you can assemble the circuit on a breadboard or perfboard. Be sure to connect the power supply to the circuit before you power it on.

Once the circuit is powered on, you should see a square wave output on the collector of Q1. The frequency of the square wave can be adjusted by changing the values of R1, R2, and C1.

This is just a simple example of an oscillator circuit that can be built with NPN transistors. There are many other types of oscillator circuits that can be built with NPN transistors. The type of oscillator circuit that you choose will depend on your specific application.

Figure 4: Only Bard was able to handle searching for an astable multivibrator.



Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

Sure, here is a simple oscillator circuit that uses only NPN transistors:

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

```
[Circuit diagram of a simple NPN transistor oscillator]
```

This circuit is a Colpitts oscillator, which uses a capacitor and inductor to provide feedback to the transistor. The capacitor C1 provides positive feedback, while the inductor L1 provides negative feedback. The transistor Q1 amplifies the signal and provides the gain necessary to sustain the oscillation.

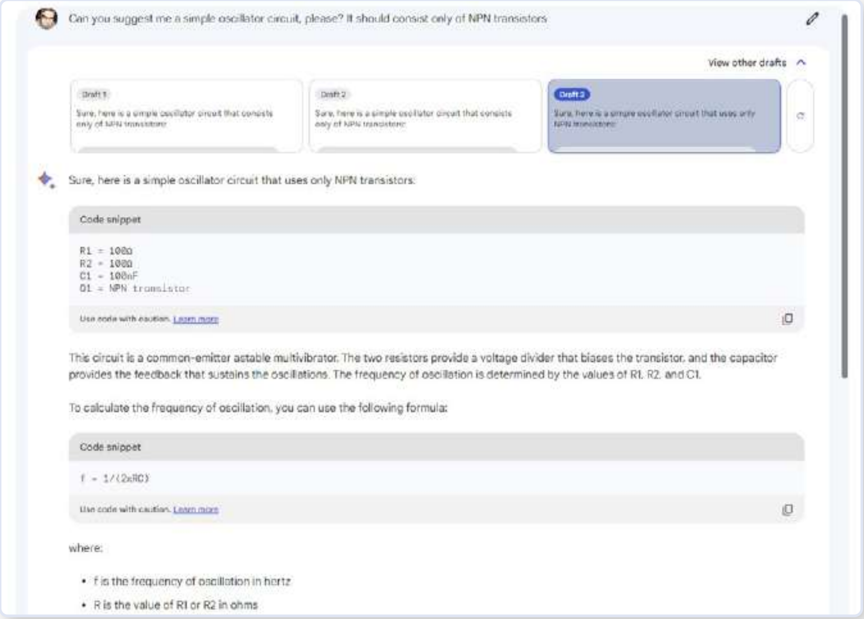
The frequency of oscillation is determined by the values of the capacitor and inductor. The formula for calculating the frequency is:

```
f = 1 / (2π√LC)
```

where:

- f is the frequency in hertz
- π is a mathematical constant (approximately 3.14)
- L is the inductance in henries

Figure 5: Bard provided detailed information for both the second result...



Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

Sure, here is a simple oscillator circuit that consists only of NPN transistors:

Sure, here is a simple oscillator circuit that uses only NPN transistors:

Sure, here is a simple oscillator circuit that uses only NPN transistors:

```
R1 = 100Ω
R2 = 100Ω
C1 = 100nF
Q1 = NPN transistor
```

This circuit is a common-emitter astable multivibrator. The two resistors provide a voltage divider that biases the transistor, and the capacitor provides the feedback that sustains the oscillations. The frequency of oscillation is determined by the values of R1, R2, and C1.

To calculate the frequency of oscillation, you can use the following formula:

```
f = 1/(2πRC)
```

where:

- f is the frequency of oscillation in hertz
- R is the value of R1 or R2 in ohms

Figure 6: ...and the third result.

Conference was that from now on, Bard would be able to return pictures in response to questions.

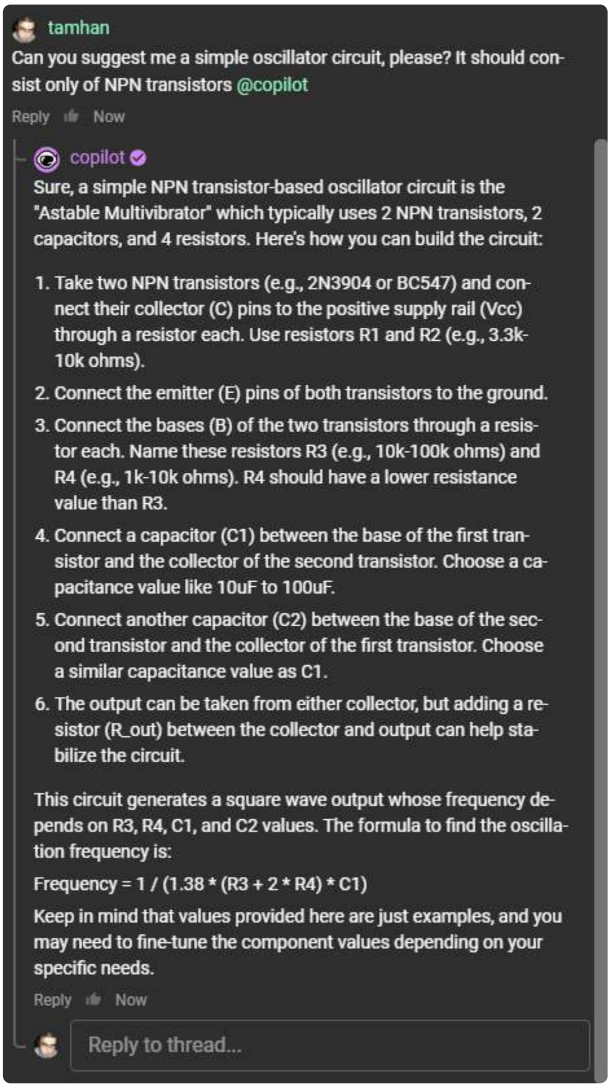
However, my request yielded only the result shown in **Figure 4**, which was not especially satisfactory due to the lack of a schematic diagram. After all, the *Google It* button at the bottom of the screen takes me to a list of frequent requests, and, in combination with the provided keyword, I can use these to find a usable schematic sooner or later. The bottom line is therefore that Bard offloads part (if only a small part) of the documentation search.

In this connection, it is interesting to note that Bard offered several versions of the schematic or answer, as can be seen in **Figure 5** and **Figure 6**. One of these versions also contained a classical image placeholder. A striking aspect of both answer versions is that they provide detailed information about the formulas necessary to calculate the operating frequency.

Next, I wanted to see how Flux Copilot would deal with the same task. The first difficulty turned out to be correctly formulating the request. If you don't make sure to tag Copilot as shown in Figure 3, you won't be able to activate its artificial intelligence. It took three requests before I got the result shown in **Figure 7**. Just like Bard, the answer from Copilot is limited to a text description of the circuit (albeit a detailed description). Here, as well, I searched in vain for a ready-to-use schematic.

Copilot sometimes took a while to respond to requests. However, the service now displays the message shown in **Figure 8**, which reports on the active thinking process of the AI tool.

Flux.ai updated the AI engine a few days before publication of this article. The updated version delivered the schematic of a Colpitts oscillator, as shown in **Figure 9**. My request to add the new components to an existing circuit design was acknowledged, but the new components did not appear in the schematic view. After I added them manually, the service was able to wire them without any problems. At the time of



tamhan

Can you suggest me a simple oscillator circuit, please? It should consist only of NPN transistors @copilot

copilot

Sure, a simple NPN transistor-based oscillator circuit is the "Astable Multivibrator" which typically uses 2 NPN transistors, 2 capacitors, and 4 resistors. Here's how you can build the circuit:

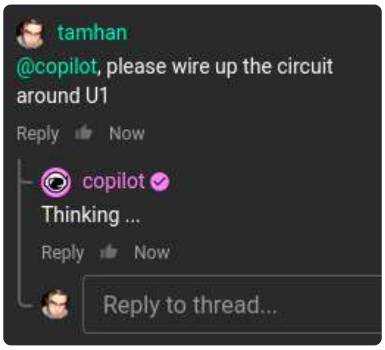
1. Take two NPN transistors (e.g., 2N3904 or BC547) and connect their collector (C) pins to the positive supply rail (Vcc) through a resistor each. Use resistors R1 and R2 (e.g., 3.3k-10k ohms).
2. Connect the emitter (E) pins of both transistors to the ground.
3. Connect the bases (B) of the two transistors through a resistor each. Name these resistors R3 (e.g., 10k-100k ohms) and R4 (e.g., 1k-10k ohms). R4 should have a lower resistance value than R3.
4. Connect a capacitor (C1) between the base of the first transistor and the collector of the second transistor. Choose a capacitance value like 10uF to 100uF.
5. Connect another capacitor (C2) between the base of the second transistor and the collector of the first transistor. Choose a similar capacitance value as C1.
6. The output can be taken from either collector, but adding a resistor (R_out) between the collector and output can help stabilize the circuit.

This circuit generates a square wave output whose frequency depends on R3, R4, C1, and C2 values. The formula to find the oscillation frequency is:

Frequency = 1 / (1.38 * (R3 + 2 * R4) * C1)

Keep in mind that values provided here are just examples, and you may need to fine-tune the component values depending on your specific needs.

Figure 7: You need a bit of natural intelligence to use the Copilot instructions.



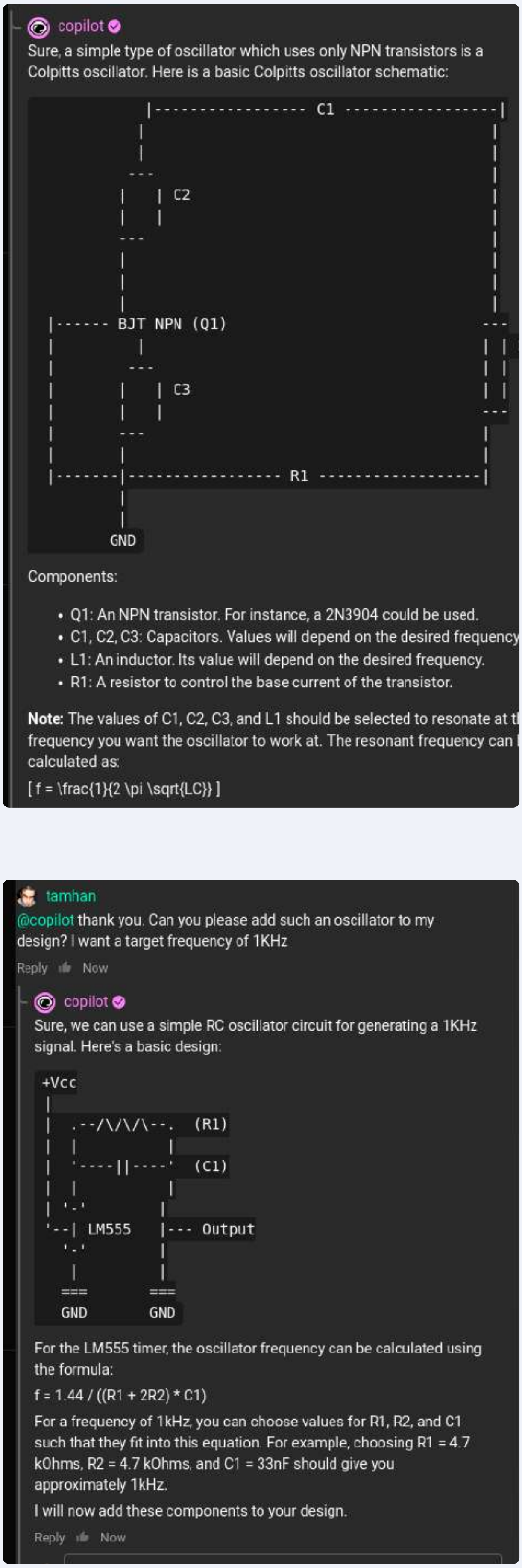
tamhan

@copilot, please wire up the circuit around U1

copilot

Thinking ...

Figure 8: This informs you of the AI tool's active thinking process.



copilot

Sure, a simple type of oscillator which uses only NPN transistors is a Colpitts oscillator. Here is a basic Colpitts oscillator schematic:

```
            |---------------- C1 ----------------|
            |                                    |
            |                                    |
           ---                                   |
          |   | C2                               |
          |   |                                  |
          ---                                    |
          |                                      |
          |                                      |
          |                                      |
  |------ BJT NPN (Q1)                          ---
  |         |                                   | |
  |        ---                                  | |
  |       |   | C3                              | |
  |       |   |                                 | |
  |       ---                                   ---
  |       |                                     |
  |       |                                     |
  |-------|---------------- R1 -----------------|
          |
          |
         GND
```

Components:

- Q1: An NPN transistor. For instance, a 2N3904 could be used.
- C1, C2, C3: Capacitors. Values will depend on the desired frequency
- L1: An inductor. Its value will depend on the desired frequency.
- R1: A resistor to control the base current of the transistor.

Note: The values of C1, C2, C3, and L1 should be selected to resonate at th frequency you want the oscillator to work at. The resonant frequency can b calculated as:

[f = \frac{1}{2 \pi \sqrt{LC}}]

tamhan

@copilot thank you. Can you please add such an oscillator to my design? I want a target frequency of 1KHz

copilot

Sure, we can use a simple RC oscillator circuit for generating a 1KHz signal. Here's a basic design:

```
+Vcc
 |
 |  .--/\/\/\--.  (R1)
 |  |          |
 |  '----||----'  (C1)
 |  |          |
 | '-'        |
 '--| LM555   |--- Output
   '-'        |
    |         |
   ===       ===
   GND       GND
```

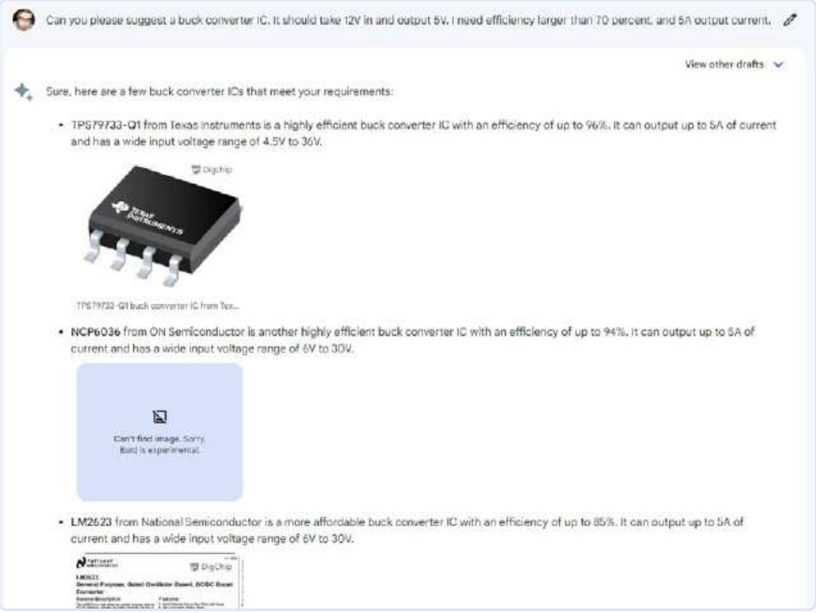
For the LM555 timer, the oscillator frequency can be calculated using the formula:

f = 1.44 / ((R1 + 2R2) * C1)

For a frequency of 1kHz, you can choose values for R1, R2, and C1 such that they fit into this equation. For example, choosing R1 = 4.7 kOhms, R2 = 4.7 kOhms, and C1 = 33nF should give you approximately 1kHz.

I will now add these components to your design.

Figure 9: Copilot's proposed schematic diagram for a Colpitts oscillator on its own, and integrated into my circuit.


Can you please suggest a buck converter IC. It should take 12V in and output 5V. I need efficiency larger than 70 percent, and 5A output current.

Sure, here are a few buck converter ICs that meet your requirements:

- **TPS79733-Q1** from Texas Instruments is a highly efficient buck converter IC with an efficiency of up to 96%. It can output up to 5A of current and has a wide input voltage range of 4.5V to 36V.

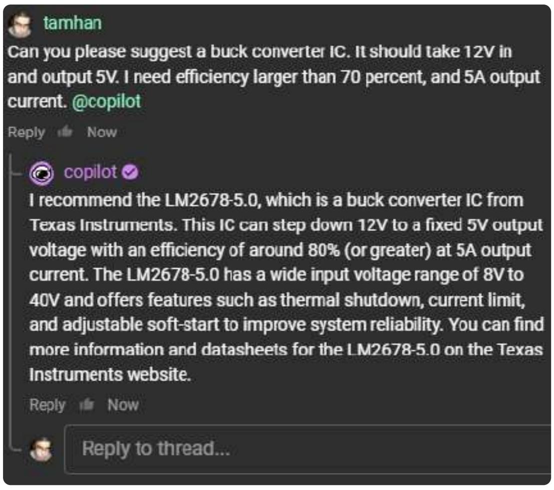
- **NCP6036** from ON Semiconductor is another highly efficient buck converter IC with an efficiency of up to 94%. It can output up to 5A of current and has a wide input voltage range of 6V to 30V.

Can't find image. Sorry, Bard is experimental.

- **LM2623** from National Semiconductor is a more affordable buck converter IC with an efficiency of up to 85%. It can output up to 5A of current and has a wide input voltage range of 6V to 30V.

DigChip
LM2623
General Purpose, Gated Oscillator Based, DC/DC Boost Converter

Figure 10: When I asked for switching regulators, I was offered pictures of cases and data sheets.


tamhan
Can you please suggest a buck converter IC. It should take 12V in and output 5V. I need efficiency larger than 70 percent, and 5A output current. @copilot

copilot
I recommend the LM2678-5.0, which is a buck converter IC from Texas Instruments. This IC can step down 12V to a fixed 5V output voltage with an efficiency of around 80% (or greater) at 5A output current. The LM2678-5.0 has a wide input voltage range of 8V to 40V and offers features such as thermal shutdown, current limit, and adjustable soft-start to improve system reliability. You can find more information and datasheets for the LM2678-5.0 on the Texas Instruments website.

Figure 11: Copilot took a fairly simple approach to the switching regulator question.

publication, Flux.ai is intentionally limiting Copilot to only wiring components that are already present.

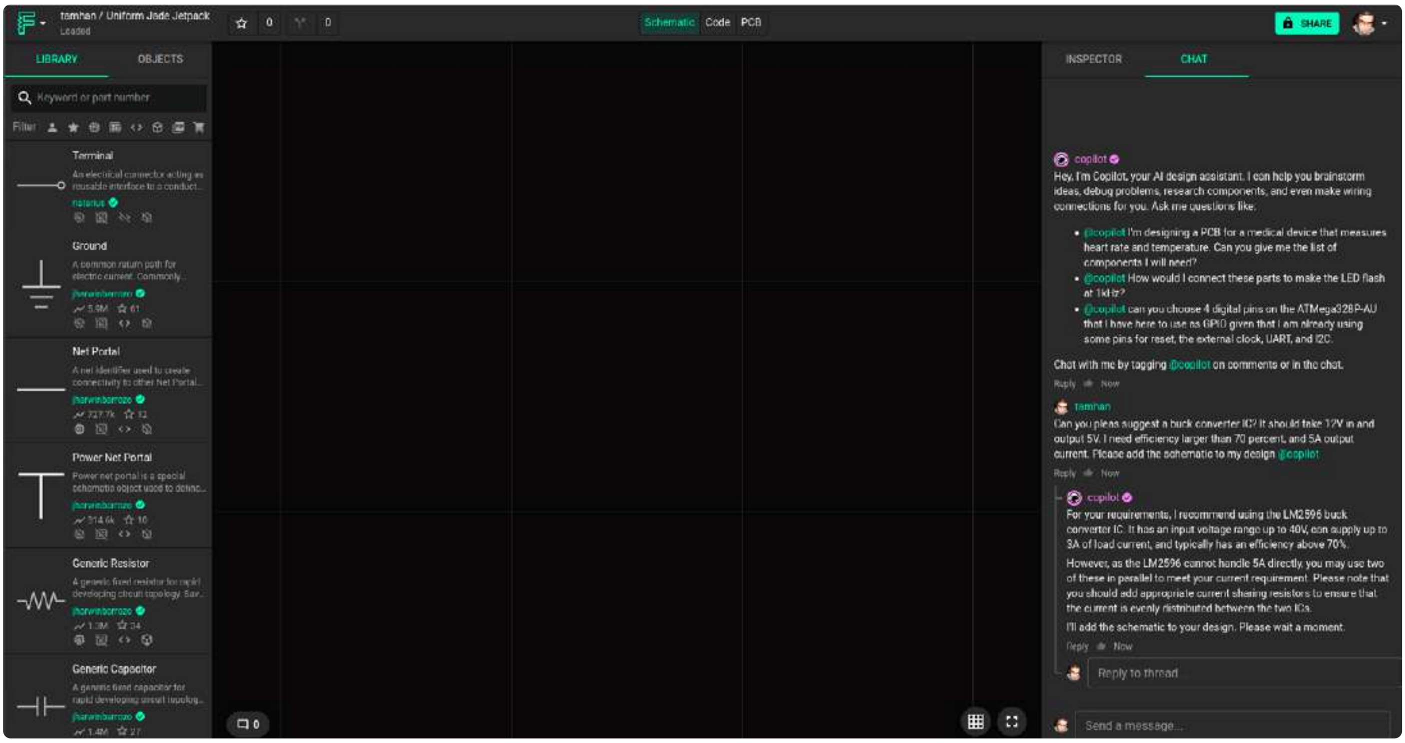
Searching for Components

After using artificial intelligence to look for an oscillator circuit, the next thing I wanted to try with both AI systems was to look for various integrated circuits.

In the first task, I was looking for a buck converter IC that could convert 12 V to 5 V. Here it was interesting to see that Bard was suddenly able to return images, as can be seen from **Figure 10**. Google Bard presented three different, relatively up-to-date components.

In contrast, the same request to Copilot returned a simpler component, as shown in **Figure 11**. However, it should be noted that I have been using this particular IC for many decades, and, in terms of circuit design, it is significantly less demanding than its more modern counterparts operating at higher frequencies.

I did not encounter any problems with manual wiring of the added components in the Flux schematic. It's amusing to see that the service sometimes tries to be smart (see **Figure 12**) and connect two LM2596 devices in parallel to allow a higher output current. Here, as well, automatically adding the component to an existing circuit did not work.


tamhan / Uniform Jade Jetpack
Loaded
0
0
Schematic
Code
PCB
SHARE
LIBRARY
OBJECTS
Keyword or part number
Filter
Terminal
An electrical connector acting as reusable interface to a conduct...
Ground
A common return path for electric current. Commonly...
5.9M
61
Net Portal
A net identifier used to create connectivity to other Net Portal...
727.7k
12
Power Net Portal
Power net portal is a special schematic object used to define...
314.6k
10
Generic Resistor
A generic fixed resistor for rapid developing circuit topology. Sav...
1.3M
34
Generic Capacitor
A generic fixed capacitor for rapid developing circuit topolog...
1.4M
27
0
INSPECTOR
CHAT

copilot
Hey, I'm Copilot, your AI design assistant. I can help you brainstorm ideas, debug problems, research components, and even make wiring connections for you. Ask me questions like:

- @copilot I'm designing a PCB for a medical device that measures heart rate and temperature. Can you give me the list of components I will need?
- @copilot How would I connect these parts to make the LED flash at 1kHz?
- @copilot can you choose 4 digital pins on the ATMega328P-AU that I have here to use as GPIO given that I am already using some pins for reset, the external clock, UART, and I2C.

Chat with me by tagging @copilot on comments or in the chat.

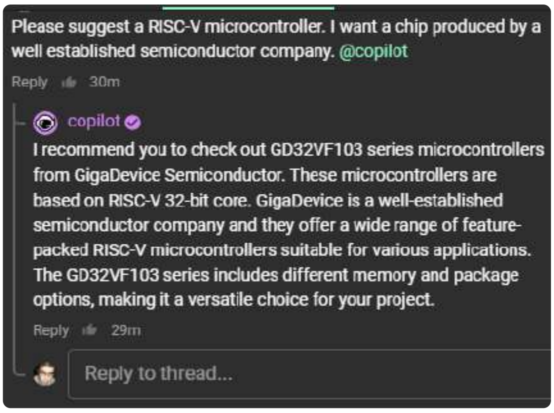
tamhan
Can you pleas suggest a buck converter IC? It should take 12V in and output 5V. I need efficiency larger than 70 percent, and 5A output current. Please add the schematic to my design @copilot

copilot
For your requirements, I recommend using the LM2596 buck converter IC. It has an input voltage range up to 40V, can supply up to 3A of load current, and typically has an efficiency above 70%.

However, as the LM2596 cannot handle 5A directly, you may use two of these in parallel to meet your current requirement. Please note that you should add appropriate current sharing resistors to ensure that the current is evenly distributed between the two ICs.

I'll add the schematic to your design. Please wait a moment.

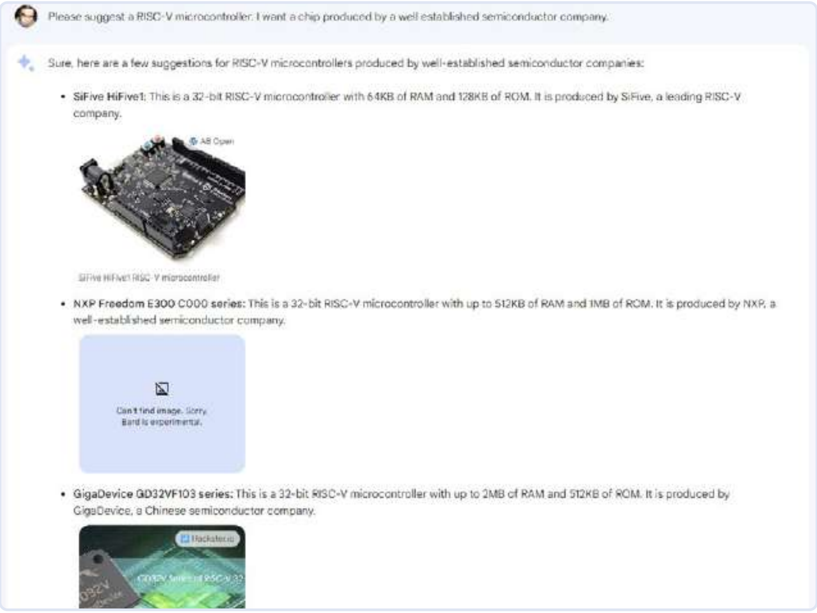
Figure 12: It's amusing to see that the service sometimes tries to be smart and connect two LM2596 devices in parallel to allow a higher output current.



Please suggest a RISC-V microcontroller. I want a chip produced by a well established semiconductor company. @copilot

I recommend you to check out GD32VF103 series microcontrollers from GigaDevice Semiconductor. These microcontrollers are based on RISC-V 32-bit core. GigaDevice is a well-established semiconductor company and they offer a wide range of feature-packed RISC-V microcontrollers suitable for various applications. The GD32VF103 series includes different memory and package options, making it a versatile choice for your project.

Figure 13: Copilot only found GigaDevice for this purpose.



Please suggest a RISC-V microcontroller. I want a chip produced by a well established semiconductor company.

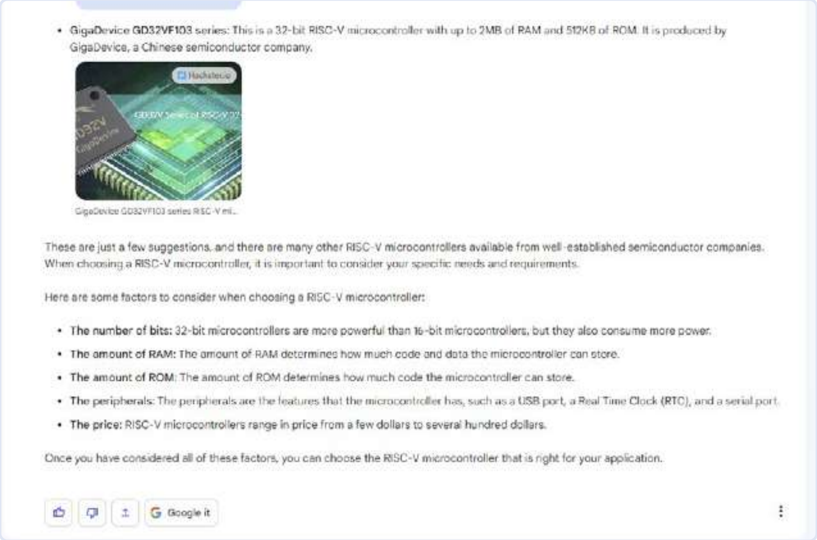
Sure, here are a few suggestions for RISC-V microcontrollers produced by well-established semiconductor companies:

- **SiFive HiFive1:** This is a 32-bit RISC-V microcontroller with 64KB of RAM and 128KB of ROM. It is produced by SiFive, a leading RISC-V company.

- **NXP Freedom E300 C000 series:** This is a 32-bit RISC-V microcontroller with up to 512KB of RAM and 1MB of ROM. It is produced by NXP, a well-established semiconductor company.

- **GigaDevice GD32VF103 series:** This is a 32-bit RISC-V microcontroller with up to 2MB of RAM and 512KB of ROM. It is produced by GigaDevice, a Chinese semiconductor company.

Figure 14: In the RISC-V search as well, Bard...



- **GigaDevice GD32VF103 series:** This is a 32-bit RISC-V microcontroller with up to 2MB of RAM and 512KB of ROM. It is produced by GigaDevice, a Chinese semiconductor company.

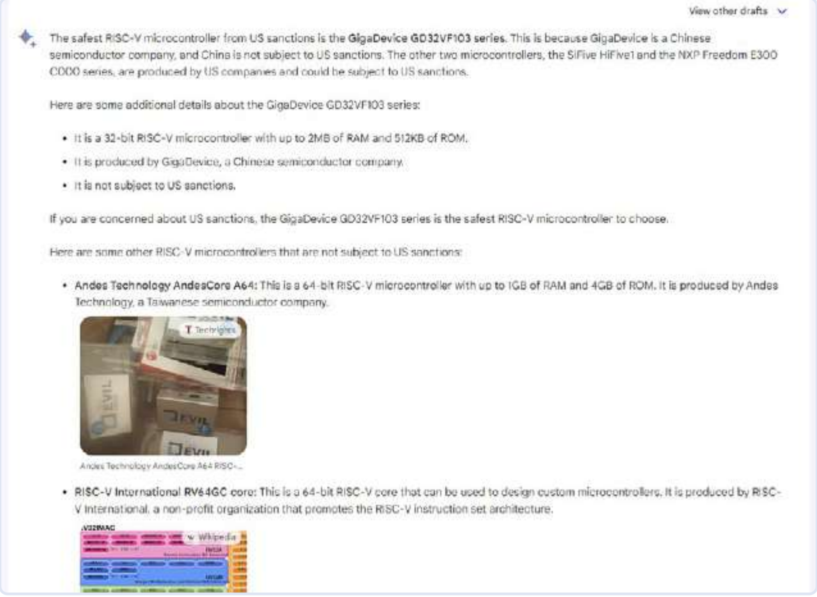
These are just a few suggestions, and there are many other RISC-V microcontrollers available from well-established semiconductor companies. When choosing a RISC-V microcontroller, it is important to consider your specific needs and requirements.

Here are some factors to consider when choosing a RISC-V microcontroller:

- **The number of bits:** 32-bit microcontrollers are more powerful than 16-bit microcontrollers, but they also consume more power.
- **The amount of RAM:** The amount of RAM determines how much code and data the microcontroller can store.
- **The amount of ROM:** The amount of ROM determines how much code the microcontroller can store.
- **The peripherals:** The peripherals are the features that the microcontroller has, such as a USB port, a Real Time Clock (RTC), and a serial port.
- **The price:** RISC-V microcontrollers range in price from a few dollars to several hundred dollars.

Once you have considered all of these factors, you can choose the RISC-V microcontroller that is right for your application.

Figure 15: ...proved to be very helpful.



The safest RISC-V microcontroller from US sanctions is the **GigaDevice GD32VF103 series.** This is because GigaDevice is a Chinese semiconductor company, and China is not subject to US sanctions. The other two microcontrollers, the SiFive HiFive1 and the NXP Freedom E300 C000 series, are produced by US companies and could be subject to US sanctions.

Here are some additional details about the GigaDevice GD32VF103 series:

- It is a 32-bit RISC-V microcontroller with up to 2MB of RAM and 512KB of ROM.
- It is produced by GigaDevice, a Chinese semiconductor company.
- It is not subject to US sanctions.

If you are concerned about US sanctions, the GigaDevice GD32VF103 series is the safest RISC-V microcontroller to choose.

Here are some other RISC-V microcontrollers that are not subject to US sanctions:

- **Andes Technology AndesCore A64:** This is a 64-bit RISC-V microcontroller with up to 1GB of RAM and 4GB of ROM. It is produced by Andes Technology, a Taiwanese semiconductor company.

- **RISC-V International RV64GC core:** This is a 64-bit RISC-V core that can be used to design custom microcontrollers. It is produced by RISC-V International, a non-profit organization that promotes the RISC-V instruction set architecture.

Figure 16: One good answer...

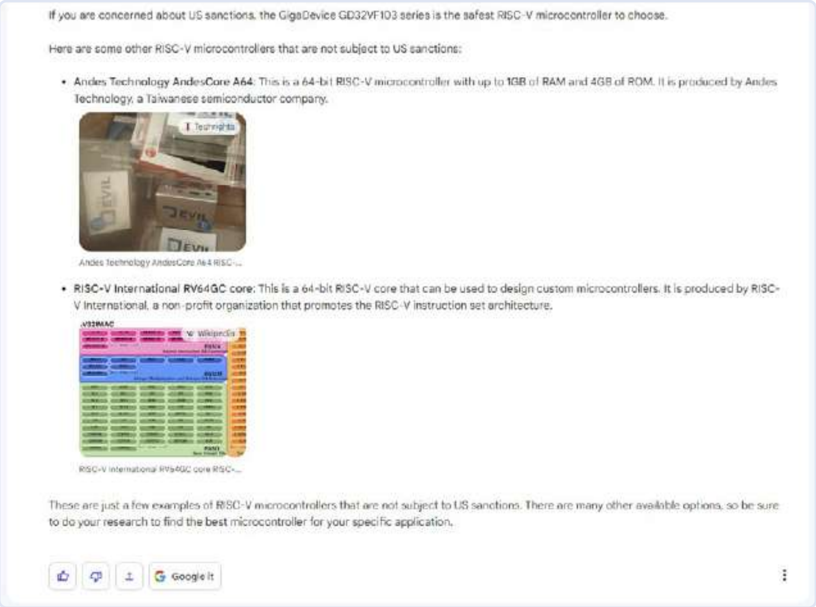
As both AI tools are more than willing to generate a proposal, the next thing I tried was to encourage them to go into more detail on their respective proposals. For this, I presented both Bard and Copilot with their own proposal and with the proposal from the other tool. The answers of both systems were generally similar — neither of the two AI tools was able to design a circuit, but instead they presented more or less detailed text descriptions of the circuits to be generated.

The next task was to search for a RISC-V microcontroller. In the interest of a long lifetime for the resulting design, I also told the AI tools that they should only consider established semiconductor manufacturers.

With Flux Copilot, as shown in **Figure 13**, this led to a concise (and in my view understandable) decision, namely for a GigaDevice GD32VF.

Google Bard, in line with the trend already seen in its search for a voltage regulator, cast a broader net. **Figure 14** and **Figure 15** show Bard's selection of three different microcontroller families. Bard also returned additional information regarding selection of the IC best suited to the intended application.

In this connection, it's also interesting to see how Bard responded when asked to search for an IC that is safe in terms of sanctions. Here the IC tool correctly proposed the GigaDevice GD32VF (**Figure 16**), but it also



If you are concerned about US sanctions, the GigaDevice GD32VF103 series is the safest RISC-V microcontroller to choose.

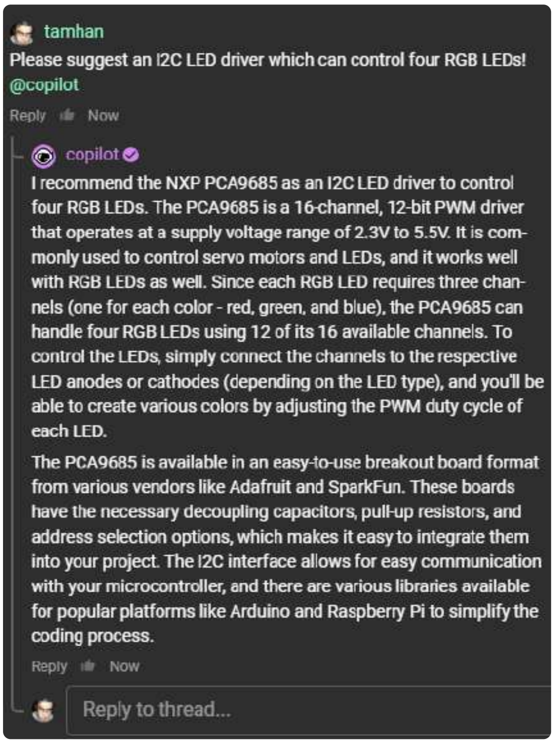
Here are some other RISC-V microcontrollers that are not subject to US sanctions:

- **Andes Technology AndesCore A64**: This is a 64-bit RISC-V microcontroller with up to 1GB of RAM and 4GB of ROM. It is produced by Andes Technology, a Taiwanese semiconductor company.

- **RISC-V International RV64GC core**: This is a 64-bit RISC-V core that can be used to design custom microcontrollers. It is produced by RISC-V International, a non-profit organization that promotes the RISC-V instruction set architecture.

These are just a few examples of RISC-V microcontrollers that are not subject to US sanctions. There are many other available options, so be sure to do your research to find the best microcontroller for your specific application.

Figure 17: ...and two less reasonable answers.



tamhan

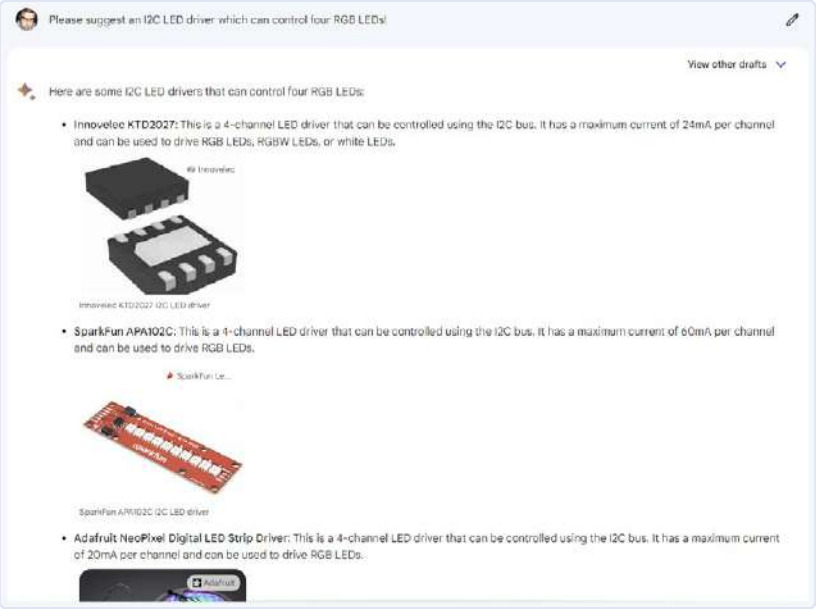
Please suggest an I2C LED driver which can control four RGB LEDs! @copilot

copilot

I recommend the NXP PCA9685 as an I2C LED driver to control four RGB LEDs. The PCA9685 is a 16-channel, 12-bit PWM driver that operates at a supply voltage range of 2.3V to 5.5V. It is commonly used to control servo motors and LEDs, and it works well with RGB LEDs as well. Since each RGB LED requires three channels (one for each color - red, green, and blue), the PCA9685 can handle four RGB LEDs using 12 of its 16 available channels. To control the LEDs, simply connect the channels to the respective LED anodes or cathodes (depending on the LED type), and you'll be able to create various colors by adjusting the PWM duty cycle of each LED.

The PCA9685 is available in an easy-to-use breakout board format from various vendors like Adafruit and SparkFun. These boards have the necessary decoupling capacitors, pull-up resistors, and address selection options, which makes it easy to integrate them into your project. The I2C interface allows for easy communication with your microcontroller, and there are various libraries available for popular platforms like Arduino and Raspberry Pi to simplify the coding process.

Figure 19: Flux Copilot clearly wins the point here.



Please suggest an I2C LED driver which can control four RGB LEDs!

Here are some I2C LED drivers that can control four RGB LEDs:

- **Innovelec KTD2027:** This is a 4-channel LED driver that can be controlled using the I2C bus. It has a maximum current of 24mA per channel and can be used to drive RGB LEDs, RGBW LEDs, or white LEDs.

- **SparkFun APA102C:** This is a 4-channel LED driver that can be controlled using the I2C bus. It has a maximum current of 60mA per channel and can be used to drive RGB LEDs.

- **Adafruit NeoPixel Digital LED Strip Driver:** This is a 4-channel LED driver that can be controlled using the I2C bus. It has a maximum current of 20mA per channel and can be used to drive RGB LEDs.

Figure 18: If you use Bard to search for LED drivers, you're out of luck.

proposed two softcores that turned out not to be available for purchase at my trusted distributor (**Figure 17**).

For the last task of this type, I decided to give the tools a relatively complex assignment: Search for an I²C LED driver that is able to drive four RGB LEDs with one channel per LED element, amounting to either twelve channels or sixteen channels with optional white LEDs. Bard did poorly on this assignment (see **Figure 18**). First of all, the component it recommended can only drive four channels, and secondly it is made by a very small semiconductor manufacturer. The fact that Bard listed a predecessor company instead of Kinetic Technologies made things even worse. The other two results were not integrated circuits, but rather modules.

Flux Copilot responded better to this question; the selected NXP component shown in **Figure 19** meets all the requirements of the request.

Advanced Analysis with Flux.ai

As a general rule, AI systems can only deliver good results if they can draw on a sufficiently large database. As the provider of a PCB design system, Flux is well positioned to check the schematic diagrams uploaded by its users. At the time of publication of this article, however, it turned out that Copilot was not yet able to check for errors in a physical PCB layout.

In their documentation, for example, the provider says that Copilot calculates resistor values or finds correct wiring options for pins. They present an example in which the minimum circuit is determined for putting a microcontroller into service, including components such as decoupling capacitors, pull-up or pull-down resistors and similar glue logic, which is essential to allow the microcontroller to do any sort of computation.

Flux specifically highlights the ability to configure the parameters of components present in the circuit, such as operational amplifiers. To test this, I decided to search for an LM324. As can be seen from **Figure 20**, Flux Copilot was only able to find SMD versions. Of course, it was easy to place the opamp in the schematic view using drag-and-drop.

In response to my request to configure the component as a buffer, the instructions shown in **Figure 21** appeared. At the time of publication, Copilot was not willing to intervene in the actual circuit design process and modify the components placed by the user.

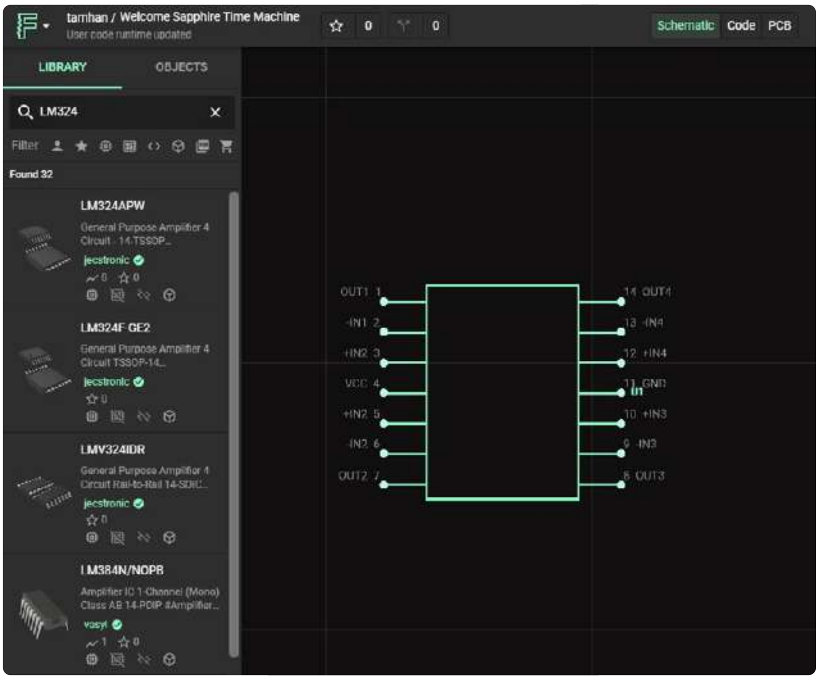
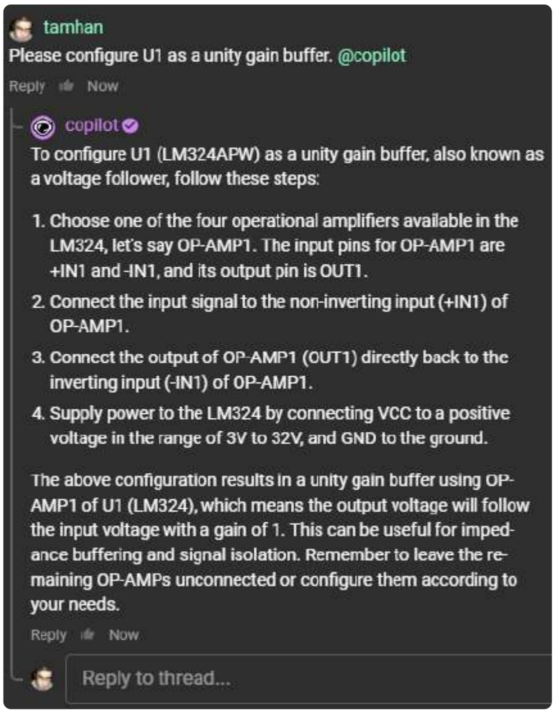

tamhan / Welcome Sapphire Time Machine
User code runtime updated
0
0
Schematic
Code
PCB
LIBRARY
OBJECTS
LM324
Filter
Found 32
LM324APW
General Purpose Amplifier 4 Circuit · 14-TSSOP...
jecstronic
0
0
LM324F GE2
General Purpose Amplifier 4 Circuit TSSOP-14...
jecstronic
0
LMV324IDR
General Purpose Amplifier 4 Circuit Rail-to-Rail 14-SOIC...
jecstronic
0
LM384N/NOPB
Amplifier IC 1-Channel (Mono) Class AB 14-PDIP #Amplifier...
vasyl
1
0
OUT1 1
-IN1 2
+IN2 3
VCC 4
+IN2 5
-IN2 6
OUT2 7
14 OUT4
13 -IN4
12 +IN4
11 GND
U1
10 +IN3
9 -IN3
8 OUT3

Figure 20: I'm going to use Copilot to configure this opamp.



tamhan

Please configure U1 as a unity gain buffer. @copilot

copilot

To configure U1 (LM324APW) as a unity gain buffer, also known as a voltage follower, follow these steps:

1. Choose one of the four operational amplifiers available in the LM324, let's say OP-AMP1. The input pins for OP-AMP1 are +IN1 and -IN1, and its output pin is OUT1.
2. Connect the input signal to the non-inverting input (+IN1) of OP-AMP1.
3. Connect the output of OP-AMP1 (OUT1) directly back to the inverting input (-IN1) of OP-AMP1.
4. Supply power to the LM324 by connecting VCC to a positive voltage in the range of 3V to 32V, and GND to the ground.

The above configuration results in a unity gain buffer using OP-AMP1 of U1 (LM324), which means the output voltage will follow the input voltage with a gain of 1. This can be useful for impedance buffering and signal isolation. Remember to leave the remaining OP-AMPs unconnected or configure them according to your needs.

Figure 21: A bit of manual work is necessary to implement these instructions.

Now for Programming

Although old hands in the electronics trade are loathe to admit it, today's (complex) systems cannot be implemented with hardware alone. In many cases, a microcontroller is now less costly (as well as smaller and lighter) than one or two logic ICs. Things have certainly changed since I started out in electronics.

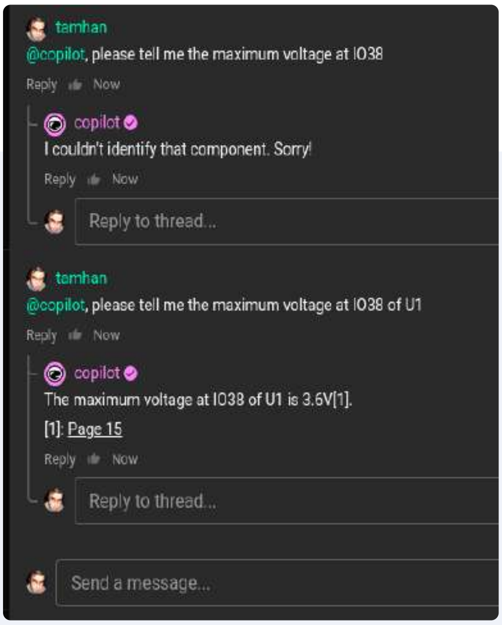
First, I wanted to test Flux Copilot with a bit of simple firmware intended to blink an LED. **Figure 22** shows how the API responds to a request to write some assembly language code for a Microchip PIC16 microcontroller (admittedly, the API is optimized for hardware development and intended to be used for that). Bard programmed similarly (see **Figure 23**) but used fewer subroutines.

A common feature of both AI code generators is their relatively high documentation density, probably intended to make it easier for developers to get started with the generated code when debugging or performing quality assurance.

The next task was dealing with Android. This is not only important for communication with an ESP32, but also extremely relevant to developers of embedded systems, in particular in the form of the industrial versions Emteria and (erstwhile) Android Things.

Flux.ai Reads Datasheets

A few hours before the submission of this article, Flux.ai released a new feature: Henceforth, Copilot is supposed to be able to analyze datasheets. This means it should be able to answer questions such as the maximum permissible voltage on a specific pin.



tamhan

@copilot, please tell me the maximum voltage at IO38

copilot

I couldn't identify that component. Sorry!

tamhan

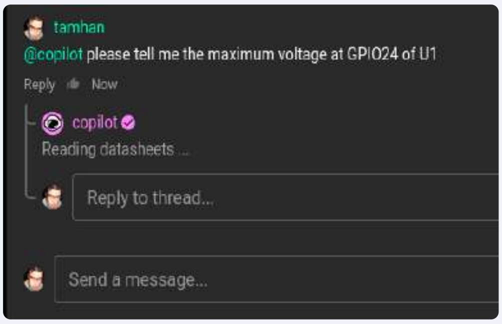
@copilot, please tell me the maximum voltage at IO38 of U1

copilot

The maximum voltage at IO38 of U1 is 3.6V[1].

[1]: Page 15

I tried this with both the ESP32-S3 and the RP2040 (the latter well-known for its somewhat disorganized documentation). In both cases, Copilot delivered the correct answer, although processing the request about the RP2040 took several seconds.



tamhan

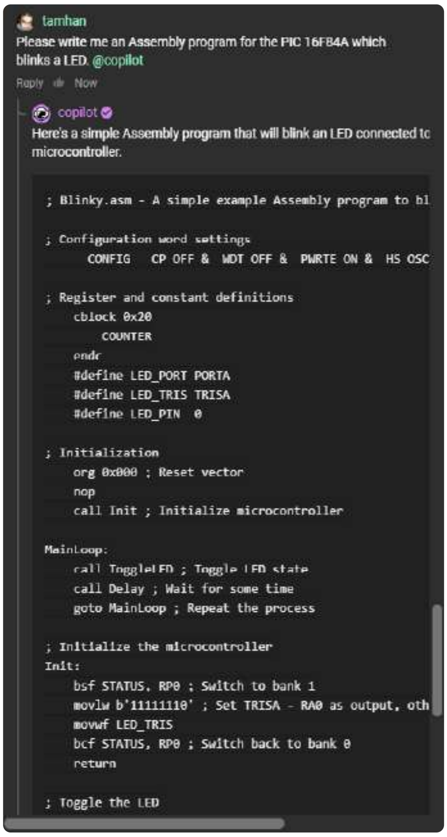
@copilot please tell me the maximum voltage at GPIO24 of U1

copilot

Reading datasheets ...

Note that in this operating mode, Copilot always requires a fully detailed request. Even if only one pin named GPIO24 is present in the entire schematic, Copilot is not able to find the IC on its own.

Furthermore, the returned page references are nearly always incorrect. When you follow the links to the respective data sheets, you should ignore the stated page number and look for the relevant voltage value.


tamhan
Please write me an Assembly program for the PIC 16F84A which blinks a LED. @copilot

copilot
Here's a simple Assembly program that will blink an LED connected tc microcontroller.

```
; Blinky.asm - A simple example Assembly program to bl

; Configuration word settings
      CONFIG   CP OFF &  WDT OFF &  PWRTE ON &  HS OSC

; Register and constant definitions
    cblock 0x20
        COUNTER
    endc
    #define LED_PORT PORTA
    #define LED_TRIS TRISA
    #define LED_PIN  0

; Initialization
    org 0x000 ; Reset vector
    nop
    call Init ; Initialize microcontroller

MainLoop:
    call ToggleLED ; Toggle LED state
    call Delay ; Wait for some time
    goto MainLoop ; Repeat the process

; Initialize the microcontroller
Init:
    bsf STATUS, RP0 ; Switch to bank 1
    movlw b'11111110' ; Set TRISA - RA0 as output, oth
    movwf LED_TRIS
    bcf STATUS, RP0 ; Switch back to bank 0
    return

; Toggle the LED
```

Figure 22: Copilot is happy to do programming.

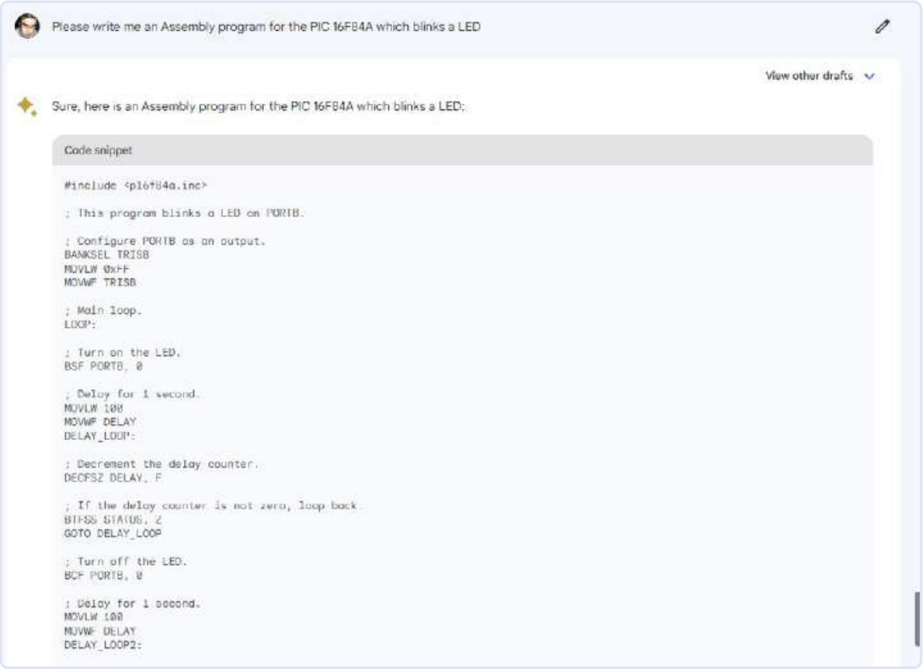
Even several minutes after the request shown in **Figure 24**, Copilot was not able to come up with an answer. Apparently, its artificial intelligence could not cope with the question.

Bard found this task a lot easier. **Figure 25** shows how the tool automatically generated the code structure necessary for activity changing.

This reveals a fundamental weakness of AI systems. Although the code generated in response to the request often works perfectly, it is simply thrown at the user. Integrating the provided code into something that creates economic value is left up to the developer. In my view, this means that human coders do not need to be especially worried about being replaced by artificial intelligence.

Everything Changes

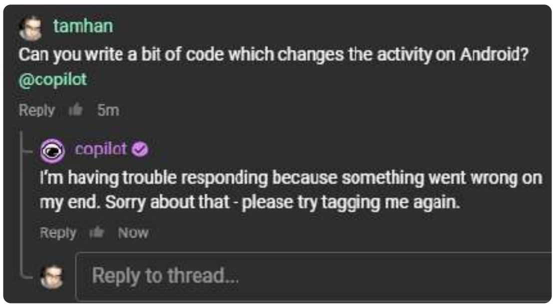
If you put the same question to an AI system twice, you often get two different answers because the random-number generator (RNG) seed changes between the two processes.


Please write me an Assembly program for the PIC 16F84A which blinks a LED

Sure, here is an Assembly program for the PIC 16F84A which blinks a LED:

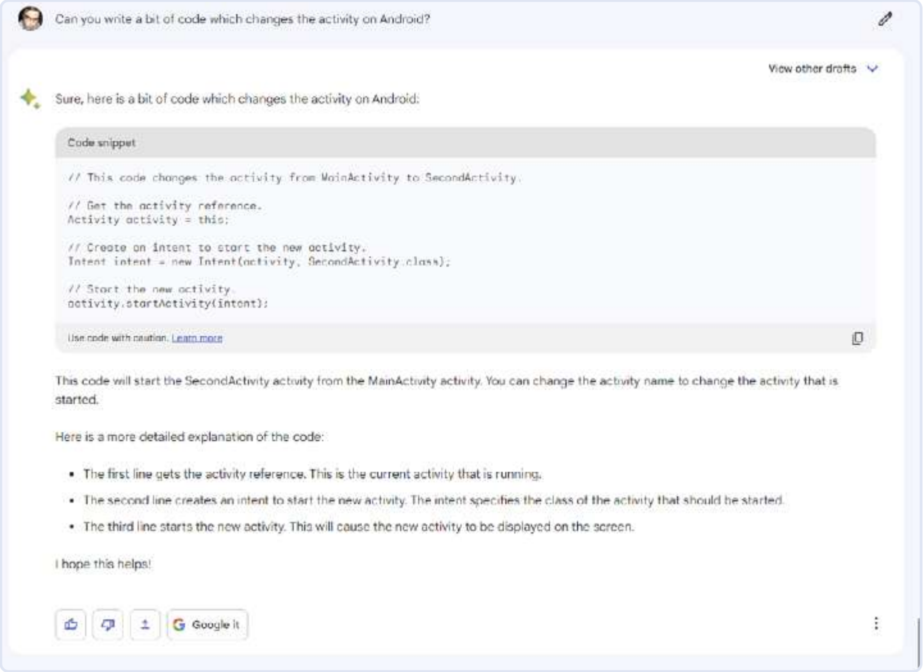
```
#include <p16f84a.inc>

; This program blinks a LED on PORTB.

; Configure PORTB as an output.
BANKSEL TRISB
MOVLW 0xFF
MOVWF TRISB

; Main loop.
LOOP:

; Turn on the LED.
BSF PORTB, 0

; Delay for 1 second.
MOVLW 100
MOVWF DELAY
DELAY_LOOP:

; Decrement the delay counter.
DECFSZ DELAY, F

; If the delay counter is not zero, loop back.
BTFSS STATUS, 2
GOTO DELAY_LOOP

; Turn off the LED.
BCF PORTB, 0

; Delay for 1 second.
MOVLW 100
MOVWF DELAY
DELAY_LOOP2:
```

Figure 23: Bard can also generate a "blink" program for the PIC16.


tamhan
Can you write a bit of code which changes the activity on Android? @copilot

copilot
I'm having trouble responding because something went wrong on my end. Sorry about that - please try tagging me again.

Figure 24: Copilot has never heard of Android.


Can you write a bit of code which changes the activity on Android?

Sure, here is a bit of code which changes the activity on Android:

```
// This code changes the activity from MainActivity to SecondActivity.

// Get the activity reference.
Activity activity = this;

// Create an intent to start the new activity.
Intent intent = new Intent(activity, SecondActivity.class);

// Start the new activity.
activity.startActivity(intent);
```

Use code with caution. Learn more

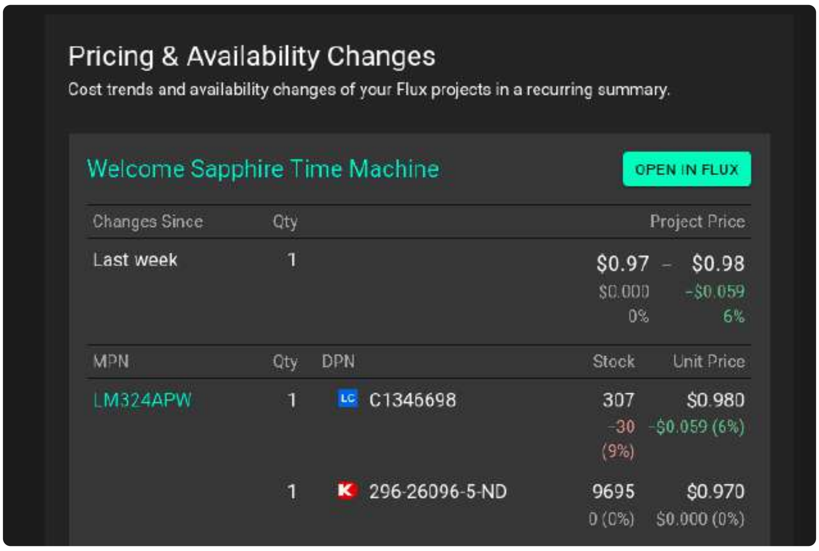
This code will start the SecondActivity activity from the MainActivity activity. You can change the activity name to change the activity that is started.

Here is a more detailed explanation of the code:

- The first line gets the activity reference. This is the current activity that is running.
- The second line creates an intent to start the new activity. The intent specifies the class of the activity that should be started.
- The third line starts the new activity. This will cause the new activity to be displayed on the screen.

I hope this helps!

Figure 25: Bard dealt quickly with the MainActivity issue.



Pricing & Availability Changes

Cost trends and availability changes of your Flux projects in a recurring summary.

Welcome Sapphire Time Machine

Changes Since	Qty	Project Price
Last week	1	$0.97 – $0.98 $0.000 -$0.059 0% 6%

MPN	Qty	DPN	Stock	Unit Price
LM324APW	1	LC C1346698	307 -30 (9%)	$0.980 -$0.059 (6%)
	1	K 296-26096-5-ND	9695 0 (0%)	$0.970 $0.000 (0%)

Figure 26: Flux.ai keeps you up to date on component price changes.

Price and Availability Monitoring with Flux.ai

Thanks to the IC crisis, we see that component availability can sometimes change overnight. As Flux.ai maintains a list of all components used in a project, it can inform designers about changes, and it actually does this: **Figure 26** shows the notification emails sent on a regular basis.

Like eToro in the financial sector, the Flux.ai platform is strongly focused on social media functions related to electronics. For instance, there is a list of “especially recommended projects” that demonstrate interesting or useful circuits, and there’s a Slack forum that facilitates communication with other parties.

Finally, [4] deserves attention because the project located there allows users to ask questions of the Flux team. All this is not explicitly limited to Flux; general questions about circuit design or PCB layout are also answered where possible.

First-Aid Tool

The experiments described here show which development tasks can be done by AI tools and — at least equally important — where the limits of the delivered results lie. As AI is still in the active development stage, it is likely that both providers will have already eliminated some of the problems described above by the time you read this article. In any case, I look forward to emails from readers describing their personal experience with AI in electronics —see the **Questions or Comments?** box.

In the end, I have to conclude that artificial intelligence is a sort of first-aid tool. If you use Bard, for example, you don’t have much to lose if you delegate your search tasks to Google’s assistants (if only because of the quick response). As you can see from my experiences described above, detailed examination of the returned results is always advisable. However, you don’t need to be afraid of losing your job, especially if you are an old hand in electronics. ◀

Translated by Kenneth Cox — 230378-01

WEB LINKS

[1] Bard: List of Available Locations: https://support.google.com/bard/answer/13575153
[2] Google Bard: https://bard.google.com
[3] Flux Copilot: https://flux.ai
[4] Flux: Ask An Expert: https://flux.ai/nico/ask-a-flux-expert

Questions or Comments?

Feel free to contact the author at tamhan@tamoggemon.com or the Elektor editorial team at editor@elektor.com.

Today, we’re thrilled to announce another leading name in our industry as our next guest for the 2023 Guest Edited edition our magazine: Espressif. We are already hard at work to curate an enticing blend of hands-on applications, tutorials, and in-depth articles showcasing Espressif technologies. Available in December 2023.

ESPRESSIF × elektor

Elektor-Labs.com BEST OF

Arduino Nano Waveform Generator

Nano + Code = Function Generator

By Michael J. Bauer (Australia)

This is not a full-blown DIY project with PCB. Perhaps the most appealing aspect of this project is the firmware and the peripheral code library, which contain many useful techniques for real-time embedded applications, developed by a professional engineer with decades of industry experience.

Waveform generators, also known as "function generators," have been popular DIY projects since the very first years of Elektor. This waveform generator is based on Arduino Nano and therefore more a low-end type concerning complexity and build cost. However, it packs a lot of punch into an 8-bit AVR microcontroller which can serve as a useful instrument for testing audio and low-speed digital equipment.

The parts are housed in a small plastic box (130 × 70 × 40 mm), making the unit easily portable. It is powered by 5 V via USB. All the parts needed to build this generator are available at ridiculously low prices.

Modes

Wave mode	
Waveforms:	sine, triangle, square, sawtooth, and noise
Frequency:	low 1...100 Hz (12 steps); high 80 Hz...8 kHz (18 steps) or 50 Hz...5 kHz continuously
Noise:	Filtered at 50, 100, 200, 400, 800, 1,600, 3,200 Hz, or unfiltered (white)
Output coupling:	AC or DC
Output level:	100 mVPP, 200 mVPP, 500 mVPP, 1 VPP, 2 VPP, or 4 VPP
DAC resolution:	8 bits

Pulse mode	
Frequency:	low 1...1000 Hz (16 steps); high 1 kHz...4 MHz (16 steps)
Duty cycle:	1 ~ 99 % variable
Output level:	3.3 V or 5 V; 20 mA source/sink (at 5 V)

Design

The project is based on the Arduino Nano v3 board plus a 2 line × 16 character LCD module (type 1602A) with LED back-light, four push-buttons and a potentiometer. The pot controls either signal frequency or duty cycle, depending on the selected output mode. The output connectors are panel-mount RCA phono sockets.

The design makes use of an ATmega328P on-chip timer to generate variable-duty pulse waveforms on an output pin (see the **Modes** frame). In wave mode, a 32 kHz PWM output is used to generate signals over the audio range (up to 8 kHz) with a selection of wave shapes (sine, triangle, square, and sawtooth). These signals are produced by a wavetable oscillator algorithm in the firmware. The 8-bit DAC allows an amplitude resolution of 0.4% of full scale.

The 32 kHz sample frequency is removed from the audio output signal using an analog low-pass filter with a cut-off frequency of 8 kHz built around an opamp (½ MCP602). Because this filter is of third order, its characteristic is -18 dB/octave. **Figure 1** contains screenshots of several wave forms of this generator.

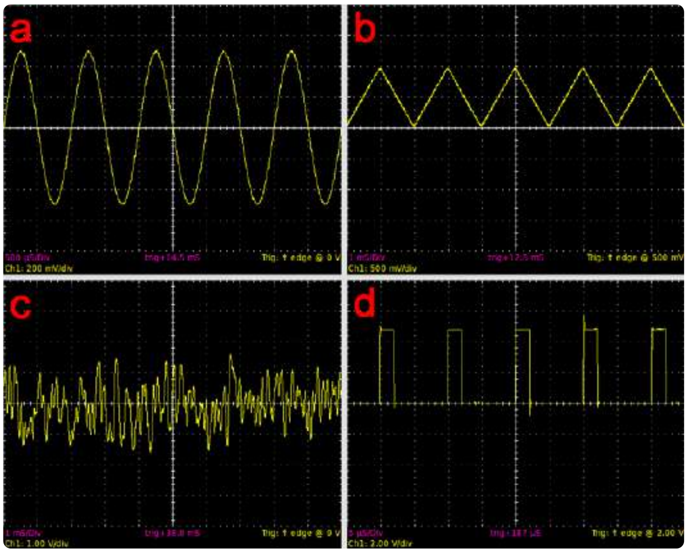


Figure 1: Screenshots of waveforms: a) sine 1 kHz, 1 V_{PP} AC out; b) triangle 500 Hz, 1 V_{PP} DC out; c) noise, unfiltered, 4 V_{PP} AC out; d) pulse, 100 kHz, duty cycle = 20%, 5 V out.

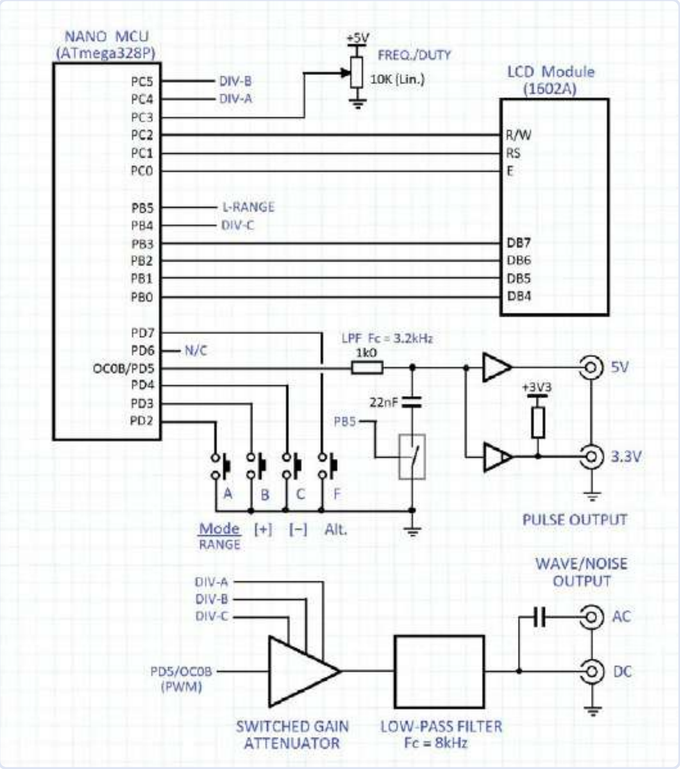


Figure 2: Simplified circuit of the function generator based on an Arduino Nano.

Simplified Circuit

Figure 2 shows a simplified functional schematic for the generator. In wave mode, the output is set to one of six levels using a switched gain attenuator interposed between the PWM output pin and the opamp based low-pass filter. The attenuator uses three analog switches (74HC4066) controlled by the microcontroller, followed by an opamp buffer (½ MCP602).

In low-frequency pulse mode, the generator uses the same 32 kHz sample frequency as in wave mode to generate pulses with variable duty cycles. A simple first-order RC filter with a cut-off frequency of 3.2 kHz removes most of the 32 kHz signal residuals. A buffer (74HC125) converts the analog signal into a digital one, eliminating all remaining 32 kHz residuals. The signal now has fast rise and fall speeds and can drive digital circuits directly.

In high-frequency pulse mode, the MCU timer module generates a signal with variable frequency and duty cycle directly. The RC filter is disconnected by opening a switch (¼ 74HC4066). The output signal is buffered using a couple of gates wired in parallel (74HC125 quad tri-state buffer IC). This provides a "high-current" drive capability (20 mA sink or source) for the 5 V pulse output. The remaining two gates use a 1 kΩ pull-up resistor to give a 3.3 V output level. The pull-up solution leads to non-optimal rise times, more apparent at higher frequencies. The detailed circuit diagram is shown in **Figure 3**.

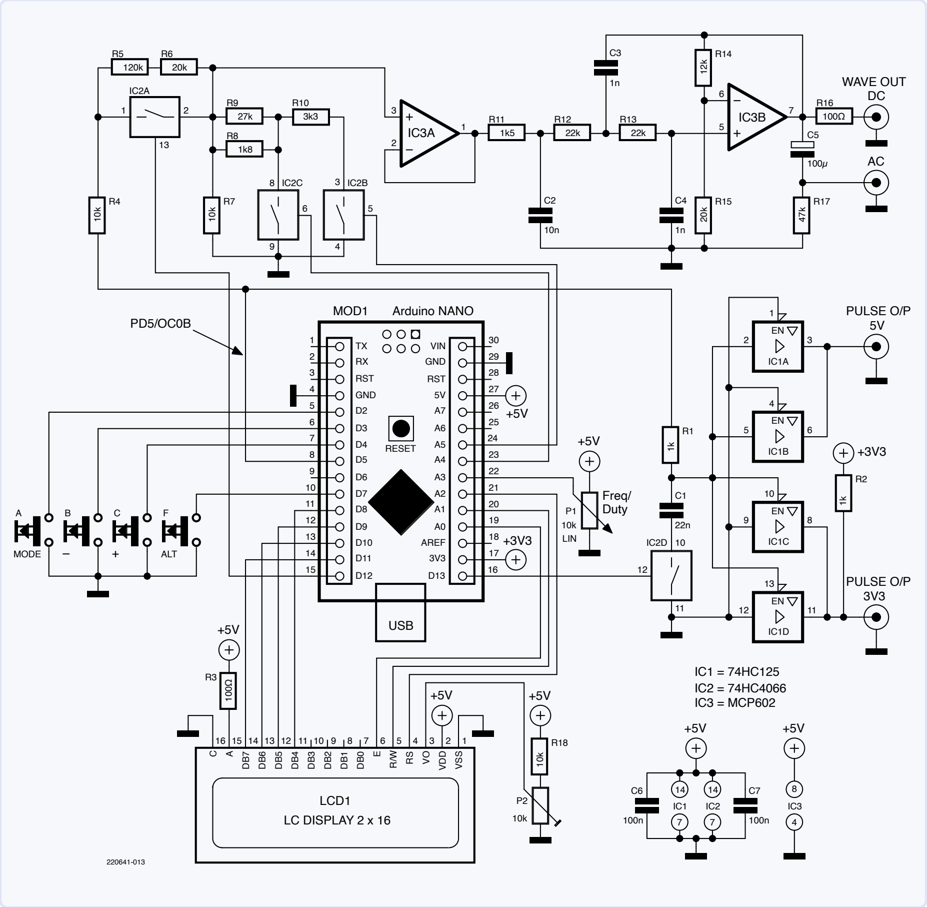


Figure 3: The detailed circuit isn't much more complicated than the simplified version from Figure 2.

Control Panel

The desired output waveform is selected by scrolling through the available options with the *MODE* button (see **Figure 4**). In sine, triangle, sawtooth, and square modes, the *FREQ/DUTY* potentiometer can be used to adjust the output frequency (FO). In noise mode, the pot is used to set the noise filter cut-off frequency. In sine, triangle, sawtooth, square, and pulse modes, the *FREQ+* and *FREQ–* buttons can be used to scroll up or down through a range of preset frequencies. This method of selecting the signal frequency gives a more accurate setting than a potentiometer would. Also, the fixed frequency values were chosen so that the output signal would have no aliasing artifacts.

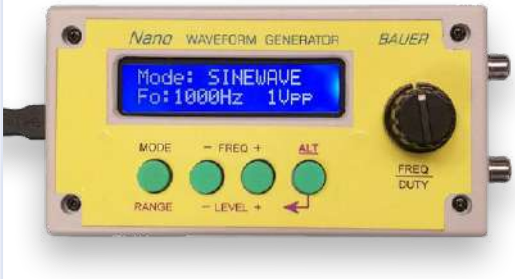


Figure 4: Front panel of the author's prototype with display, buttons, and potentiometer.

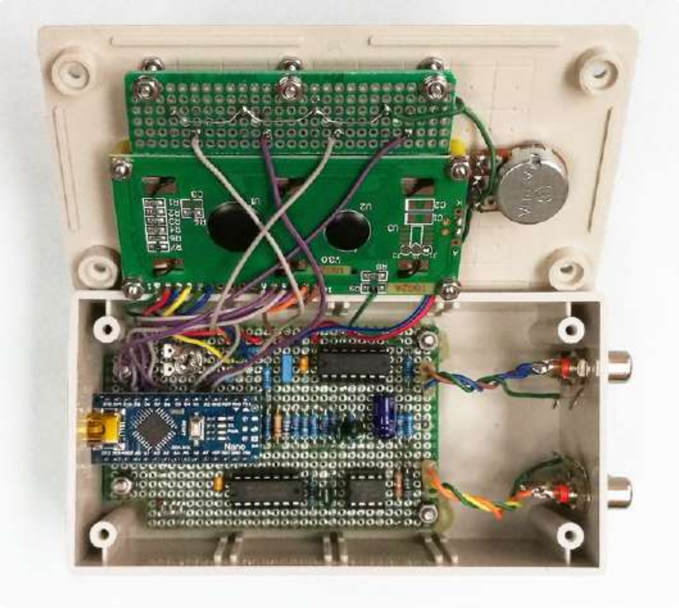


Figure 5: The complete set of electronics mounted in a plastic case. The Arduino board is placed on one side so that the USB port can be reached from outside.

In pulse mode, the pulse width is adjusted with the potentiometer. The value is quantized to give a "rounded" accurate setting and is not continuously variable. The resolution is dependent on the frequency. At high frequencies, the pulse width values are coarse and at lower frequencies they are spaced finer. Pulse width is displayed as a percentage of the period.

If the *ALT* button is held pressed, it enables alternative functions for the other three buttons. To select the frequency range, hold down the *ALT* button and press the *RANGE* button to toggle between high and low ranges. To adjust the output signal level in all modes except pulse, hold down the *ALT* button and press the *LEVEL+* or *LEVEL–* button to scroll through the available levels.

Construction

As already mentioned, no PCB has been designed for this generator. Instead, the electronics are built on a prototyping board with isolated soldering pads (0.1″ grid) as shown in **Figure 5**. Using a double-sided plated-through board is ideal for this purpose. I used a piece of 55 × 90 mm (2.2″ × 3.5″ with 21 × 34 pads). If the enclosure has posts in each corner for screws to fix the top cover, cut out two corners of the board so that the short side fits flush with one end of the box. The Arduino Nano USB socket is located on this side.

The board is fixed to the bottom of the enclosure on top of four 6 mm spacers. It's a good idea to glue the spacers inside the enclosure so that they stay put while screws are inserted. Four pushbuttons are soldered onto another piece of protoboard, which is also mounted inside the top cover, below the LCD module. Before soldering the buttons on, drill 2 mm holes in the board at the center location of each button.

In **Figure 6**, you can see how the components are soldered on top of the prototype board. The wiring on the bottom side of the board is a bit more complicated: **Figure 7** shows the basic wiring using bare tinned copper wire around the perimeter of the board, following the outermost pads, steering clear of mounting holes. This is the GND rail. Tie all connections to GND with bare tinned copper wire, as shown in the diagram. A wire gauge similar to resistor leads (~24 AWG, 0.5 mm) is preferable. Most of the connections to V_{CC} (+5 V) can also be made

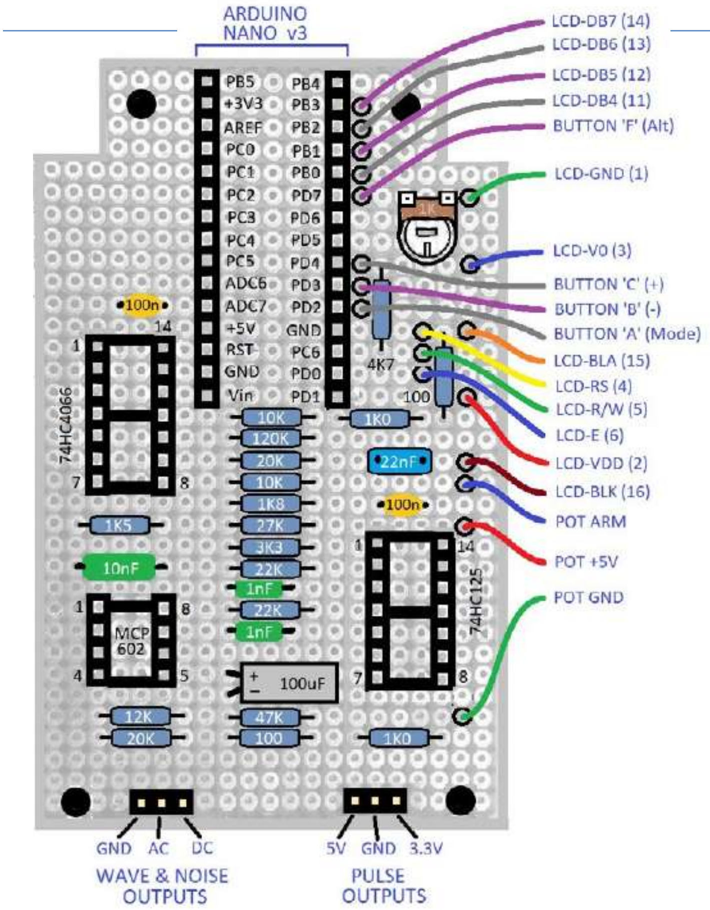


Figure 6: Top side of the prototype board with the exact position of the components and sockets.

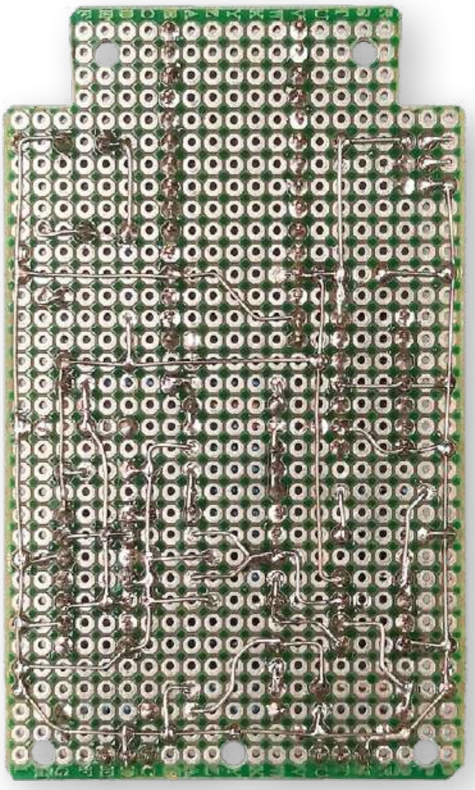


Figure 7: Bottom side of the prototype board with basic wiring with tinned copper wire.

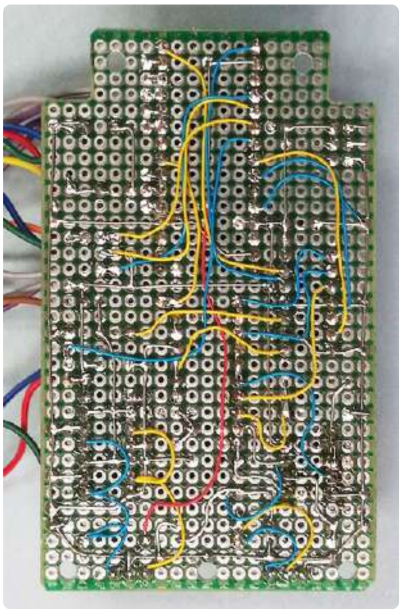


Figure 8: Bottom side of the prototype board with additional stranded wires and hookup wires.

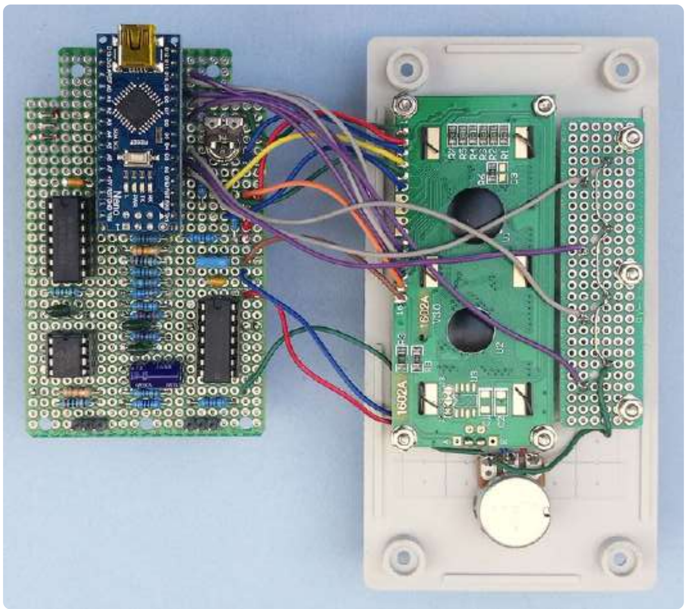


Figure 9: Complete prototype board with Arduino Nano connected to the display, buttons and a potentiometer ready for a first test.

using bare wire. The rest of the connections can be done with short pieces of thin insulated wire, as shown in **Figure 8**. Different colors may help in keeping track.

After the circuit board is well-checked against the schematic diagram, connect the LCD module, pushbuttons, and control pot to the board. The easiest and most reliable method is to solder stranded wires directly to the board. Then, fit the Arduino Nano module and ICs in their sockets. Set the LCD contrast adjust trim-pot to about halfway. Mount the circuit board in the box as shown in **Figure 9**.

Now, it's time for a "smoke test." Connect the Arduino Nano to a 5 V DC power source via the USB port. The LCD backlight should be on. If the DC voltages are correct and you don't see any smoke, your board should be ready for testing with firmware.

Firmware Installation

The firmware was developed using Microchip Studio for AVR and SAM Devices (formerly Atmel Studio). The primary reason for choosing this IDE over Arduino is that the waveform generator hardware design is incompatible with available Arduino code libraries. In particular, the 1602A LCD interface scheme (MCU I/O pin assignment) does not appear to be supported by an Arduino library.

Programming the target ATmega328P device can be achieved without Microchip Studio and without any hardware programming tool. An alternative way is described in the **Own Firmware with Microchip Studio** frame. The Arduino Nano has an on-board USB-serial bridge and a resident AVR bootloader. The Windows application AVRDUDE [1] communicates with the bootloader via USB to program firmware into the flash memory of the MCU. After connecting your Arduino Nano to a PC, open the Windows Device Manager utility and click on *Ports (COM & LPT)*. You should see the Nano USB-serial device listed. Note the associated COM port's number and use this number when editing the AVRDUDE command line. The programming is done using Windows *Command Prompt*. A single command line does the trick. However, the required AVRDUDE command line is quite long, and it needs to be customized to suit your particular PC software setup and COM port allocation.

Copy and paste (or type) the command line below into a text editor such as Notepad. Be sure to delete any line feeds so that the command is all on a single line. The file *Program Nano.bat* containing these commands is inside the archive file, which can be downloaded for free from the Elektor Labs page for this article [2]. Select *Word Wrap* from Notepad's *Format* menu to make it more readable. Here is the command line:

```
avrdude.exe -C avrdude.conf -p atmega328p -c arduino -P
COM4 -b 115200
-U flash:w:nano-wave-gen-v1.5.hex:i
```

Some of the bold highlighted fields will need to be edited to suit your PC setup and the firmware revision to be programmed: Replace COM4 with the actual COM port your Arduino Nano is connected to, as you found using Windows Device Manager. Annoyingly, the allocated COM port can change when the USB cable is unplugged and reconnected. This is because a USB-serial connection is made via a "virtual serial port" — not a physical port. The name of the firmware file from [2] will be something like *nano-wave-gen-v1.5.hex*. Edit the AVRDUDE command line so that the hex filename (highlighted) is the same as the one downloaded. Save the edited command file as *Program Nano.bat*.

Note: Some cheap Arduino Nano board clones use a non-standard Baud rate for the serial bootloader, which may be 57,600 baud. If AVRDUDE outputs an error message, try using a different Baud rate. In this case, replace "115200" with "57600" in the command line.

Operating Principle

The technique employed is called "wavetable synthesis." A wavetable is an array of data representing one full cycle (1 period) of a waveform. The array elements contain the signal amplitudes of every sample. The MCU outputs the samples at a fixed frequency called the "sample rate." This sample rate needs to be (much) higher than the maximum frequency component in the output signal. As a rule of thumb, one should use a factor of four. To generate audio signals up to 10 kHz, for example, a sample rate of ≥40 kHz is needed. For convenience, due to the MCU clock rate of 16 MHz, the lower sample rate of 32 kHz is chosen.

The firmware uses ATmega328P on-chip Timer/Counter module TC0 in PWM mode to generate arbitrary waveforms in the audible and sub-audible frequency range. The wavetable contains the duty cycle of the PWM output signal. This still-digital PWM signal passes a low-pass filter which averages the PWM signal to an analog voltage level representing the waveform stored in the wavetable, albeit with a phase lag.

The filter cutoff frequency of 8 kHz removes the 32 kHz pulse frequency as far as is practical, leaving only the desired audio frequency components present in the output signal. The PWM signal output combined with the low-pass filter works as DAC, thereby saving the cost of a separate DAC chip. **Figure 10** shows the principle of operation in displaying the PWM signal from a wavetable containing sine data in the upper screen and the resulting wave form after the low-pass filter in the lower part of the screen.

Timer Functions in the Firmware

Not only the ready-to-program hex file, but the complete source code can be found in the firmware archive file at [2]. If you are interested in how this piece of software transforms an Arduino Nano into a waveform generator, please have a look into the code.

The `TC0_setup()` function initializes Timer-Counter TC0 in "dual slope" (aka phase correct) PWM mode to generate a pulse waveform with variable duty cycle on pin OC0B (= PD5). Timer TC0 Output Compare register A is used to generate a periodic interrupt at the sample rate of 32 kHz, so the interval between two IRQs will be 31.25 µs. The interrupt service routine (ISR) needs to fetch, process and output one sample every 31.25 microseconds. This is quite challenging for a low-end 8-bit MCU, but with a clock frequency of 16 MHz, up to 16 instructions can be executed in 1 µs. Hence, up to 16 × 31.25 = 500 instructions can be executed in the ISR. This is enough computing power and there is no need to use assembly language.

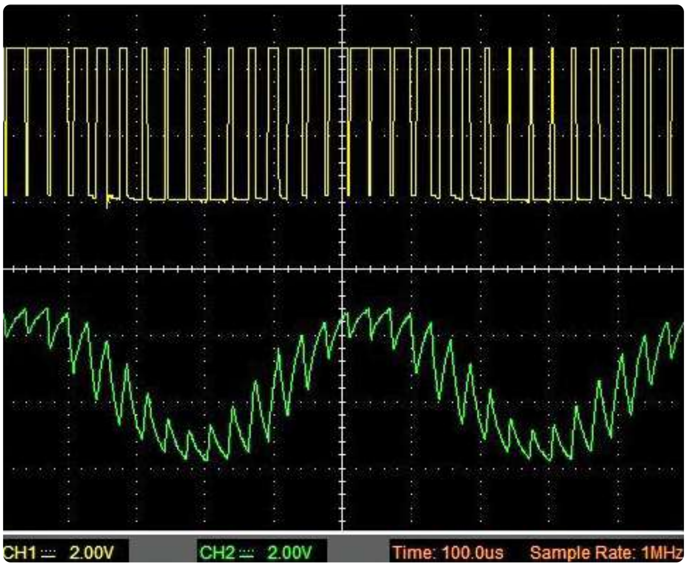


Figure 10: The upper screen shows the PWM signal of a 2 kHz sine wave. After low-pass filtering, the waveform on the lower screen is the result.

Of course, the MCU can't spend all its time in the ISR, otherwise nothing else would get done in the main (background) loop, so we need to make sure that the ISR's maximum execution time is much less than the IRQ interval. A good choice is to use half of the IRQ interval as execution time for the ISR.

The timer clock prescaler is disabled (N = 1) to get the highest clock rate. With a clock rate of 16 MHz, register OCR0A is set to 249 for a timer period of 250 clocks (in each direction — up and down) so the PWM output frequency will be exactly 32 kHz. Output Compare B register OCR0B is used to generate a PWM output on pin PD5/OC0B. The PWM duty cycle is varied proportionally to the sample amplitude, updated every sample (every 31.25 µs) by the timer Output Compare A ISR).

When the counter register (TMR0) reaches its "TOP count" (OCR0A = 250), the count direction is reversed and an IRQ flag (OCA) is raised. At the same time, the PWM output pin, PD5/OC0B, is set high. When TMR0 matches the OCR0B value, the level on pin PD5/OC0B is set to Low automatically. The PWM pulse duty cycle is therefore proportional to the OCR0B value. The duty cycle obviously cannot exceed the period, so the maximum duty cycle (100 %) is obtained by writing 249 into OCR0B. For a more detailed explanation of PWM waveform generation using an AVR microcontroller, please refer to the ATmega328P datasheet [3], in the section on 8-bit Timer/Counter TC0.

Wavetable Algorithm

The wavetable algorithm works by fetching samples from a wavetable at the fixed sample rate of 32 kHz. The frequency of the output signal is determined by the "distance" (phase angle) between points taken from the wavetable. The shorter the distance, the lower the output frequency. When the table index of the next sample to be fetched goes beyond the end of the table, the index is adjusted so that it wraps around to a point within the table maintaining the correct sample distance.

Let's call the distance between sample points the "Phase Angle Step." For most practical purposes, the Phase Angle Step needs to be a real number, i.e., having both an integer and fractional part. Floating-point arithmetic would make the task easy, but is too slow for the ATmega328P because there is no floating-point hardware integrated. This is why fixed-point arithmetic is used instead.

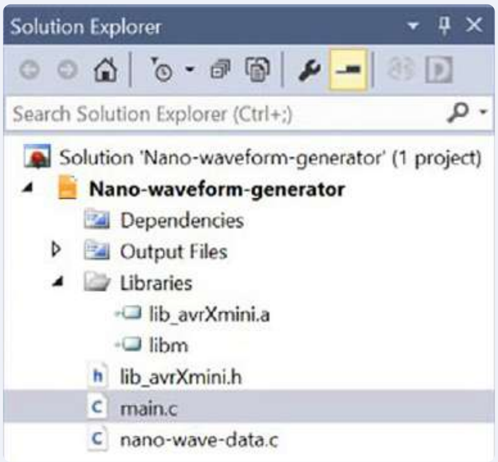
Fixed-point arithmetic uses fast integer arithmetic with integer words (or long words) composed of two "bit fields" representing the "integer" and "fractional" parts of a number. A 16-bit fixed-point number could be composed of an 8-bit integer part and an 8-bit fractional part. For unsigned numbers, the range would be 0.0 to 255.99 (approx. decimal equivalent). For signed numbers, the range would be -127.99 to +127.99. The resolution is 1/256 ≈ 0.004 which is good enough for low-precision applications.

Own Firmware with Microchip Studio

To customize the waveform generator firmware to suit your own requirements, first download and install Microchip Studio for AVR and SAM Devices (IDE) on your Windows PC or Mac [4]. The tutorial mentioned at [5] will probably assume a "target platform" using a Microchip or Atmel development board with a hardware programming tool, e.g., Atmel AVRISP mk2 or similar device. But there is no need for one of these tools to load firmware into the Arduino Nano MCU. You can create a "software programming tool" in Microchip Studio (see "Option 2" further down.)

After downloading the firmware archive from [2] and extracting it, copy all the files to a new folder named *Nano-waveform-generator* on a local drive, preferably in the projects folder created by Microchip Studio in your *Documents library*. Now connect your Arduino Nano to a USB port on your PC. Then open Microchip Studio and choose Open Project... from the start page. Navigate to the project folder you created and click the file name *Nano-waveform-generator.atsln*.

On the right side of the IDE window, find and click the *Solution Explorer* tab (see **Figure 11**). Expand the *Libraries* list and ensure that the *lib_avrXmini.a* library is present. The library file contains pre-built functions to support various peripherals commonly used in ATmega328P MCU applications.



Solution 'Nano-waveform-generator' (1 project)
- **Nano-waveform-generator**
  - Dependencies
  - Output Files
  - Libraries
    - lib_avrXmini.a
    - libm
  - lib_avrXmini.h
  - main.c
  - nano-wave-data.c

Figure 11: Screenshot of the **Solution Explorer** *panel.*

To edit a source file if it is not already open in the editor, click the file name in the Solution Explorer panel. To keep the file open in the editor, click the pin icon in the file tab at the top-right-hand side of the editor window. The file tab will move to the left-hand side of the window and remain open. Feel free to peruse the library header file, *lib_avrXmini.h*, but do not modify it unless you intend to make changes to the library. The C source code is free to modify any function in the library or to add more functions.

When done with your modifications, build your solution, fix any compilation errors, and build again. Then follow the instructions below to transfer the firmware to your Arduino Nano. There are two options; in both cases, it is assumed that the AVRDUDE distribution files can be found in a folder named *Nano Wave Gen* on your PC local drive.

Option 1:
Use Windows Command Prompt as before. You can use the same method as described under the heading *Firmware Installation*. You just need to copy the generated hex file into the folder *Nano Wave Gen*. Every time you hit *Build Solution*, Microchip Studio will replace the Hex file in the project folder, in a sub-folder named Debug. If your project folder is named *Nano-waveform-generator*, the hex file will be written into the sub-folder *Nano-waveform-generator\Debug*. Before programming, rename the generated Hex file so that it is identical with the file name contained in the Windows command file *Program Nano.bat*.

Option 2:
Create a "Programming Tool" in Microchip Studio. Click in the menu *Tools* → *External tools*. You should see a dialog box asking for some parameters: In *Title*, write: *Program Nano* or any other name you prefer. In *Command*, write: *C:\Nano Wave Gen\avrdude.exe*. In *Arguments*, write (all on one line):

```
-C "C:\Nano Wave Gen\avrdude.conf" -p
 atmega328p -c arduino -P COM4 -b 115200 -U
 flash:w:"$(ProjectDir)Debug\$(TargetName).
 hex":i
```

Replace COM4 with the actual COM port your Nano board is connected to, and be aware of the fact that some Arduino Nano clones only support lower Baud rates. Tick the *Use output window* box. Click OK. Done.

You should see a new option, *Program Nano*, in the *Tools* menu. After the code is built, it can be programmed into the Nano board simply by clicking on the *Program Nano* item in the *Tools* menu.

A 32-bit fixed-point number is often composed of a 16-bit integer part and a 16-bit fractional part. For signed numbers, the range is roughly -32,000 to +32,000. The resolution (accuracy) in this case is roughly 1/65,000 ≈ 0.00001, which is more than adequate for our waveform generator. The accuracy of the output signal frequency is limited only by the MCU clock.

In our wavetable oscillator, the Phase Angle Step and Phase Angle (wave sample point) are represented by 32-bit fixed-point numbers. The integer part of the Phase Angle is used as an array index to fetch a sample point from the table. The formula relating "Phase Angle Step" to oscillator frequency is:

PhaseStep = OscFreq × (TableSize / SampleRate)

where *PhaseStep* is the distance between sample points in the wavetable, *OscFreq* is the required output frequency (Hz), *TableSize* is the total number of samples in the wavetable, and *SampleRate* is the sample rate (32 kHz).

Although wavetables may be any arbitrary size, within the constraints of program memory, the table size should be consistent with the output signal resolution. As a rule of thumb, the size should be around the same order as the maximum value of the samples in the wavetable. For example, if the output sample values are 8-bit unsigned numbers, a suitable wavetable size would be 256 samples. A larger table size gives the same benefit as interpolation between sample points, i.e., reduced waveform distortion.

Signal Aliasing

With some knowledge about Digital Signal Processing, you may be familiar with the term "aliasing" and its causes and effects. Aliasing is the distortion of a sampled waveform that results when a frequency component in the analog input waveform is too high in relation to the sampling rate. The distortion manifests itself as unwanted frequency components being introduced into the sampled signal.

For example: If a signal contains a frequency component of 16 kHz and is sampled at 20 kS/s, there will be new and unwanted components in the (reconstructed) output signal at the sum and difference frequencies, i.e., 4 kHz and 36 kHz. The 4 kHz component is well within audible range and would be a noticeable artifact. Pure sine waves at frequencies below half the sample rate will not suffer any aliasing effect. Waveforms rich in high-order harmonics like square, pulse, and sawtooth waves are very susceptible to aliasing.

Likewise, aliasing occurs with wavetable oscillators when a waveform represented by sample points from a wavetable contains frequency components above half the sample rate, except where the ratio of output frequency to sampling frequency is a rational number. Careful choice of wavetable size relative to the sample rate can eliminate aliasing at certain output frequencies. In this waveform generator, the table contains 640 samples and selection of fixed output frequencies was chosen to avoid aliasing effects.

Finally

As said before, this is not a full-blown DIY project but more a technical demonstration of what can be done with a cheap Arduino Nano. Feel free to change the code and the circuit to your own need or use it as a source of inspiration for your own projects.

220641-01

About the Author

Michael Bauer is a veteran engineer interested in electronic music tech (e.g., wind instruments and DSP sound synthesis). He lives in Victoria, Australia.

Questions or Comments

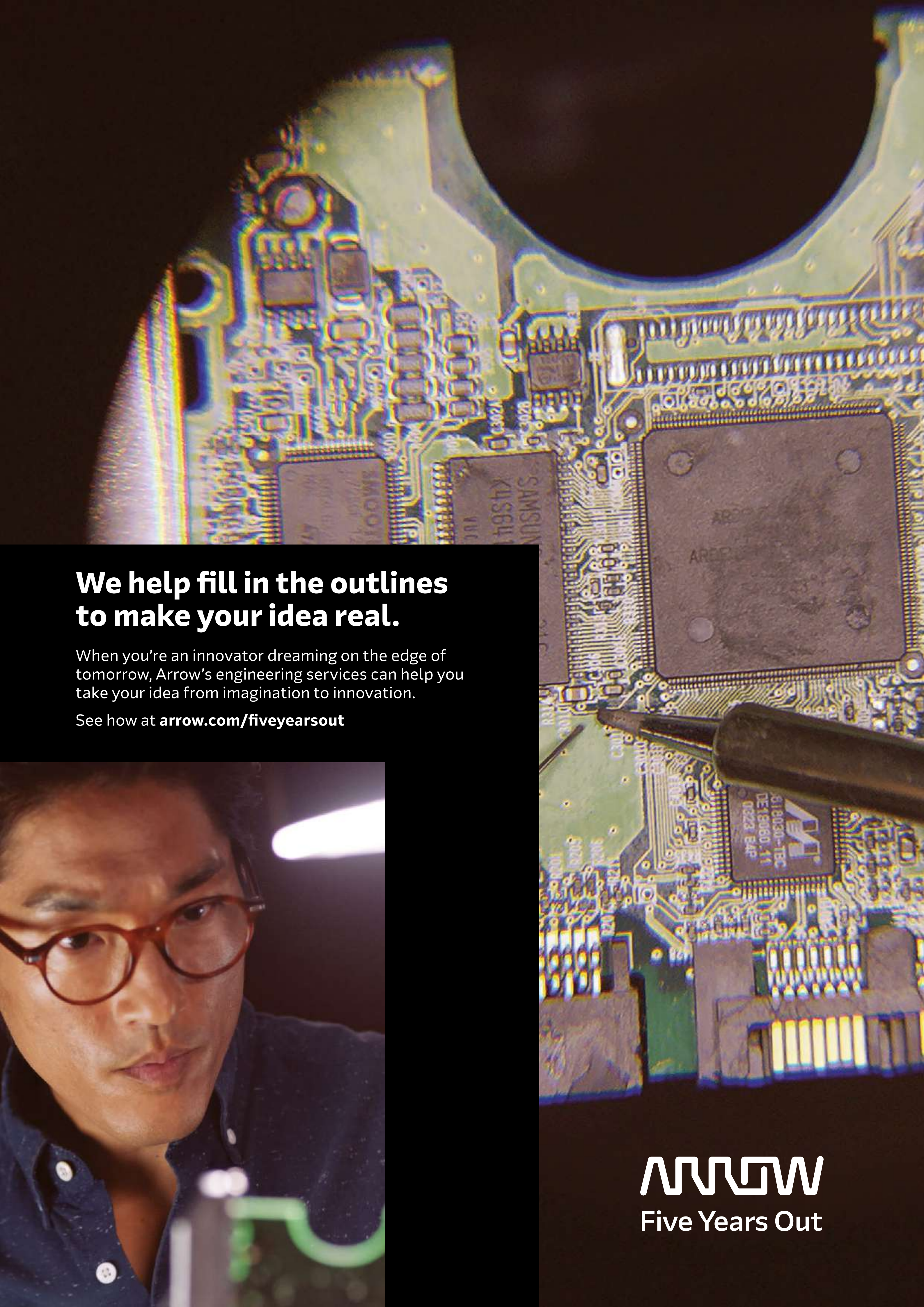
Do you have technical questions or comments about this article? Email Elektor at editor@elektor.com or contact the author at mjbauer@iprimus.com.au.

Related Products

- **OWON HDS1021M-N 1-ch Oscilloscope (20 MHz) + Multimeter**
  https://elektor.com/18778
- **Robert Sontheimer, *Arduino & Co – Measure, Control, and Hack, Elektor 2023***
  https://elektor.com/20243

WEB LINKS

[1] AVRDUDE on GitHub: https://github.com/avrdudes/avrdude
[2] Project downloads at Elektor Labs: https://elektormagazine.com/labs/nano-waveform-generator
[3] ATmega328P Datasheet: https://tinyurl.com/4w2uzpy2
[4] Microchip Studio: https://microchip.com/en-us/tools-resources/develop/microchip-studio
[5] Getting Started with Microchip Studio: https://tinyurl.com/39bsmzbd



We help fill in the outlines to make your idea real.

When you're an innovator dreaming on the edge of tomorrow, Arrow's engineering services can help you take your idea from imagination to innovation.

See how at **arrow.com/fiveyearsout**

ARROW
Five Years Out

Solar-Powered Christmas Garland

An Eco-Friendly Solution for Garnishing Your Balcony

By Laurent Labbe (France)

If you own one of the many variants of battery-powered LED decorative garlands and want something "smarter" and more environmentally friendly, you will find what you need in this article. It is a smart, microcontroller-based circuit that can recharge its own Li-Po battery with a photovoltaic cell and go into operation automatically when darkness falls, with many attractive lighting effects for your balcony!

During recent years, every Christmas we've been garnishing our balcony with a simple China-made garland that we bought for pocket money, powered by two simple AA batteries and featuring 20 LEDs. Every year, I thought about expanding on it, making it funner and without the need for the two alkaline cells. Finally, thanks to my son, Louis, on the mechanical part, I finalized this modding project.

In my mind, the project would have had the two batteries replaced with a dummy casing, with two cylinders simulating them; one for a lithium cell and one for the electronics, as visible in **Figure 1**.

Furthermore, to avoid recharging it, I was planning to fit a small solar panel power supply (**Figure 2**) that, other than simply charging the battery, would have also detected day and night conditions, so the device could stay in charge state during the day, and blinking during the night.

Hardware

For the blinking function, as you may see in the schematics in **Figure 3**, I've chosen a Microchip 12LF1572 8-pin microcontroller driving a 2N2222 NPN transistor. This switch shorts the LED's series array cathodes to ground to realize the blinking effects. A pushbutton allows the user to select the desired pattern of blinking among the available ones, including a random flashing option.

The solar panel (5 V, with a size of 65×35 mm) output is connected to a dividing network made by fixed resistor R3 and an adjustable resistor, R1, whose terminal is connected to the AN input of the microcontroller's A/D converter (see Figure 3). To charge the battery, a simple, but not "orthodox" solution was adopted: a Schottky diode directly connected to the cell. Traditionally, in such cases, a voltage regulator IC with 4.2 V control would have been needed to avoid exceeding the LiPo battery's maximum charge voltage, but here, due to the low current generated by the solar cell, the risk of damaging the circuit or the battery is low, and the overall level of efficiency is high enough to the scope of this project.

To detect day and night, the solar panel output voltage is being monitored through a AN input channel of the microcontroller on pin 6. The purpose of the resistor in series with the trimmer is to protect the µC against voltages higher than 4.2 V that might come from the solar cell.

The diode is connected directly to the battery and is located before the switch. This is to avoid any high voltage from the solar cell to the µC in case of high sunshine and the power switch being off. In that case, the solar cell could deliver more than 5 V, which is the absolute maximum rating of the microcontroller. The latter is programmed using Microchip's PICkit 3 debugger and the appropriate connector.

Software

The software is written in C, under MikroC Pro, and it's available here [1]. After initialization of the variables, parameters, and registers, a setup loop starts. If, in this loop, the pushbutton is being held at power on, the user may change the number of hours that the garland operates at night. One flash indicates one hour, two flashes represent two hours, and so on, up to a maximum of twelve hours. The default value is six hours. When the button is released, the timer will be set to that value. However, this value will not be saved, and will need to be set again every time the unit is turned on. This aspect of the software functionality should be improved, indeed.

After this phase, the endless loop starts. It first checks the status of the push button. Each press will change among the various LED flashing patterns, which are:

- Random pattern between 300 ms (off) and 1500 ms (on), changing on each cycle with the `srand()` function.
- 200 ms on, 1000 ms off
- 800 ms on, 1000 ms off
- 500 ms on, 500 ms off
- 300 ms on, 300 ms off, 300 ms on, 1000 ms off

At the end of each pattern, there's a wait cycle of 2,000 ms.

The loop will activate the pattern only if the voltage on the solar panel is 0 V. When daylight is present, no pattern will be activated and the loop will just wait 2 s each time, consequently.

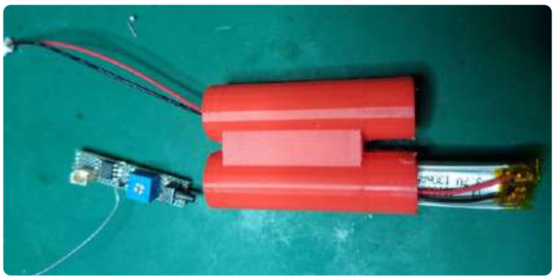


Figure 1: The 3D-printed housing for both the electronics and the LiPo cell.

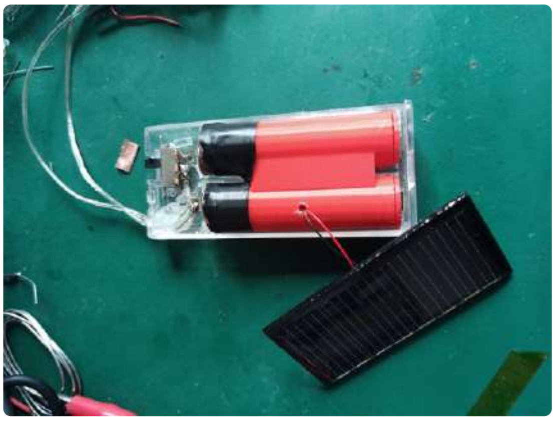


Figure 2: The small solar panel for charging the LiPo cell in the daytime.

All the long delays utilize the `tempo()` function. This function uses the `sleep()` function and the watchdog feature, which is adjusted to 2000 ms. This is the best solution for power-saving purposes.

Implementation

After realizing the first functioning prototype, with both sides shown in **Figure 4a** and **4b**, respectively, my son created the dummy casing for the two AA batteries using Tinkercad [2], already shown in Figure 1. On the positive

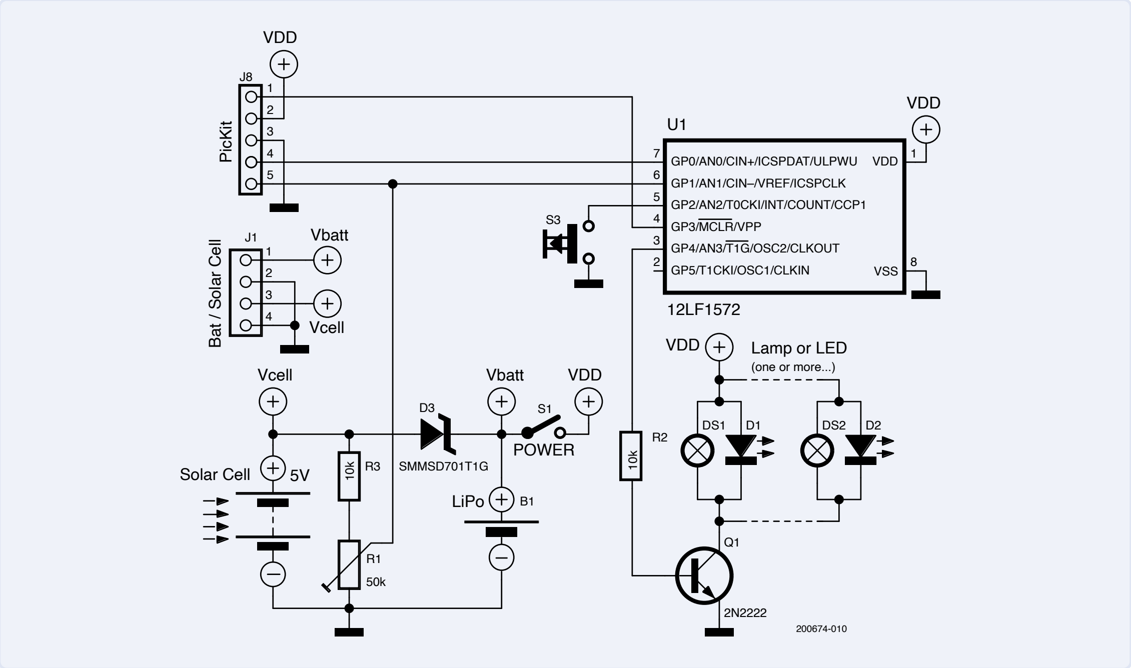


Figure 3: The simple circuit diagram for this project.

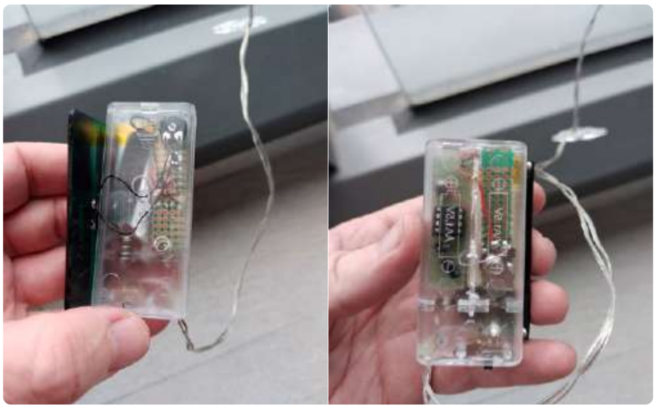


Figure 4: The first functional prototype in front (left) and back (right) perspectives.

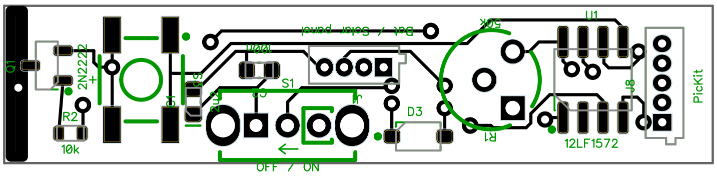


Figure 5: The AA-battery-sized single-sided PCB layout.

connection side, we put the lithium polymer battery, of any size and capacity that would fit inside it. The positive lead of the battery is connected to the positive wire of the original garland contact and, with another wire, to the PCB, whose layout is visible in **Figure 5**.

The negative lead of the battery is connected directly to the PCB. The collector of the 2N2222 (you can use any other NPN switch transistor) is connected to the negative lead of the original garland.

The PCB can be installed in the dummy case's second cavity, with wires to the battery and the solar panel (second connector).

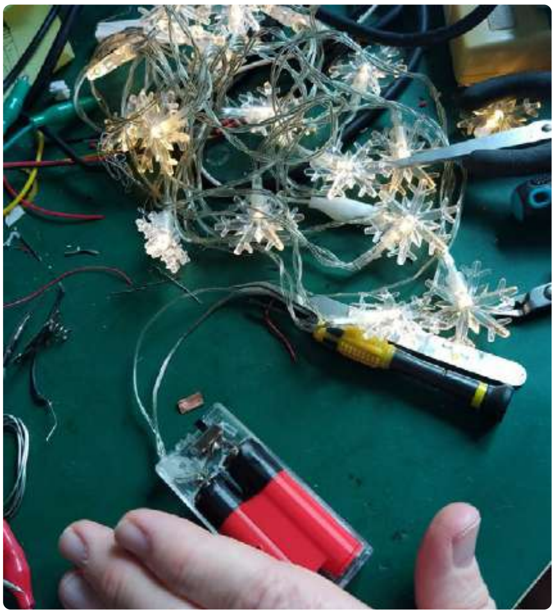


Figure 6: The finished project, lighting the garland.

To connect to the original garland contact, I used some copper film glued on the dummy battery. The final result of my hard work is visible in **Figure 6**, with the LED garland nicely lit by my project.

200674-01

WEB LINKS

[1] PCB and software package for this project:
https://elektormagazine.com/labs/solar-power-christmas-garland
[2] Autodesk Tinkercad Website: https://tinkercad.com

About the Author

Laurent is 60 years old, has lived in Asia for more than 20 years, and has been passionate about electronics since he was 10 years old. He's read many magazines, including Elektor since 1978. Laurent started playing with microcontrollers in the 1980s, the 8051 family and, more recently, with the PIC family and ESP32, whenever a network connection is needed. He has built many amplifiers, both solid-state (Elektor) and tube-driven (300B SE), as well as speaker systems and DACs. He has been working in the mobile phone industry for more than 30 years (hardware, software, operation, quality) for a French company.

Questions or Comments?

Do you have technical questions or comments about this article? Please write to the Elektor team at editor@elektor.com.

Related Products

- **Bert van Dam, *PIC Microcontrollers* (Elektor E-book)**
  www.elektor.com/18093
- **Seeed Studio Solar Panel for Outdoor Environments**
  www.elektor.com/19131
- **Diamex LED Player M**
  www.elektor.com/19911

Every week that you don’t subscribe to Elektor’s Newsletter (e-zine) is a week with great electronics-related articles and projects that you miss!

Stay Informed, Get Creative, and Win Prizes – Subscribe Now to be Part of it!

www.elektor.com/ezine

What can you expect?

Editorial
Every Friday, you’ll receive the best articles and projects of the week. We cover MCU-based projects, IoT, programming, AI, and more!

Store
Don’t miss our Elektor Store promotions. Every Tuesday and Sunday (and occasionally Thursdays), we’ll have a special deal for you.

Partner mailing
You want to stay informed about the ongoing activities within the industry? Then this e-mail will give you the best insights. Non-regular but always Wednesdays.

elektor
design > share > earn

USB Killer **Detector**

Better Safe Than Sorry

By Carlos Guzman (USA)

USB killers are malicious devices disguised as USB flash drives. When plugged into a computer, tablet, or phone USB port, the device will try to destroy the USB port and more. The device described here lets you identify such pen drives before they can do any harm.

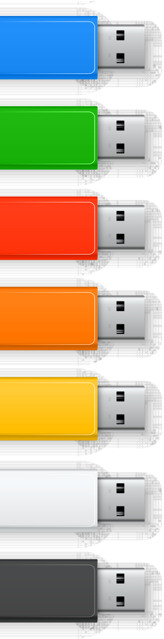
USBKill [1] is a pen-testing USB flash drive that transforms a USB port's 5 V into about –200 V, leading to the destruction of, for example, a personal computer, laptop, or even a phone. As they have a logo on the outside clearly indicating that they are pen-testing devices that must be handled with care, they do not present a real danger.

More dangerous, however, are the so-called "USB killer" sticks that cannot be distinguished from normal flash drives. They look like any ordinary flash drives, but can cause serious damage to devices they're plugged into.

My Personal Experience

My father buys many items online. He purchases electronics components, as he is an electronics technician, but he also brings home unnecessary products such as an excess of nuts, bolts, and tools. He has a room at home where I can barely walk.

Some time ago, I had an old movie on my computer that I wanted to watch on the TV, so I was looking for a USB flash drive. As I did not have one at hand, I went to my father's electronics room and I found one. I connected it to my computer, went around the table, and discovered that the computer had shut off. Only the weak sound that had started just after connecting the flash drive remained.



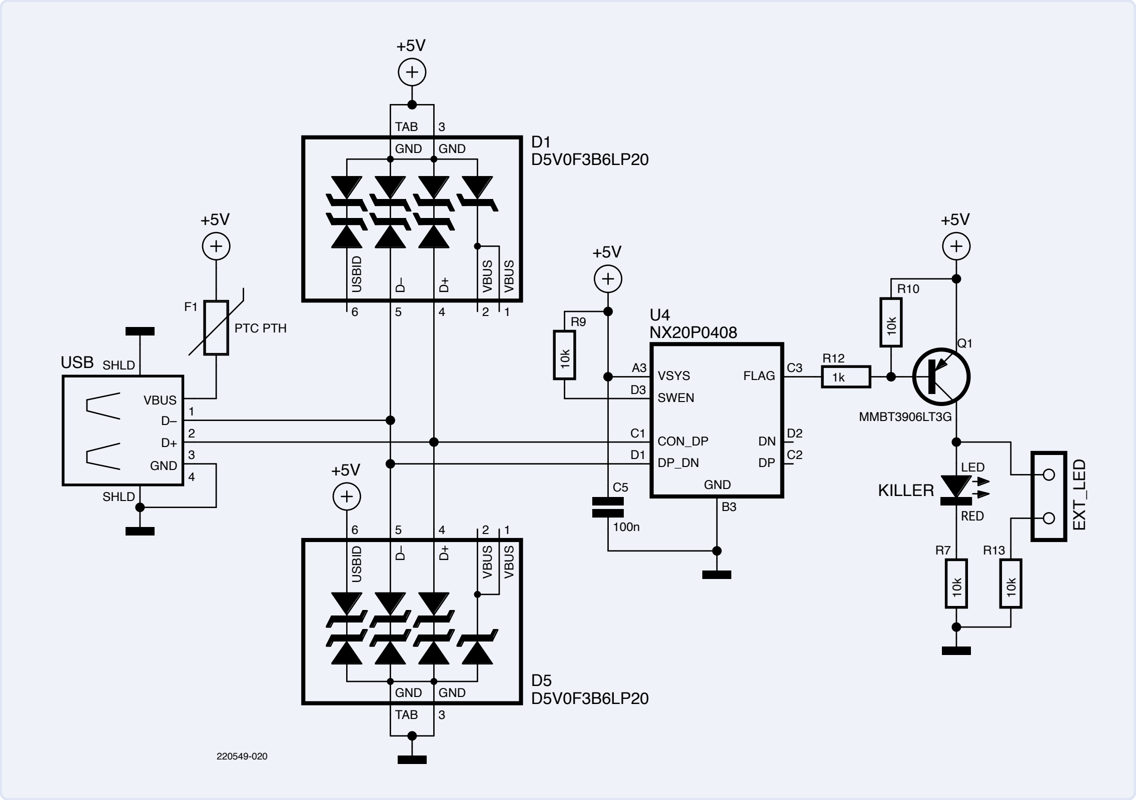


Figure 1: The schematic of the USB Killer Detector without power supply. The red LED signals danger.

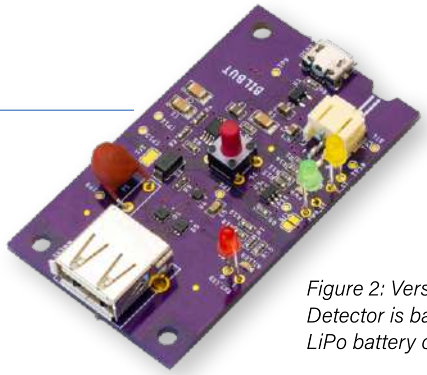


Figure 2: Version 1 of the USB Killer Detector is battery powered and packs a LiPo battery charger plus boost converter.

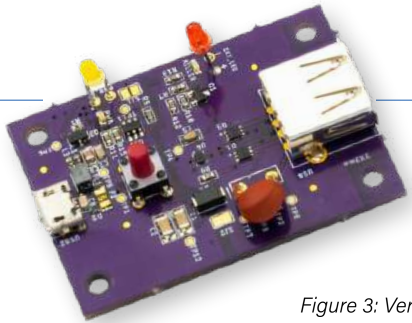


Figure 3: Version 2 requires an external 5 VDC power supply.

After realizing that my computer did not turn on anymore, I checked all cables and the mains supply. Everything was working as usual, except my computer, which did not turn on. After a few days of investigation, the cause turned out to be the USB flash drive from my father's room. It did not have any indications on the outside of what it was. It looked like any other USB stick but was, in fact, a USB killer.

Why?

Like me, you may be wondering why USB killers exist. The answer seems to be mainly along the lines of "just because." So, my father and I decided to develop a device that can detect USB killers before they do damage to a computer.

Our Solution

A USB killer uses the USB port's 5 V to generate bursts of about –200 V pulses on the USB data lines. This is a very effective technique for destroying USB ports and more. The USB Killer Detector makes use of a special USB data line protection IC, an NX20P0408 from NXP [2] that flags a fault when something is not right on these pins. The flag signal controls a red LED. If the LED goes on, you should destroy the flash drive you just plugged in.

Additional TVS diodes ensure that the detector itself doesn't get killed. A PTC provides even more protection by switching off the killer stick if it's just too dangerous to handle.

Two Versions

We developed two versions of the USB Killer Detector:

1. A portable battery-powered version that you can carry anywhere you, as shown in **Figure 2**. It includes a LiPo battery charger and a boost converter to turn the battery voltage into 5 V for the USB stick.
2. An externally powered version because it is difficult to ship LiPo-based batteries and devices that contain them, and we wanted to sell the USB Killer Detector online. It requires a Micro USB cable of providing 5 V (**Figure 3**). Use a phone charger for power, not your computer.

The design files for both versions of the device can be found at GitHub [3].

A Final Word

Do you think that the USB Killer stick is an urban legend? Well, think again. In 2019, in the USA, a man was sentenced to 12 months in prison and a $58,471 fine after intentionally destroying 66 computers with a USB Killer at a school in Albany [4].

The video at link [5] has totalled over 20 million views over the past five years, which means ten thousand views per day, every day. A similar video [6] has been viewed almost two million times over the last two years. And have you ever searched for USB Killer sticks at online retailers? If you do, you will find that they are easy to buy.

USB Killer sticks are real, and you have been warned.

If you would like to help me fight USB Killers, join me on [7].

Thanks to my father, Alberto Guzman, for the time we spent together working on developing the device.◀

220549-01

Questions or Comments?

Do you have technical questions or comments about his article? Email the author at instant.devices@yahoo.com or contact Elektor at editor@elektor.com.

Related Products

- **Elektor Archive 1974-2022 (USB Stick)**
  www.elektor.com/20372
- **The Elektor Power Supply Collection (USB Stick)**
  www.elektor.com/20451

WEB LINKS

[1] USBKill: https://usbkill.com
[2] NX20P0408 datasheet: https://nxp.com/products/power-management/load-switches/single-chip-usb-data-lines-protection-solution:NX20P0408
[3] Full project details at GitHub: https://github.com/instantdevices/USB-Killer-Detector
[4] USA court case: https://justice.gov/usao-ndny/pr/former-student-sentenced-destroying-computers-college-st-rose
[5] Example video 1: https://youtu.be/Y1o1nwlpY4I
[6] Example video 2: https://youtu.be/FLkXiP1fC2A
[7] Help me fight USB Killers: https://opencollective.com/usb-killer-detector
[8] This project at Elektor Labs: https://elektormagazine.com/labs/usb-killer-detector-v1

A Simple CNCed Enclosure

With Autodesk Fusion 360 for Personal Use

By Vaclav Krejci (Czech Republic)

Custom cases for your DIY projects don't necessarily have to be 3D-printed — they could also be machined out of metal such as aluminum, with a highly professional result. Even if you don't have your own CNC machine, the manufacturing cost is affordable, the software tools are free, and the steps to follow are simple.

In this tutorial, you'll learn how to create a custom enclosure for the Rotary Encoder Module from DFRobot. With parts common to many enclosures, this should be a great starting point for creating a custom one for your own project.

For modeling, we use Autodesk Fusion 360 for Personal Use. This software is free and intended for hobbyists and non-commercial work. There are some limitations compared to the full (paid) version, but none of them should affect our work.

Tools used:

- Autodesk Fusion 360 (Personal Edition) [1]

Components used:

- Rotary Encoder Module from DF Robot [2]
- Arduino UNO (but any Arduino board will work with the module)

Steps

1) The first step in creating any 3D model is starting with a 2D drawing called a Sketch (**Figure 1**). In Fusion 360, click the *Create Sketch* button on the toolbar and select a plane upon which to place the sketch. Since our enclosure will be extruded to the top, placing the sketch on the bottom plane in the 3D space makes sense.

2) Using the caliper, I measured the rotary encoder module size to be 37×37 mm. Let's create a new rectangle to visualize the PCB's size (**Figure 2**). Go to *CREATE* → *Rectangle* → *Center Rectangle*. Once the tool is active, place the first point on the origin. Use a keyboard to type one dimension (37), jump to the next edit field with the *Tab* key, and type the second value (37). To complete the creation process, hit the *Enter* key.

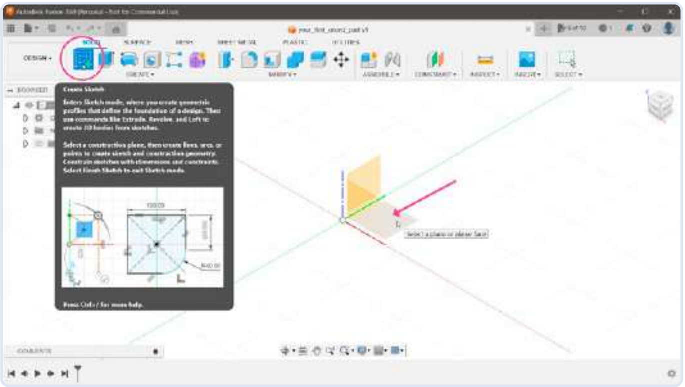


Figure 1: First steps for placing your Sketch. (Source of all screenshots: Autodesk)

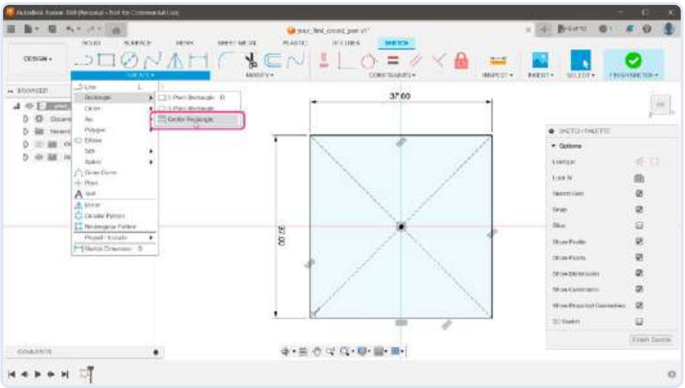


Figure 2: Defining the rectangle sizes.

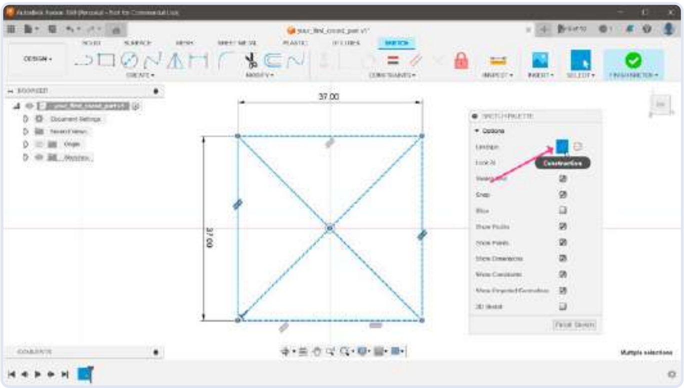


Figure 3: Changing lines definition into *Construction* mode.

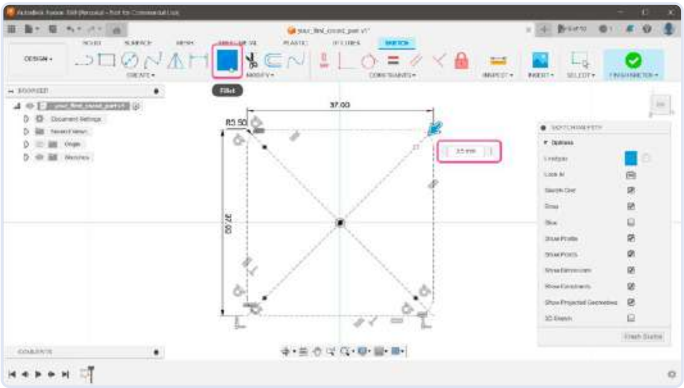


Figure 4: Rounding the enclosure's corners.

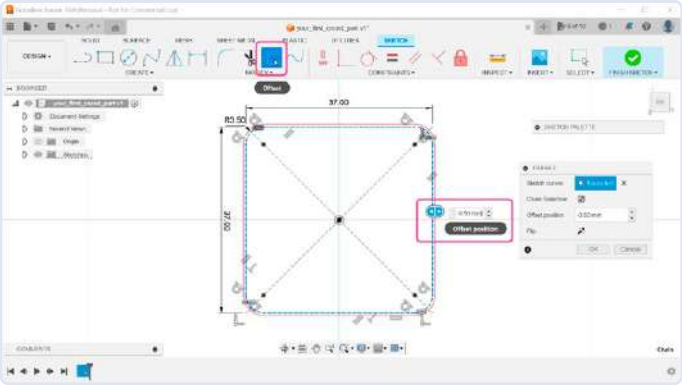


Figure 5: Offsetting the enclosure to get an easier fit for the encoder board.

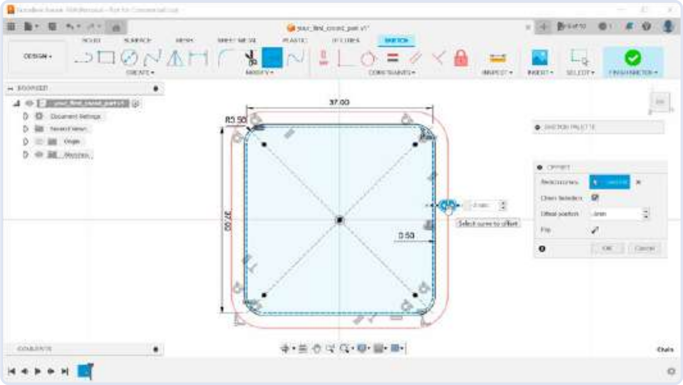


Figure 6: Setting the walls' thicknesses.

3) We won't use this shape to extrude the enclosure since we need space around the PCB. It makes sense to select all the lines and turn those into construction lines by clicking the *Construction* button in the *Sketch Palette* dialog. This tells Fusion 360 that we use these only as helper lines (**Figure 3**).

4) The PCB has rounded corners, and it would be nice if our enclosure matched those (**Figure 4**). A rough measurement shows that the corner radius is about 3.5 mm. In the toolbar, select the *Fillet* tool, and click on all the outer lines individually. Enter 3.5 mm for *Fillet Radius*.

5) We must offset the rounded rectangle to ensure the PCB fits freely (**Figure 5**). In the toolbar, select the *Offset* tool and enter a value of -0.5 mm.

6) Another offset is required to build the walls of our enclosure (**Figure 6**). Repeat the step again, but enter the offset of -3 mm. This will later create 2.5 mm-thick walls.

7) We have sufficient shapes to start extruding them. Close sketch editing by clicking the *Finish Sketch* button, and then select the *Extrude* tool from the toolbar (**Figure 7**). Click the inner rounded rectangle on the sketch and enter a distance of 2 mm.

8) We wanted to extrude the walls, but they suddenly disappeared. Don't worry; it's just the sketch that's hidden. In the browser in the top-left corner (**Figure 8**), open the *Sketches* group and make *Sketch1* visible. Now you can select the *Extrude* tool from the toolbar and click the outer rounded rectangle. Enter the distance of 15 mm, which will be the height of our enclosure.

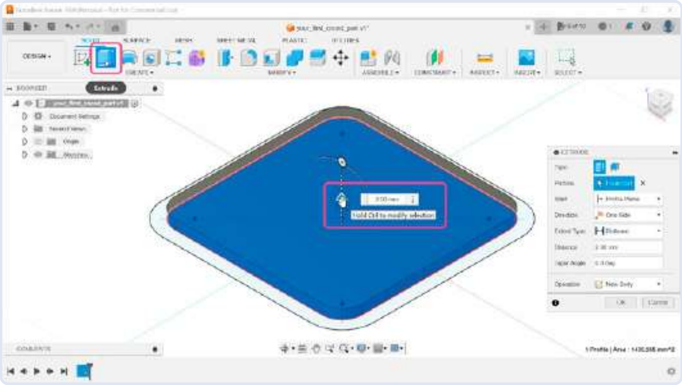


Figure 7: Starting the extrusion!

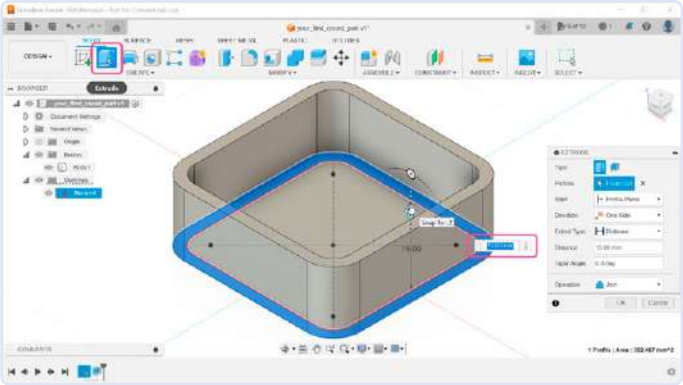


Figure 8 : Setting the enclosure's height.

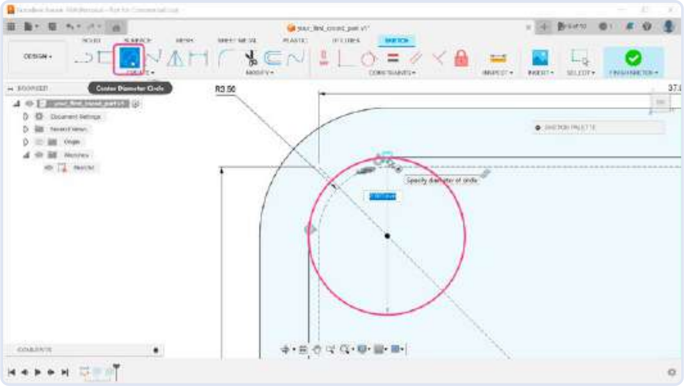


Figure 9: Positioning the center for one of the supports.

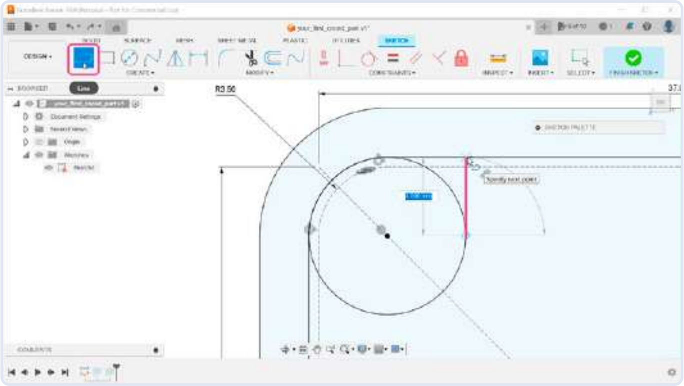


Figure 10: Shaping the support using the *Line* tool.

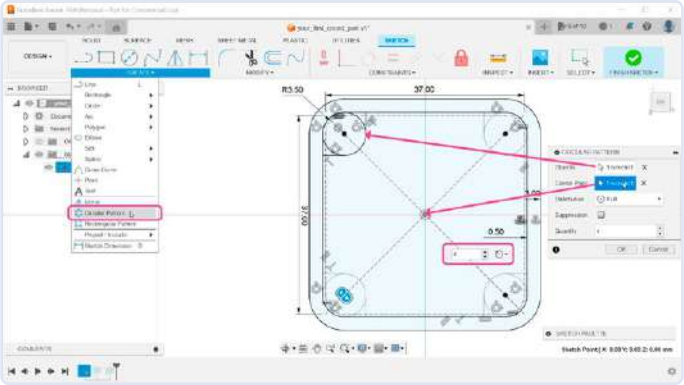


Figure 11: Replicating the support for the other three corners.

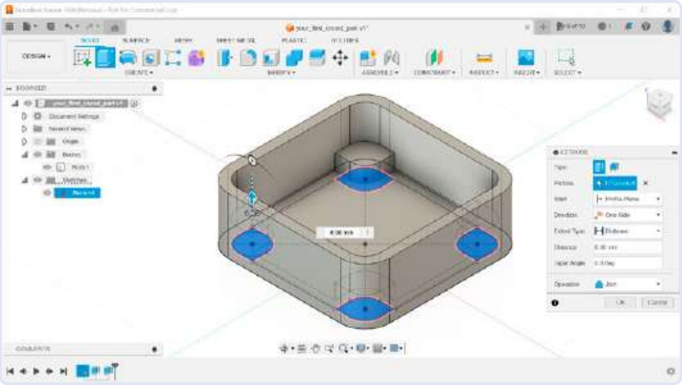


Figure 12: Extruding the four supports.

9) The module should sit on supports of some kind, and it would be nice if those supports had holes for screws. For these supports, we need another sketch, but, instead of creating a new one, we can reuse the one we have as well. To return to editing mode, double-click the *Sketch* item in the browser window. The holes in the PCB are 3 mm in diameter; we need to make the supports bigger than that. For ease of manufacturing, we can even make them big enough to touch the walls of the enclosure. Select the *Center Diameter Circle* tool from the toolbar and draw a circle in the top-left corner — large enough to touch the inner shell (8 mm diameter) (**Figure 9**).

10) The area between the circle and the wall is very thin (**Figure 10**) and will be difficult to manufacture. We can close it by drawing more shapes with a line tool. In the toolbar, select the *Line* tool and move the cursor over the circle's center. After that, keep moving the mouse cursor to the right side until you reach the circle. In that position, draw the first point of the line. Move the mouse cursor to the top until you reach the wall. In that position, draw a second point on the line. Make sure that the movement is somehow straight — that will tell Fusion 360 to add any necessary constraints automatically.

11) Three more supports are needed, but instead of drawing those manually, we can use some tools to duplicate the current support. An obvious choice might be the *Mirror* tool (*CREATE* → *Mirror*). However, a one-click solution, in this case, is the *Circular Pattern* tool, also located in the *CREATE* menu. With the tool active, we need to select the circle and two lines (**Figure 11**). Click the *Select* button next to the center point, and select the origin point. For *Quantity*, enter a value of 4.

12) At this point, we can again finish editing the sketch by clicking the *Finish Sketch* button, and we can extrude the supports. With the body visible, it might be hard to select the required faces from the sketch. An easier solution is to hide *Body1* temporarily in the browser, select the *Extrude* tool (**Figure 12**), and select the shapes from the sketch. However, before extruding, you need to show *Body1* for the supports to be connected with this body. The distance should be 6 mm, and, in the *Extrude* dialog, ensure that the *Operation* is set to *Join*.

13) It would be nice if those supports included holes for screws. We already have the position of these holes (the center points of the circles), but they are in the different Z positions. The points are on the bottom of the enclosure, but we need those to be on top of the supports. To move those points, we will create a new sketch by selecting *Create Sketch* button, and, for the position of this sketch, click the top plane of one of the supports (**Figure 13**).

14) To copy those points on our new sketch, select *CREATE* → *Project/ Include* → *Project* (**Figure 14**). Click on all four points and confirm using the OK button. If you rotate the 3D view, those new points are now hovering over the original sketch.

15) Close the sketch, show *Body1*, and select the *Hole* tool from the toolbar. Click on all four points and set the Hole (**Figure 15**). Set *Hole Type* to Tapped. *Thread Type* should be *ISO Metric profile* and *Size* should be 3.0 mm. As for the depth, we don't want the holes to go all the way through, and, since the supports are extruded 6 mm over the ground plane, a 4 mm depth should work fine.

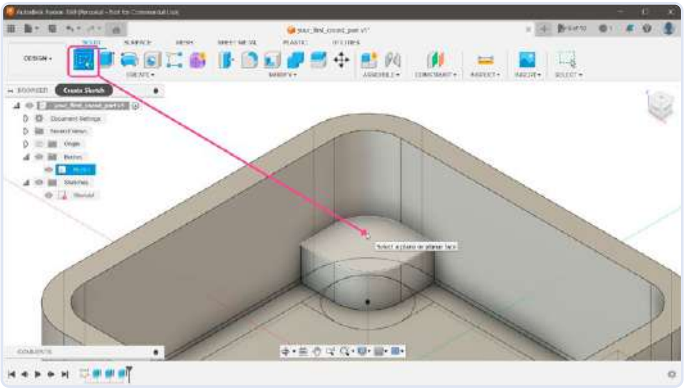


Figure 13: Setting up the hole in the support.

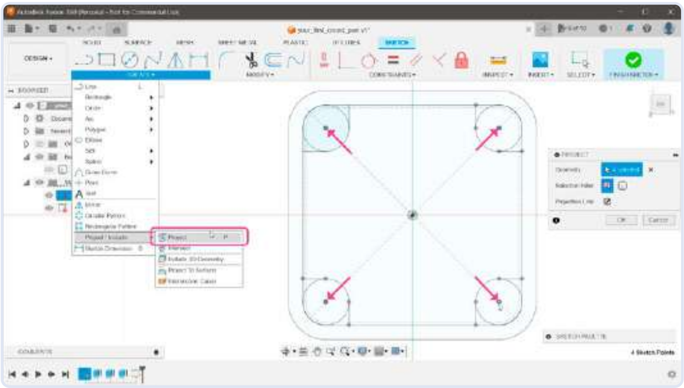


Figure 14: Replicating the operation for the other three corners.

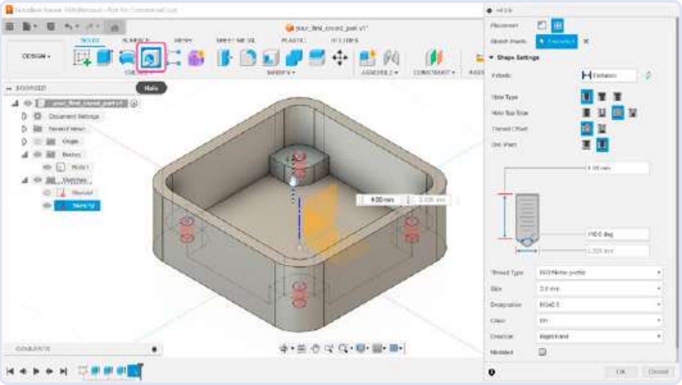


Figure 15: Setting depth and thread types for the holes.

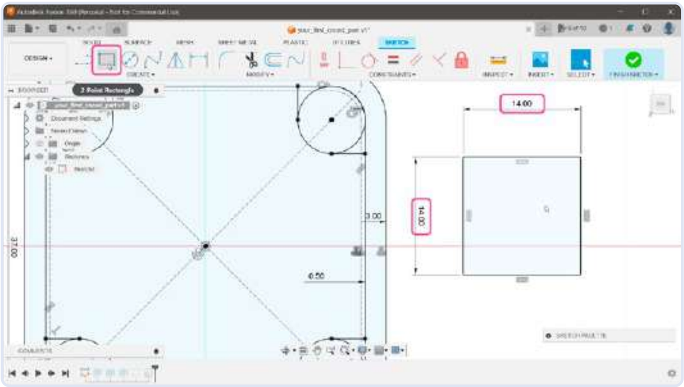


Figure 16: Preparation of the two rectangular holes on the bottom of the enclosure.

16) We need two rectangular holes on the bottom of the enclosure to access the connectors on the bottom of the PCB. The connectors are slightly different in size, but a 11×14 mm hole will work for both. We need to count with the wall thickness (3 mm), which sets the hole size to 14×14 mm. For drawing those holes, we can reuse our original sketch. Open it for editing and select the *2-Point Rectangle* tool (**Figure 16**) from the toolbar. Draw a 14×14 mm rectangle anywhere on the canvas.

17) To move the rectangle to the correct position, select the *MidPoint* (**Figure 17**) constraint from the *Constraint* group in the toolbar. First, click the right line of the newly created rectangle. Next, click the right line of the wall of the enclosure. This will ensure those two lines are center-aligned and will move the rectangle to the correct position. Repeat the steps for the hole on the left side and close the sketch.

18) Rotate the 3D view to see the bottom of our enclosure. Select the *Extrude* (**Figure 18**) tool and then select the newly created rectangles. Enter a distance of 6 mm to match the height of the supports. *Operation* should be automatically changed to *Cut*.

19) The enclosure is almost ready. We have a basic shape, supports with threaded holes, and holes on the bottom of the enclosure to access the connectors. At this point, it might be a good idea to tweak the shape slightly to make manufacturing simpler. Since the milling tool has a certain diameter, it is impossible to create sharp inner edges. We can easily fix this by adding a little rounding using

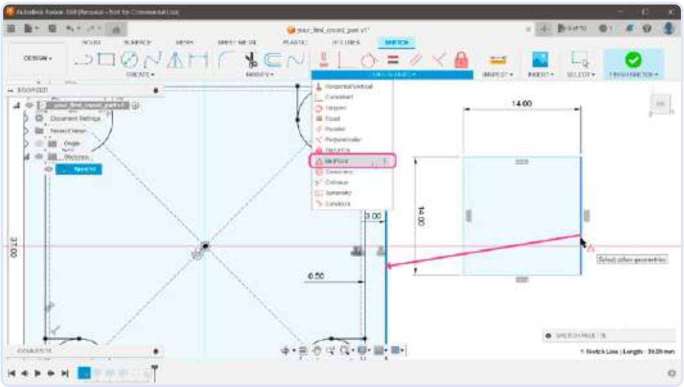


Figure 17: Centering the rectangle with the *MidPoint* function.

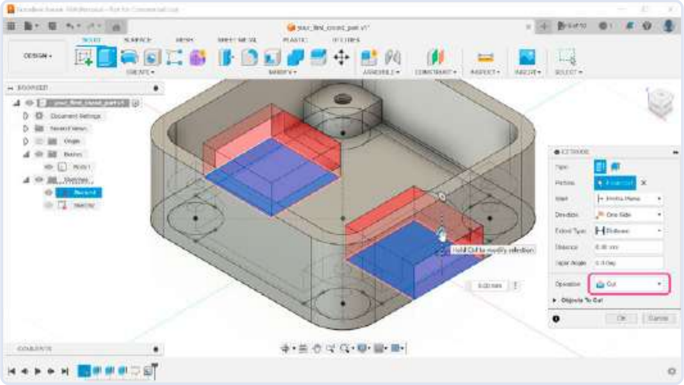


Figure 18: Extruding the two holes.

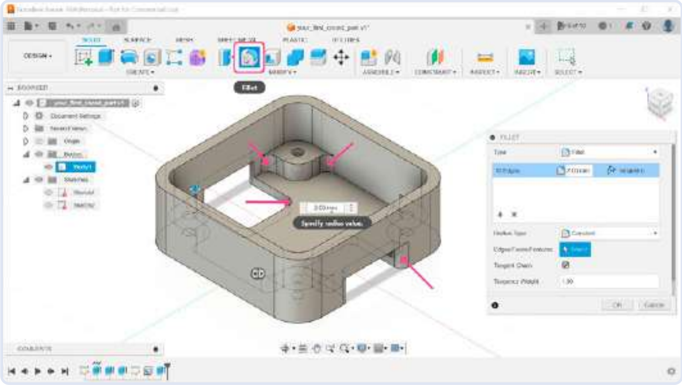


Figure 19: Rounding up the internal edges.

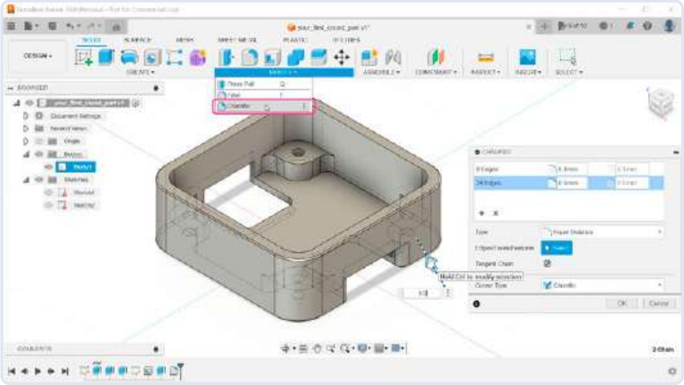


Figure 20: Smoothing the external edges.

the *Fillet* (**Figure 19**) tool from the *Modify* group. Once the tool is active, select all the hard inner corners and enter some small rounding value — for example, 2 mm.

20) The sharp edges on the outside could be smoothened using the *Chamfer* tool. Select it (**Figure 20**) from the *Modify* submenu and select the outer edges of the enclosure. For the top edge, I have used a minimal value of just 0.3 mm. For the bottom edge, a higher value of 0.5 mm was used. You can chamfer multiple edges with different values simultaneously by using the *Add Selection Set* button in the *Chamfer* dialog.

21) At this point (**Figure 21**), the enclosure is ready to be manufactured. A common file format for 3D drawings is a STEP File (.step), which could be exported using the *File* → *Export* menu option. With this file, the chosen service provider [3] will mill the real, solid-metal prototype out for you. It is visible with other components in **Figure 22** in two different finishes, with the encoder inserted (**Figure 23**), and finally (**Figure 24**) as a completed unit. A comprehensive video on this project's steps can be found at [4].

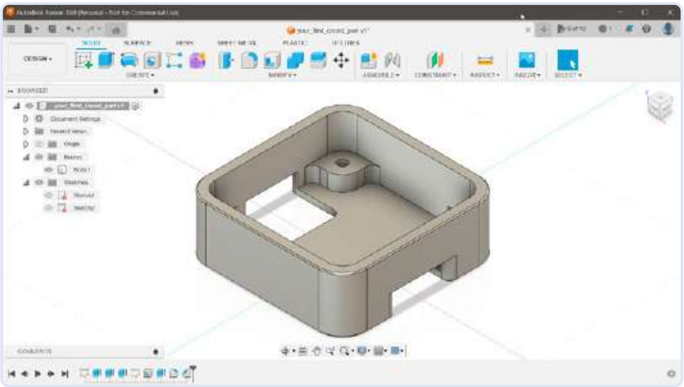


Figure 21: The enclosure ready for manufacturing.

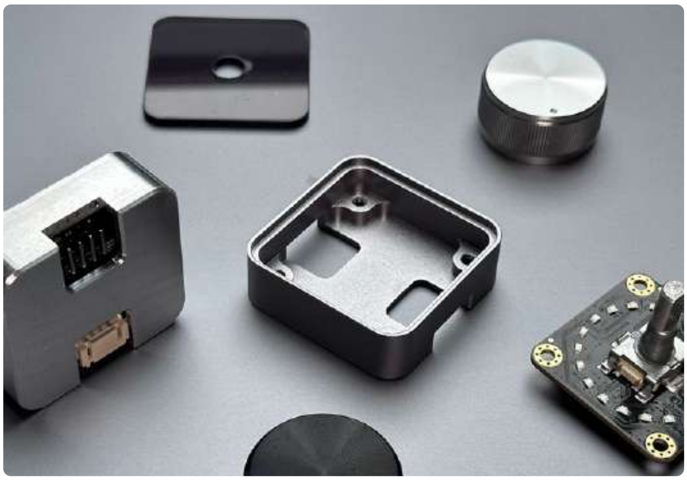


Figure 22: The enclosures with the other parts needed, just before final assembly.

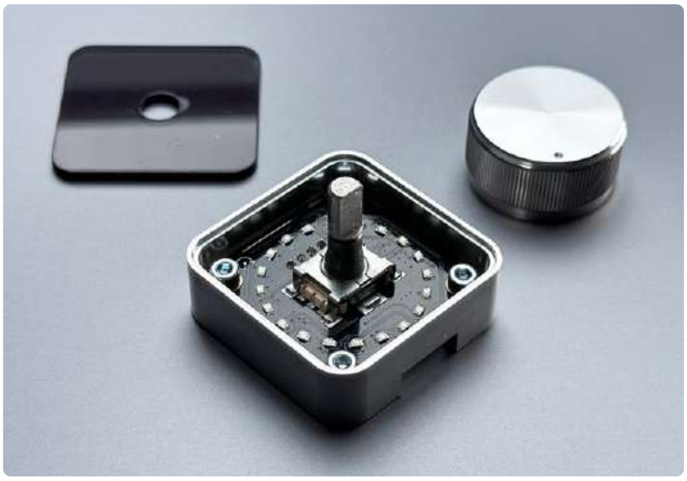


Figure 23: The machined housing with the encoder installed.

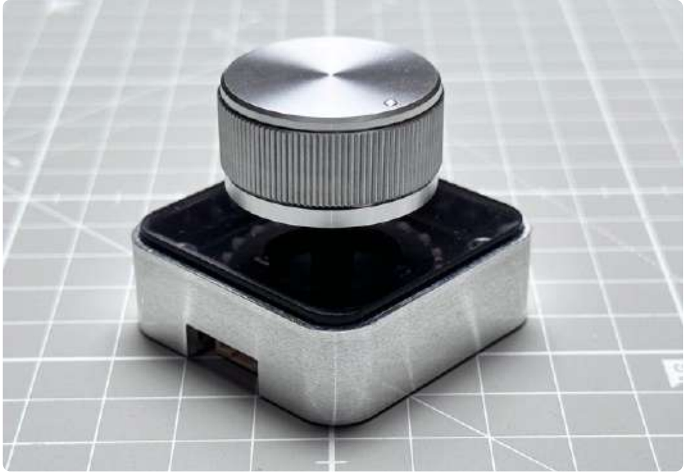


Figure 24: The fully assembled prototype.

It's That Simple

Autodesk Fusion 360 [1] is a complex application with dozens of tools, but hopefully this tutorial showed you that to create a simple enclosure, you need to use only a few of these tools. The same result can usually be achieved in multiple ways, so as long as it works for you, you don't need to worry much if it's the right approach. Every manufacturer will be willing to address any small problems, and if you're not sure about something, ask. The enclosure from this tutorial was manufactured by PCBway [3], costing $37 plus shipping. This includes bead blasting of the CNCed part (for a smooth surface) and anodizing (for better surface quality and a nice aesthetic). If you are fine with the raw metal look, you can save a few dollars. And, if you can find a local manufacturer, you can probably save on shipping as well. Good luck creating your own enclosures for your great projects!

230361-01

About the Author

Vaclav Krejci published his first (online) tutorial article in 1998 and continued to do so in the following years. After hundreds of tutorials about graphics, he decided to write a book, published in 2007, titled *GUI Design in Adobe Photoshop*. That led to great work opportunities but, unfortunately, put a long delay on any publishing activities. During Covid (in 2020), Vaclav discovered Arduino and his growing interest in electronics. He has also realized that capturing his learning and turning his findings into tutorials helps Vaclav finish projects that would otherwise be unfinished forever. This time, his main platform is YouTube, with video tutorials. Hopefully, Vaclav's projects will inspire others on the same journey.

Questions or Comments?

Do you have questions or comments about this article? Email the author at info@upir.org or contact Elektor at editor@elektor.com.

Related Products

- **Anet 4540 Desktop CNC Router Machine**
  https://elektor.com/20260
- **Arduino Uno SMD Rev3**
  https://elektor.com/19938

WEB LINKS

[1] Autodesk Fusion 360 Personal download page:
https://autodesk.com/products/fusion-360/personal
[2] Gravity 360 Encoder from DFRobot:
https://dfrobot.com/product-2575.html
[3] PCBway Website link for CNC: https://pcbway.com
[4] Tutorial Video on YouTube: https://youtu.be/EB8PISr4m4g
[5] Have some fun with Autodesk:
https://youtu.be/vNKx29qEyJ4

Elektor Webinars

DEC
14
2023
16:00 CET

Exploring the ESP32 P4 Module and the Guest-Edited Elektor Mag

Led by Anant Raj Gupta and Brian Tristam Williams

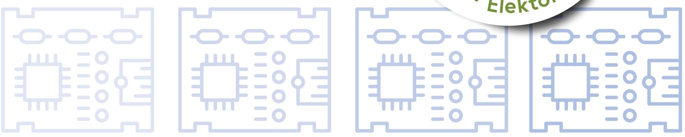
Get the chance to win an ESP32-C3-DevKitM-1 board!

www.elektormagazine.com/esp32w2

ESPRESSIF × elektor

Low-Volume Board Production

With and Without Assembly



By Saad Imtiaz (Elektor)

So, you have designed and tested an electronic circuit board, and now you want to produce it in (small) quantities. However, if you don't have the financial clout to just order 10,000 pre-assembled boards and store them somewhere from where you can sell them at leisure, what are your options?

The first option that comes to mind is assembling the boards yourself. This is very possible when the quantities are small, and the board doesn't have too many components (**Figure 1**). Now, this may sound simple, but there are a few things to think of before you start.

Start by choosing a board-manufacturing house. This can be a pooling service (see **Pooling** textbox) if you only need a few boards. But if you need (several) hundreds, then it will be more advantageous to compare quotes from different non-pooling manufacturers.

Source the Components

If you carried out the board design process in the correct order, you already have a component list or bill of materials (BoM) with, for each line, a complete manufacturer reference and an address of a company where you can buy the part. Having a one-stop shop for the entire BoM would be ideal, but, in most cases this shop does not exist. Various suppliers are required to source all the components, and each supplier will charge shipping and handling. Therefore, clever shopping will save you money. At the same time, it will cost you time, so balancing the two is important.

Assemble the Board

Once you have the PCBs and the components, it's time to assemble the boards. Do you have the tools required for this? A small soldering iron (**Figure 2**) may be fine for putting a prototype together, but is it comfortable enough for assembling twenty, fifty, or a hundred boards? And the parts — are they through-hole only, SMT only, or is it a mix? Usually, you want to avoid mixing mounting technologies because it complicates board assembly. All SMT is nice because the board can be soldered in an oven. However, placing such parts on the board is a delicate job. And, as solder paste is needed for this, did you order a stencil to apply it properly? How are you going to align the stencil on tens of boards? This will be easier when the PCBs are in a panel.

Testing

After assembling the boards, they must be tested. For this, you will need suitable equipment. Building a test rig, or a programmer, or even both, may be a good idea.

Selling and Shipping

If you've ever sold something on a site such as eBay, you'll know that selling something is more complicated than just finding a buyer for the product. Selling also includes invoicing, packing and packaging, and shipping, and these things all cost time and money.

What Are the Options?

As a matter of fact, the problems mentioned above must be solved for any production method you may choose, from one piece to mass production. And, since board manufacturing is a complex process, there exist many companies that offer services to make things easier for you. Of course, everything has a price, so it's up to you to find the right compromise between cost, time, and peace of mind.

PCBA Pooling

Some PCB manufacturers propose a board assembly service. They are a so-called PCBA houses, where the "A" means "assembly." So, instead of ordering a bare PCB from them, you can order a fully assembled board. That sounds convenient, right? It is convenient, but not only does it come at a cost, it also makes component sourcing a bit complicated. You have your BoM, and they have their inventory, and the two will not match on every line, and maybe not

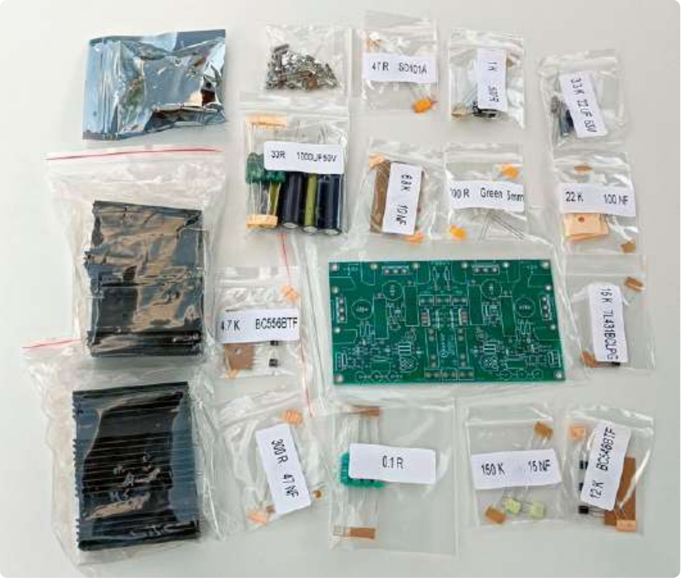


Figure 1: PCB with selected components.

Figure 2: Digital soldering station.

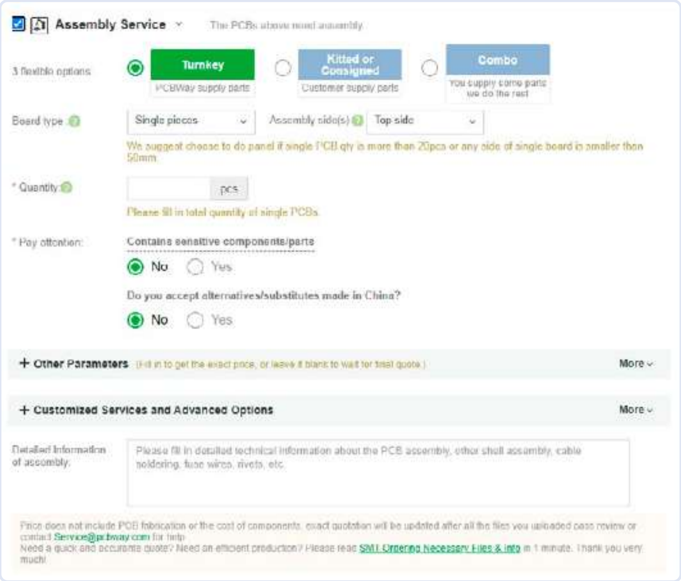
even on a single line. Therefore, you must carefully check every part on your component list against the parts that the assembly house suggests. This is a time-consuming task. And, of course, not every part will be in stock or even available, making costs and delays unsure. You can adapt your circuit design beforehand to the assembly service's component stock to make this step easier.

Most PCB pooling services also provide assembly services (**Figure 3**) along with component sourcing for you, which can be very helpful in reducing the hectic task of finding and then soldering the components; soldering a few PCBs can be fun, but more than a few can be tedious.

Just like ordering a PCB, ordering assembly is almost similar, but some additional details are required by the manufacturer, such as number of unique components, number of SMD and through-hole parts, and so on (**Figure 4**). It's good practice to add this information to the BoM and mention the types of components next to their name. This avoids any mistake by the manufacturer and saves time while fulfilling the order.

The manufacturer will provide you with a quote for the assembly service after you provide information about the component types and quantities, etc. This quote does not include the cost of components. After you submit your order, the manufacturer's support team will get involved and inform you of whether the components and their references in the BoM match with your PCB or not. They will then provide you with a quote for the components.

To save money on components, it's necessary to carefully review the price of the components and compare them to other sources such as Mouser, DigiKey, and AliBaba (**Figure 5**). It's reasonable to proceed with them if the components are at a fair price and the correct components are offered by the manufacturer. Usually, some components are not available, and some are priced above the normal cost. In this case, it is preferable to obtain the components from AliBaba resellers, as they typically offer a much lower price



The PCBs above need assembly

3 flexible options: ● Turnkey — PCBWay supply parts ○ Kitted or Consigned — Customer supply parts ○ Combo — You supply some parts we do the rest

Board type: Single pieces

Assembly side(s): Top side

We suggest choose to do panel if single PCB qty is more than 20pcs or any side of single board is smaller than 50mm

* Quantity: pcs

Please fill in total quantity of single PCBs

* Pay attention: Contains sensitive components/parts

● No ○ Yes

Do you accept alternatives/substitutes made in China?

● No ○ Yes

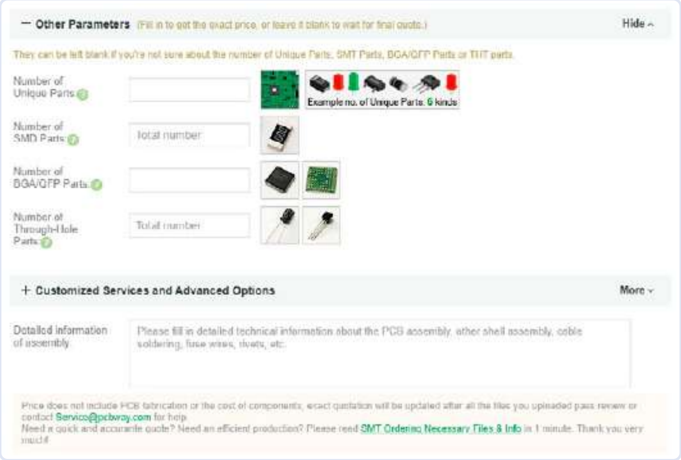
+ Other Parameters (Fill in to get the exact price, or leave it blank to wait for final quote.)

+ Customized Services and Advanced Options

Detailed Information of assembly: Please fill in detailed technical information about the PCB assembly, other shell assembly, cable soldering, fuse wires, rivets, etc.

Price does not include PCB fabrication or the cost of components, exact quotation will be updated after all the files you uploaded pass review or contact Service@pcbway.com for help.
Need a quick and accurate quote? Need an efficient production? Please read SMT Ordering Necessary Files & Info in 1 minute. Thank you very much!

Figure 3: Assembly service types on PCBWay.



— Other Parameters (Fill in to get the exact price, or leave it blank to wait for final quote.)

They can be left blank if you're not sure about the number of Unique Parts, SMT Parts, BGA/QFP Parts or THT parts.

Number of Unique Parts:

Number of SMD Parts: total number

Number of BGA/QFP Parts:

Number of Through-Hole Parts: Total number

+ Customized Services and Advanced Options

Detailed information of assembly: Please fill in detailed technical information about the PCB assembly, other shell assembly, cable soldering, fuse wires, rivets, etc.

Price does not include PCB fabrication or the cost of components, exact quotation will be updated after all the files you uploaded pass review or contact Service@pcbway.com for help.
Need a quick and accurate quote? Need an efficient production? Please read SMT Ordering Necessary Files & Info in 1 minute. Thank you very much!

Figure 4: Other parameters for assembly services on PCBWay.

Pooling

PCBs are manufactured in a so-called production panel. The same production panel can generate circuit boards for various customers with the same specs. We refer to this collection of related orders as pooling. Pooling promotes flexibility and lowers manufacturing costs.

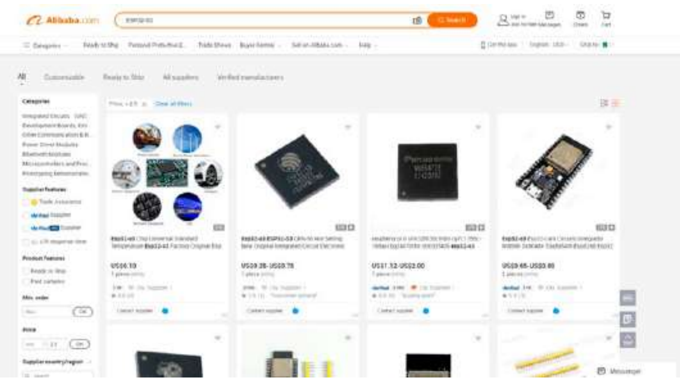


Figure 5: Chinese component vendors on Alibaba.

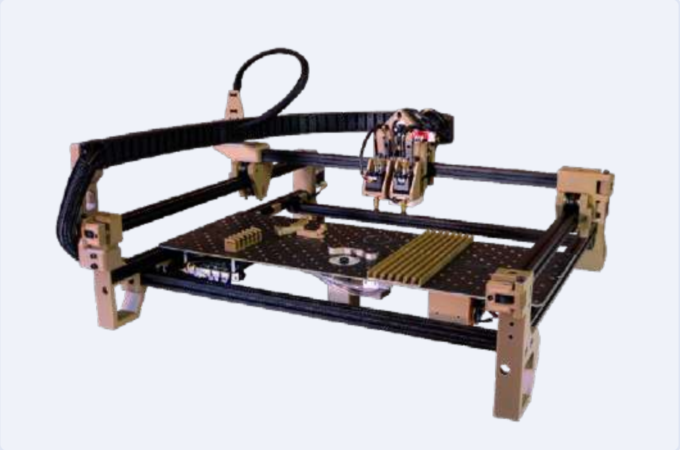


Figure 6: LumenPnP by Opulo.

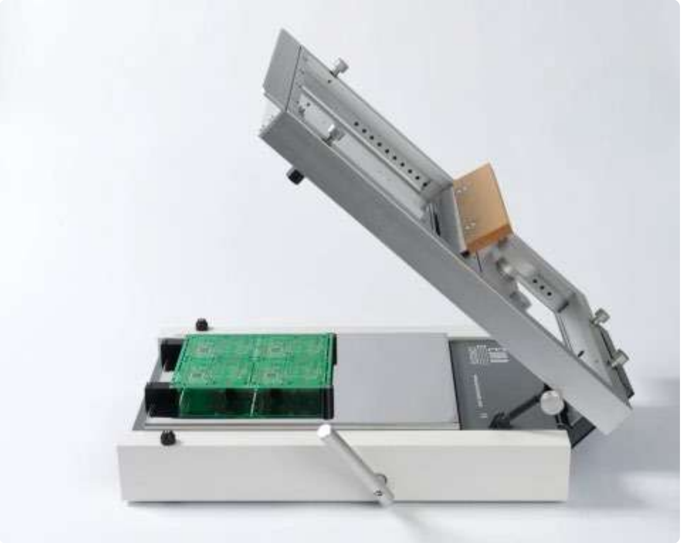


Figure 7: The eC-stencil-mate printer by Eurocircuits.

than the PCB pooling service (and that can often be further reduced by negotiating with the reseller). It's best to initiate conversations with three or more resellers before settling on one that's verified and has a better offer.

After settling with one of the resellers, components are to be sent to the PCB pooling service. Therefore, make sure the address of the PCB manufacturer is properly printed on the parcel. Ask the reseller to share a picture of the parcel showing the address and the tracking number. This ensures that everything's in order and that the shipment can be tracked. It's a good idea to share the tracking number and picture of the parcel with the pooling service's support team. This avoids any delays in order processing. A smooth assembly process is ensured by effective communication with both manufacturer and component reseller.

Five-Star Services

On the other hand, there exist companies that take your BoM and buy all the parts for you. Because they have many clients, they can buy in larger volumes than you, which lowers the price of the BoM. These companies can also order the PCB for you, removing another task from your list. They can also assemble the boards for you, and they can program and test the boards for you. Some will even stock and ship the boards for you. But, as usual, these services come at a cost. Even though super convenient, there is a catch: Many of these companies only accept orders starting at, say, one hundred or more boards, which may be just too many for what you had in mind. And, they may only accept professional customers, which means that you must start a company first.

DIY Assembly?

Assembling the circuit boards yourself can be the cheapest solution, if you do not count the hours you'll have to put in. Especially for SMD boards, all sorts of low-cost assembly tools have seen the light of day, from small pick-and-place machines, solder paste printers to reflow ovens and hot plates. For a few thousand euros, you are up and running. But, is it worth such an investment when you only have a hundred boards to do? Or, are you counting on follow-up orders? If not, just a soldering iron is the way to go. If you do, at least buy a solder fume extractor.

One of the options for having your SMD assembly setup is LumenPnP from Opulo (**Figure 6**). It's an open-source, easy-to-install, cost-effective pick-and-place machine that, most importantly, does not require too much space in your workspace. This makes it a good choice for makers and hobbyists. Other open-source options include OpenPnP and SimplePnP, etc.

Getting and using these PnP solutions is not the end of the story, but instead opens a whole new chapter. You will need a complete setup and process flow, including solder paste stencil handling (**Figure 7**), a PCB pre-heater (**Figure 8**), a reflow oven (**Figure 9**) and a board-testing setup. This sounds nice, but please note that these tools must be set up and configured for your board. You will

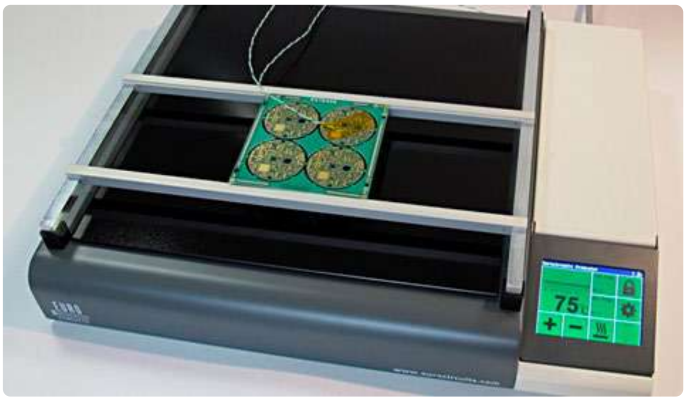


Figure 8: The eC-pre-heater and eC-fume-cube by Eurocircuits.

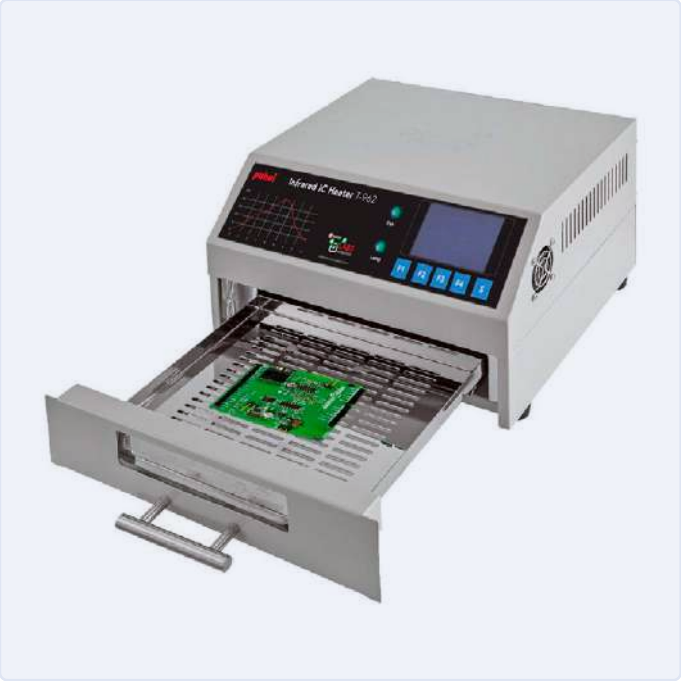


Figure 9: The T-962 Infrared Reflow Oven.

Figure 10: Fully assembled PCB: the Elektor ±40 V Linear Voltage Regulator Kit.

Comparison of Various PCB Services

Beta Layout: Stands out for its comprehensive PCB service portfolio. They provide options for PCB manufacturing, assembly, and even 3D printing. Beta Layout offers fast and reliable production, with a wide range of materials and finishes available. Their user-friendly website features an online Gerber viewer, making it easy to review and verify designs. Additionally, Beta Layout offers a Design Rule Check (DRC) tool to assist with design validation.
https://eu.beta-layout.com

Eurocircuits: Known for its high-quality PCB manufacturing and assembly services. They offer a wide range of PCB options, including prototype and production runs. It also provides a user-friendly online platform with instant pricing and a quick turnaround time. Their strong focus on customer support and extensive design rule checks ensure optimal results. Eurocircuits also offers a visualizer tool that allows users to review their PCB design before manufacturing.
https://eurocircuits.com

JLCPCB: Known for its competitive pricing and high-quality manufacturing. They offer a wide range of PCB options, including quick-turn prototypes and larger production runs. JLCPCB provides an easy-to-use online ordering platform and supports multiple file formats. Their extensive quality control processes, such as automated optical inspection (AOI), ensure reliable PCB fabrication.
https://jlcpcb.com

SH Park: A community-focused PCB service known for its distinctive, purple-colored boards. They specialize in prototype PCB manufacturing, providing high-quality results with attention to detail. OSH Park has a user-friendly website where users can upload their designs and review a visual rendering of the fabricated PCB. They prioritize transparency and offer shared projects for community feedback.
https://oshpark.com

PCBWay: A popular choice for low-volume PCB manufacturing due to its affordability and fast turnaround times. They offer a streamlined online ordering process, with real-time price calculation and instant quoting. PCBWay supports a variety of PCB options, including flexible and rigid-flex boards. Their website also provides an online Gerber viewer and a design rule check feature to ensure design accuracy.
https://pcbway.com

Summary and Recommendations:
Choosing the right PCB pooling service depends on specific project requirements, budget, and desired features. According to the comparison between the forementioned pooling services, PCBWay is the cheapest of the bunch if ordering 10 or fewer PCBs, but more than that, every other service costs almost the same. JLCPCB comes second in terms of pricing, typically, these services are also running a quite convenient offer. Consider factors such as pricing, manufacturing capabilities, turnaround time, quality control measures, and customer support. It's also beneficial to review user feedback and ratings for each service.

Tips for Better PCB Design

When aiming for cost-effective PCB designs, paying attention to specific technical details and industry standards can greatly optimize your approach. Here are some valuable tips and tricks to consider:

Track Width and Gap: Follow the IPC-2221 standard for guidelines [1] on track width and the desired current carrying capacity. For gap (clearance) between traces, pads, or components, maintain a minimum distance of 0.2 mm for low-voltage designs (<50 V) in a standard environment [2].

Manufacturing Tolerances: Ensure that your design adheres to manufacturing tolerances specified by the manufacturer. Confirm that dimensions, spacing, and hole sizes for components and vias are within acceptable limits to avoid fabrication issues.

Impedance Control: For high-frequency designs or applications requiring controlled impedance, calculate the trace width and separation based on desired characteristic impedance, substrate dielectric constant, and copper thickness. Use online calculators or specialized software tools to determine accurate trace dimensions.

Solder Mask and Silkscreen Alignment: Pay attention to proper alignment between the solder mask and silkscreen layers. Ensure the design accounts for the required alignment tolerance specified by the manufacturer to maintain visual appearance and manufacturability.

Panalization: Optimize low-volume production by panelizing multiple PCBs on a single panel. Follow manufacturer guidelines for panalization to ensure efficient manufacturing and cost reduction.

Component Footprints: Use standardized component footprints to ensure compatibility and reduce manufacturing costs. Verify that the footprints match the specific components you intend to use, referring to industry-standard libraries or component manufacturer recommendations for accuracy.

Testability: Design the PCB [3] with testability in mind to reduce testing and debugging costs. Incorporate test points, probe access areas, and built-in self-test (BIST) capabilities where applicable. This enables efficient testing during manufacturing, minimizing the need for costly rework or troubleshooting.

have to do test runs to make sure the configuration and settings are imported correctly from your PCB design. Then, add the components to the feeders and monitor the process to check that the machine is running properly, does not run out of components, and puts the components in the correct places. This can reduce the workload as compared to soldering manually, but it can also be tedious because it requires continuous monitoring if you would rather not be disappointed after returning from a coffee break. Overall, it's a nice setup to have, but if you don't want to go down this path, you always have the option of PCBA services available.

Regardless what option you choose, the sight of a fully assembled PCB (**Figure 10**) is always satisfying!

230477-01

Questions or Comments?

If you have technical questions or comments about this article, feel free to contact the Elektor editorial team at editor@elektor.com.

About the Author

Saad Imtiaz is an embedded systems engineer with a background in mechatronics engineering. He has amassed experience across diverse industry sectors, collaborating with international companies on IoT, embedded systems, automation, and product development, and having been involved in aviation and leadership of a technology startup. Saad joined Elektor in July 2023 as a Senior Engineer.

Related Products

- **Voltera V-One Desktop PCB Printer**
  www.elektor.com/18250
- **Upgraded T-962 Infrared Reflow Oven (Elektor Version)**
  www.elektor.com/20346
- **Digital Desoldering Station ZD-915**
  www.elektor.com/20142
- **Weller ESF 120ESD PCB Holder**
  www.elektor.com/17481

WEB LINKS

[1] Applying IPC-2221 Standards in Circuit Board Design [PDF]: https://tinyurl.com/ipc2221a
[2] PCB Land Pattern Design to the IPC-7351 Standard: https://tinyurl.com/ipc7351
[3] Practical PCB Layout Tips Every Designer Needs to Know: https://tinyurl.com/pcbtips

PREORDER **NOW!**

SPECTRAN® **V6**

BEYOND REALTIME

REALTIME SPECTRUM ANALYZER **5G**

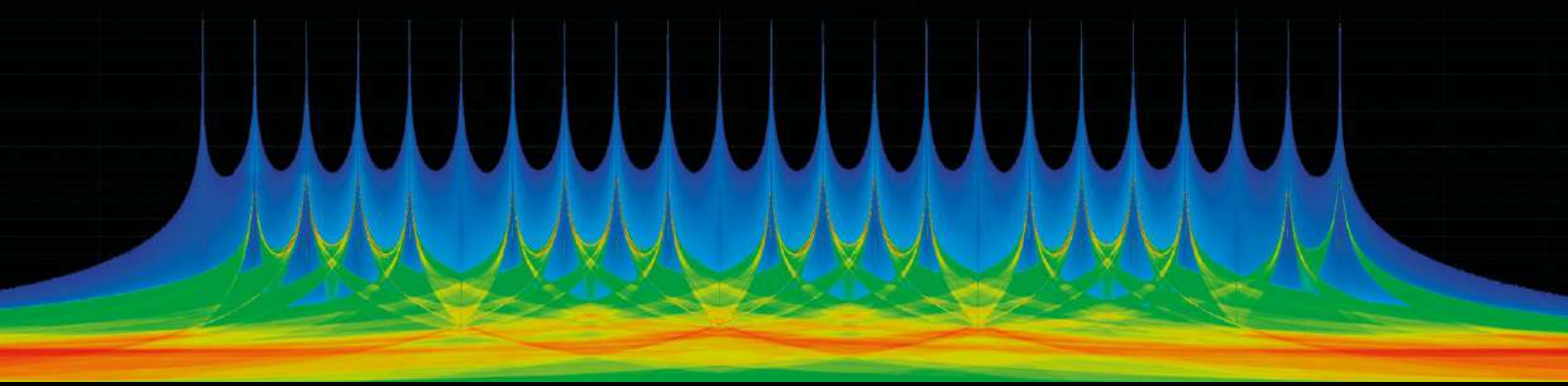
53 GHz USB Real-Time Spectrum Analyzer (FR1 & FR2)

250 MHz - 53 GHz | Dual receiver | **3 THz/s** sweep | **450 MHz** IBW | **16-Bit 2 GSPS** ADC | **IQ** streaming

-170 dBm/Hz DANL (4 dB NF) | Compact **USB** form factor | Incl. powerful **RTSA-Suite PRO** software

Image similar to final product

Every device includes the world's most advanced spectrum analyzer software **"RTSA-Suite PRO"**.



MADE IN GERMANY

www.aaronia.com
mail@aaronia.de
+49 6556 900 310

AARONIA AG®
WWW.AARONIA.DE

Q&A With Uri Shaked
(Developer on Design, Software, and More)

IoT Simulation Simplified with Wokwi

Developer Uri Shaked on Design, Software, and More

Questions by Roberto Armani (Elektor)

Want to simulate processors and boards? Check out Wokwi, an innovative open-source simulator for embedded systems and IoT devices. Its author, Uri Shaked, talks about the solution, as well as his background and interests.

Elektor: First, thanks for your time, Uri. Could you tell us about yourself, your background? Where are you currently located?

Uri Shaked: I'm a software engineer, based in Israel. As a toddler, I always dismantled my toys, being curious how they worked. In third grade, I broke my PC/XT machine. I was keen to learn how to fix it, so my aunt bought me a book about MS-DOS. The last two chapters were a brief introduction to QBasic. I was hooked. From that point, I kept programming every day.

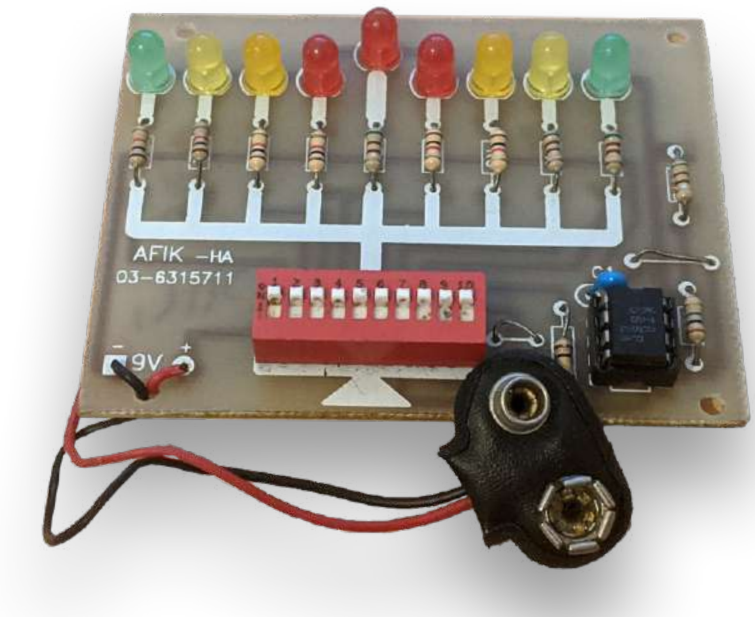


Figure 1: My Hanukkah Menorah project, which I assembled in fourth grade and is still functional!

Elektor: What sparked your interest in electronics? Was it a specific project, a particular course, a relative's passion for electronics, or perhaps a teacher?

Uri Shaked: In elementary school, we had a weekly electronics class. It was very hands-on: We learned how to solder and assembled circuit boards that the teacher gave us, while learning about the basics. I still remember how much I loved these classes, and the feeling of accomplishment when I held a finished project in my hand (**Figure 1**). Indeed, these labs provided the fundamental knowledge and experience and seeded me with the passion.

Elektor: No one forgets their "first love." Tell us about your first functional electronics project. What did you work on and why?

Uri Shaked: When I was in eighth grade, I spent a lot of time hanging out on IRC chat servers. One day, a friend shared a link explaining how to build a speaker system from old hard drives by connecting the actuator motor of the driver directly to an amplifier. I was fascinated by the idea, and decided to build my own. The sound quality wasn't perfect, but, with some trial and error, I found out that putting a heavy pot on top of the drive vastly improved the sound quality and volume (the sound resonated through the body of the pot). I ended up with a tower of pots in the living room, playing my mother's Pink Floyd vinyl through this setup. Needless to say, my mother wasn't pleased about this. Over the years, the hard drives kept coming as a theme in my projects, for example in my first MCU project, a hard-drive mounted PIC18 persistence of vision display or, as you may see in **Figure 2**, one of my first Arduino projects, a hard-drive-powered xylophone [1] that played songs on recycled beer bottles.

Elektor: Engineers and programmers learn a lot from their mistakes. Have any of your projects turned into disasters? Anything to share about that difficult project?

Uri Shaked: Definitely my GeekCon project. GeekCon is a weekend event, where a group of nerds joins together to build useless but cool things. The goal is to fail — as the motto goes, "if your project hasn't failed, you didn't aim high enough." In 2018, we tried to build a robot that plays the trumpet [2]. We used a water-filled latex glove to make up artificial lips, a pasta jar for an airflow regulator, and high-duty metal gear servo motors to control the artificial fingers (**Figure 3**). A minute before the final project presentation, there was a leak, and two of the servo motors just gave out. An epic failure. I don't like failing, so I went back home and spent the next month trying to fix the robot, and promised to bring it to Chrome's annual developer event, Chrome Dev Summit. I eventually figured out how to make a robust finger mechanism, but the lip mechanism was so tricky that I couldn't get it to work. I ended up "cheating," using a Raspberry Pi that played pre-recorded trumpet samples through the mouthpiece. But, at least the end result sounded decent, and my robot provided plenty of entertainment to the attendees of Chrome Dev Summit [3].

Elektor: Someone once said: "A clean desk is a sign of a sick mind." Would you be so brave to show us a picture of your workplace?

Uri Shaked: Figure 4 represents what my desk looked like when I tried to build a real-life version of the "Offline Dinosaur" game [4] from Google Chrome. This is what my desk looked like: My real problem was part storage. I had more than 4,600 items, kept in 40 different storage boxes, tracked in several Excel spreadsheets, consuming precious space in my small apartment. Last year, we moved, and I decided to give away almost everything to a local maker space.

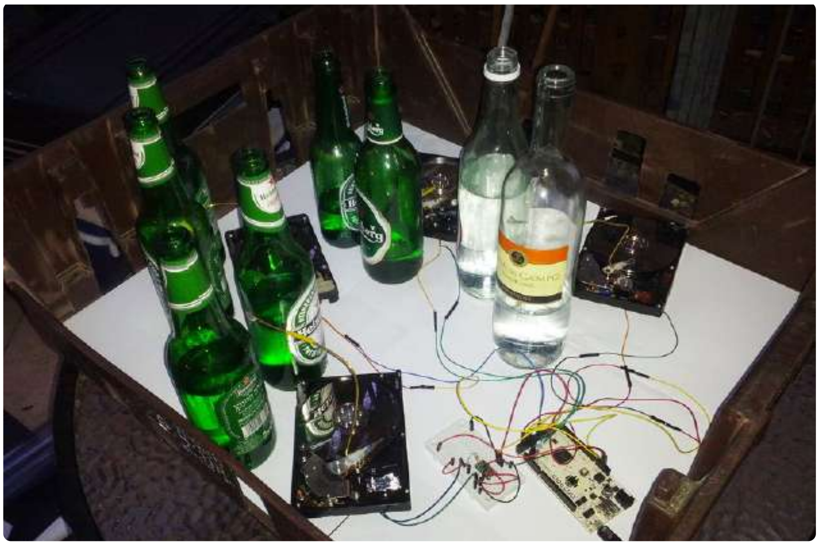


Figure 2: Hard-drive-powered xylophone.

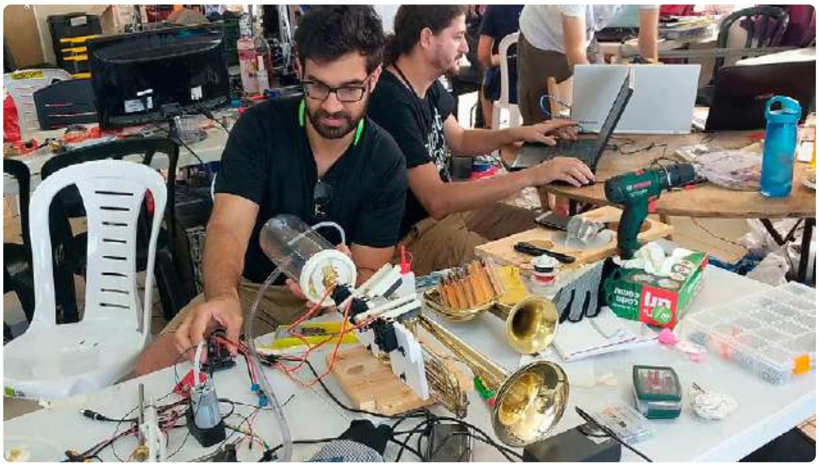


Figure 3: My friend Avi with our Trumpet Robot, just a few seconds before it blew up!

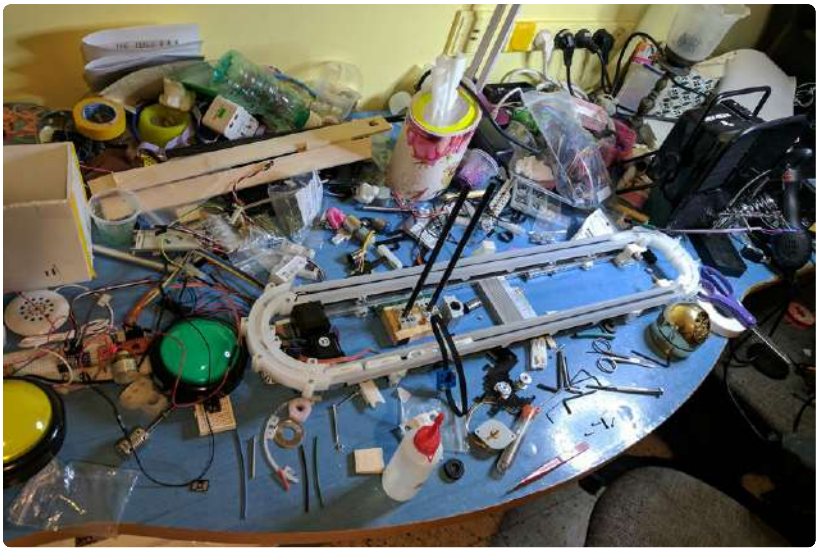
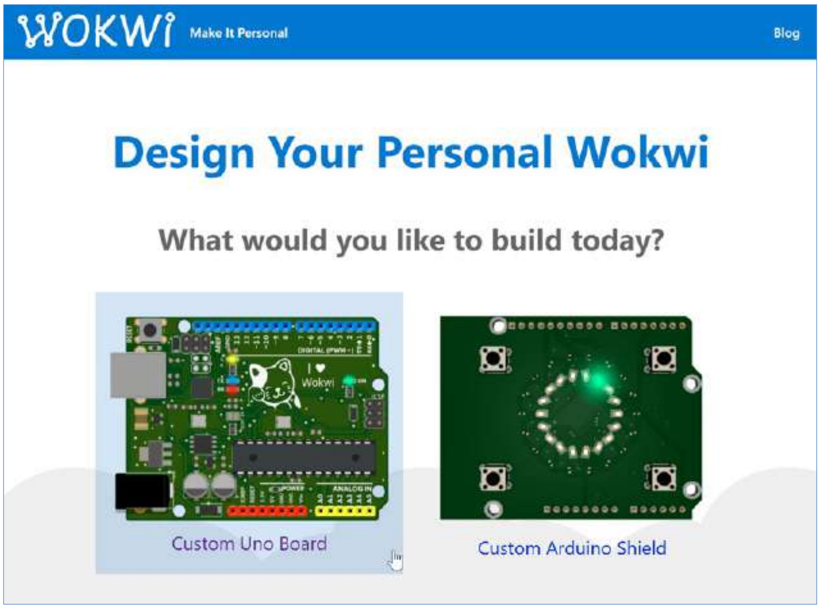


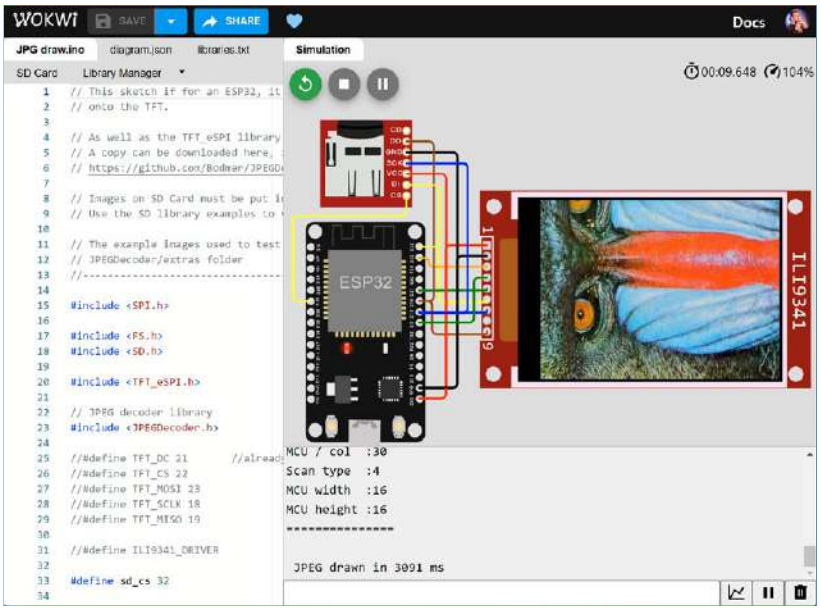
Figure 4: My tidy and neat workspace...



Design Your Personal Wokwi

What would you like to build today?

Figure 5: The old Wokwi homepage.



```
// This sketch if for an ESP32, it
// onto the TFT.

// As well as the TFT_eSPI library
// A copy can be downloaded here,
// https://github.com/Bodmer/JPEGD

// Images on SD Card must be put i
// Use the SD library examples to

// The example images used to test
// JPEGDecoder/extras folder
//--------------------------------

#include <SPI.h>

#include <FS.h>
#include <SD.h>

#include <TFT_eSPI.h>

// JPEG decoder library
#include <JPEGDecoder.h>

//#define TFT_DC 21       //alread
//#define TFT_CS 22
//#define TFT_MOSI 23
//#define TFT_SCLK 18
//#define TFT_MISO 19

//#define ILI9341_DRIVER

#define sd_cs 32

```

```
MCU / col  :30
Scan type  :4
MCU width  :16
MCU height :16
===============

 JPEG drawn in 3091 ms
```

Figure 6: Wokwi today.

Nowadays, I mostly use virtual parts. I still have a few Raspberry Pis, a Saleae Logic Analyzer, and a small collection of MCU boards (ESP32s, STM32s, Pi Picos, and Arduino), allowing me to compare physical chip behavior with the simulated one.

Elektor: If you had to suggest to beginners some "tips and tricks" on the setup of a proper workbench, what would you say?

Uri Shaked: Get a good power supply!

Elektor: Ok, on to Wowki. What is it? Where does the name come from? What led you to create it?

Uri Shaked: Wokwi is a simulator for embedded systems and IoT devices. There is a free online version, as well as a Visual Studio Code plugin. Wokwi can simulate ESP32, Raspberry Pi Pico, STM32, Arduino, and a bunch of other microcontrollers, as well as a wide variety of input and output devices: LCD screens, sensors, motors, LEDs, buttons, speakers, potentiometers, and more. As you can see in **Figure 5**, Wokwi started as a service that let you manufacture custom-designed Arduino boards with an easy drag-and-drop interface. Start from a board template, add peripherals (LEDs, sensors, etc), place them on the board, and we turn your design into a PCB, manufacture and assemble it, and ship it to you. I started blogging about Arduino to help spread the word, and I really wanted to include interactive, live demos in my articles. I couldn't find a good solution for easily sharing Arduino projects and simulating them on the web, so I started working on an open-source JavaScript library that could simulate the ATmega328P microcontroller. And, the rest is history. In **Figure 6**, you can see how it looks nowadays. When choosing the name for Wokwi, I was looking for a short, easy-to-pronounce word that didn't have any meaning yet.

Elektor: Who uses it? Is it completely free to use?

Uri Shaked: About 55% of the users are hobbyists, 35% are students and teachers (both universities and high schools), and 10% are professional firmware engineers. Some popular use cases are prototyping IoT systems (MQTT, Blynk, Thingsboard, IBM Cloud), online courses and workshops (e.g. the *Making Embedded Systems* course), and we're seeing growing interest from the Embedded Rust community. Wokwi has a free version and a paid version (*The Club*). The free version is very capable, and has the same simulation capabilities as the paid version. The paid version adds file and library uploads, local networking for the ESP32, and the ability to save private (unlisted) projects.

Elektor: Which parts or aspects of developing and leading the Wokwi project have proven to be the most challenging?

Uri Shaked: Wokwi aims to make complex systems simple and provide a great user experience. Instead of having to download and install compilers, operating system drivers, fight with permissions, faulty USB cables and flaky breadboard connections, you just click a button. Your code "magically" compiles, and you can see the result in your web browser. Creating this kind of "it just works" experience means we have to deal with many of the complexities ourselves. We set up the compiler toolchain in the cloud for multiple platforms (e.g. Rust, Arduino, esp-idf, verilog), ensure it runs fast at scale (we run about 2.3 million compilations per month), and hunt for mysterious bugs. For example, last month, I spent a day figuring out why using a specific library in an Arduino project caused the server to hang. It turned out there was an issue with the Arduino CLI [5] that caused it to hang when trying to install that library. Providing a great user experience also means helping users figure out why the code doesn't work the way they expect it to. I feel we still have a long way to go, but features such as the virtual Logic Analyzer, and the recently added Pin Function Dump function are definitely a step forward.

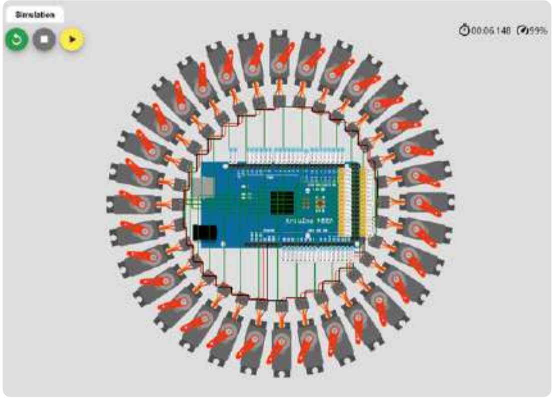
Elektor: Do you have a favorite "Featured Simulation Project?" Tell us about it.

Uri Shaked: Hard to choose, but if I had to go with just one, that would be *32 Servos Dancing* [6], also visible in **Figure 7** — an Arduino Mega controlling 32 servos and animating their arms. The project is more of an art project than a useful one. That's what I love about this project. Building it in real life would require a beefy power supply and spending a few hundred euro on parts, but, in the simulator, it's a no-brainer. Just click *Play* and see it in action. To spice things up, even the diagram for this project was generated by the Arduino code. Take a look at the `GenerateDiagram()` function, using trigonometric functions to place the servos. It prints a JSON file, which you can then paste into *diagram.json* in your Wokwi project, providing both the circuit and its visual representation. Now that's what I call innovative engineering! Speaking of user creativity: Wokwi provides a Custom Chips API [7], allowing you to code new simulation models for parts that do not yet exist in the simulation. The API provides a framebuffer interface, which allows you to create your own displays (LCD, E-Paper, etc.). What I didn't anticipate is that users will create their own oscilloscope using the framebuffer interface, as shown in **Figure 8**.

Elektor: ESP32 is quite a complex module on its own. Tell us about incorporating the ESP32 in your project. Was it difficult?

Uri Shaked: Wow, it was a tough one. The ESP32 uses the Xtensa instruction set, which contains over 250 instructions. The first step was to write code to decode and simulate each of those instructions, which ended up being about 4,500 lines of code. Then I created a GDB bridge (so I could debug the code running inside the simulator), and started working on the chip's peripherals. My initial goal was to get past the ROM and bootloader, and to run the *Hello World* application from ESP-IDF. After implementing UART, SPI, the timers (TIMG), and some other obscure peripherals (DPORT, EFUSE), I was finally able to get the first usable output from the serial console of the simulator. The biggest challenge, however, was simulating the Wi-Fi. Unlike other parts of the chip, the Wi-Fi peripheral is not documented at all, and the source code of the drivers is not available. It took weeks of intensive reverse-engineering [8] to figure out the registers that control the Wi-Fi radio and how to simulate them correctly. The hard work, paid off, though — ESP32 is becoming the most popular chip on Wokwi, and Espressif, the company behind the ESP32, is very supportive of the project. Their engineers are pushing forward the Rust support, using Wokwi in their training and workshops, and even spread the word about the simulator in professional conferences.

Elektor: Tell us about users' reaction to Wowki. What do you do with their feedback?



◄ Figure 7: 32 dancing servos, driven by an Arduino Mega board.

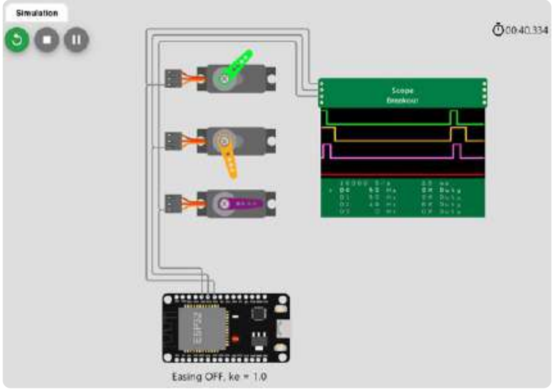
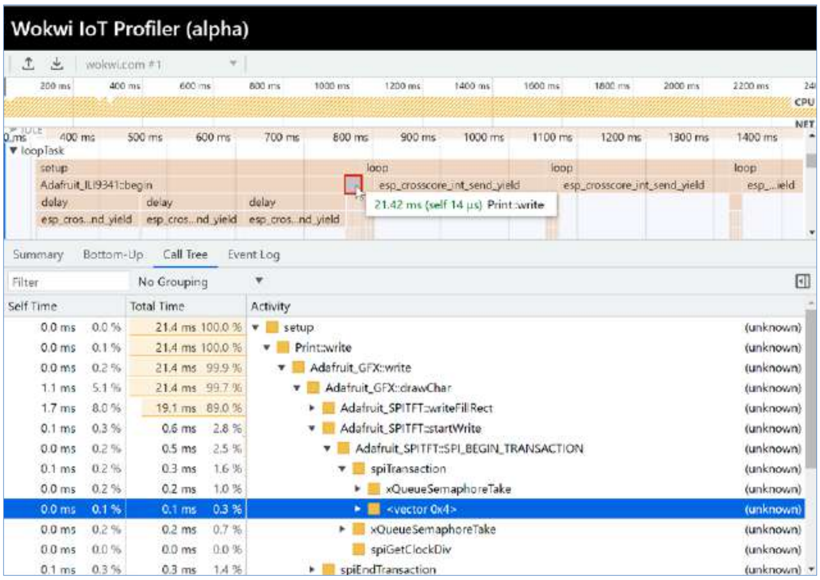


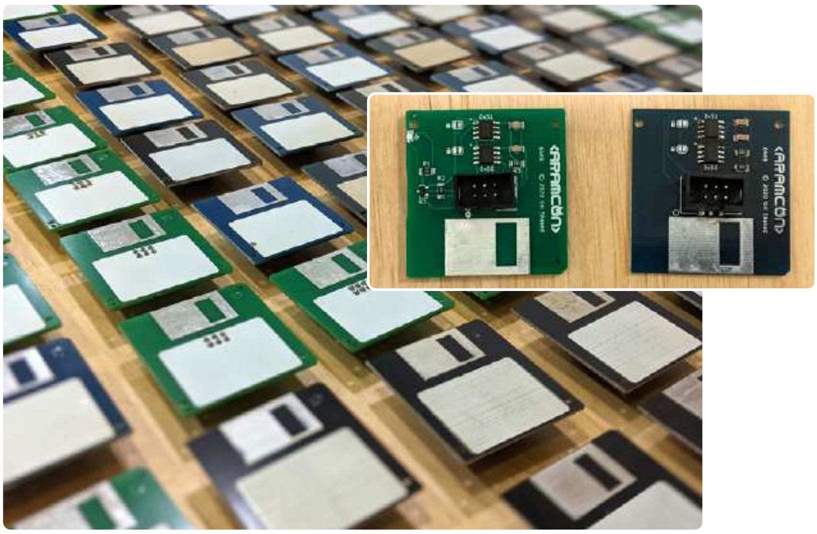
Figure 8: The Scope chip, showing servo signals in real time. ◄



21.42 ms (self 14 µs) Print::write

Self Time		Total Time		Activity	
0.0 ms	0.0 %	21.4 ms	100.0 %	setup	(unknown)
0.0 ms	0.1 %	21.4 ms	100.0 %	Print::write	(unknown)
0.0 ms	0.2 %	21.4 ms	99.9 %	Adafruit_GFX::write	(unknown)
1.1 ms	5.1 %	21.4 ms	99.7 %	Adafruit_GFX::drawChar	(unknown)
1.7 ms	8.0 %	19.1 ms	89.0 %	Adafruit_SPITFT::writeFillRect	(unknown)
0.1 ms	0.3 %	0.6 ms	2.8 %	Adafruit_SPITFT::startWrite	(unknown)
0.0 ms	0.2 %	0.5 ms	2.5 %	Adafruit_SPITFT::SPI_BEGIN_TRANSACTION	(unknown)
0.1 ms	0.2 %	0.3 ms	1.6 %	spiTransaction	(unknown)
0.0 ms	0.2 %	0.2 ms	1.0 %	xQueueSemaphoreTake	(unknown)
0.0 ms	0.1 %	0.1 ms	0.3 %	<vector 0x4>	(unknown)
0.0 ms	0.2 %	0.2 ms	0.7 %	xQueueSemaphoreTake	(unknown)
0.0 ms	0.0 %	0.0 ms	0.0 %	spiGetClockDiv	(unknown)
0.1 ms	0.3 %	0.3 ms	1.4 %	spiEndTransaction	(unknown)

Figure 9: Wokwi Embedded Profiler.



▲

Figure 10: The Floppy Disks, with a close-up of Electronics (two EEPROMs, write-lock solder jumpers, and an indication LED).

Uri Shaked: Our users are generally happy with the simulator, and it saves them a lot of time. But they also always ask for more and more features, way more than our small team can handle. To help focus on what's really important for users, I opened the "Vote for new features" page [9]. Everyone can suggest new features, and people can vote with their money for features that matter to them. It helps us prioritize, and it's also a great way for users to see what's coming next.

Elektor: Where would you like to see Wokwi in the next six to 12 months?

Uri Shaked: I'd like to see Wokwi getting more traction among professional users. Earlier this year, I released Wokwi for Visual Studio Code. It allows you to run the simulation directly in your IDE, and iterate much faster on your code. I'm already seeing Rust developers using it to enhance their productivity. Wokwi can also integrate with the debugger in VS Code, and, unlike real hardware, with Wokwi, you can set any number of breakpoints. I'm now working on a new embedded profiler — see **Figure 9**. Wokwi traces your code as it's running inside the simulator. You can see which functions are called, in what order, and how much time the microcontroller spends inside each function. I also plan to expand the ESP32 line (with ESP32-H2, ESP32-P4), add new STM32 devices, and perhaps get in touch with additional silicon vendors and add their microcontrollers as well. Finally, I would like to see Wokwi used in CI (continuous integration) environments. The idea is to test your embedded code on every commit. This is much easier to set up and scale with Wokwi, compared with maintaining a physical hardware setup. I'm already using a CI setup internally to ensure critical components, such as the ESP32 Wi-Fi, do not break with updates to the code and new ESP-IDF releases, and now working on a CLI (command-line interface) for Wokwi, to make it easy to use the simulator inside GitHub actions.

Elektor: You seem to be an artist as well as an engineer, as many can tell by reading your post, *A Practical Guide to Designing PCB Art* [10]. Tell us about your creative approach to art and electronics.

Uri Shaked: Sitting together with friends, planning a smart name badge for a conference. After discussing hardware add-ons for the badge, someone mentions it'd also be cool if the attendees could write small games for the badge, and have some kind of app store to share their creations. At that moment, an idea sparked in my mind. We don't need an online app store — instead, we can create a hardware add-on with a small EEPROM chip. The attendees will write their apps and games to the add-on, then give it away to their friends, so they could copy the code to their badge. Thus, the Floppy Disk add-on [11] was born. Once I had the idea, it was a matter of looking up the dimensions of a floppy disk, drawing a sketch with Inkscape, and deciding which PCB layer to use for each element: silk screen for the sticker, solder-plated copper (HASL) for the metal shutter, and copper covered with solder mask for the HD symbol. I did the schematic capture and layout in KiCad, and the result — well, you can see in **Figure 10**.

Elektor: Do you have any other important projects round the corner?

Uri Shaked: Tiny Tapeout [12]. I'm working on it together with Matt Venn, and our goal is to make custom silicon more affordable and accessible. You can create a digital design with Verilog or use Wokwi to create a design out of individual logic gates, then harden your design using OpenLane (an open-source digital design workflow), and submit your design for inclusion in our chip. We take your design, along with hundreds of other people's designs, and combine everything into a single-chip design. Then, we manufacture the chips, assemble them onto a PCB, and ship you a physical version of your design. The PCB that you see in **Figure 11** has a set of DIP switches for selecting a design, so you can also play around with the other designs in the chip. In fact, we encourage that — all the designs are open source, and we compile a datasheet with all the documentation. We're now starting to collaborate with universities to bring Tiny Tapeout to their students, and creating teaching materials to help hobbyists and silicon enthusiasts learn about semiconductors and digital design. Earlier this year, we released Siliwiz, an open-source app that integrates a visual layout editor, DRC engine, and SPICE simulation, all running inside the browser. There's a lot more to tell about Tiny Tapeout, the community, and the creative designs our users are putting together, but, that's a story for another time!

230366-01

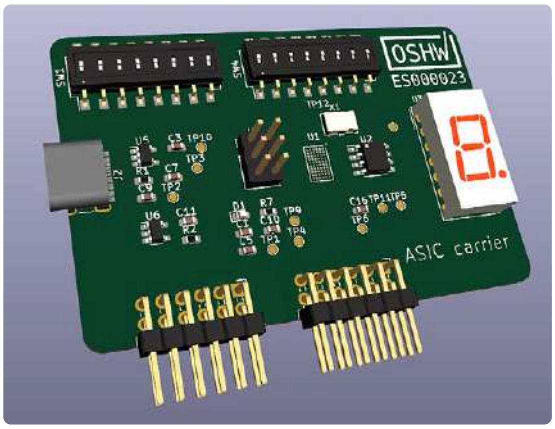


Figure 11: Tiny Tapeout breakout PCB (preliminary render).

About Uri Shaked

Uri Shaked is a longtime maker. He's currently working on Wokwi, an online IoT and embedded systems simulation platform, and on Tiny Tapeout, making custom ASIC manufacturing affordable and accessible. His projects and blog can be found at [13], with many tech talks and video interviews [14].

About the Author

Roberto Armani is an electronic engineer with over thirty-five years of experience in a variety of different sectors. Before joining the Elektor team as a Senior Editor, He gained experience and knowledge in the computer industry, electronic imaging, telecommunications, material testing equipment and web publishing. Besides electronics, he loves listening to (and singing) classical music, and taking high-altitude walks in the mountains.

WEB LINKS

[1] Hard Drive Xylophone Arduino Project: https://youtu.be/dw9U0WxtK9c
[2] Experiments with a robotized trumpet: https://bit.ly/3N2FfCz
[3] Playing Trumpet Demo at Chrome Dev Summit: https://youtu.be/PEVAczB9uUQ
[4] Real-Life version of the "Offline Dinosaur Game" from Google Chrome: https://bit.ly/3qgsifz
[5] CLI Issues with Arduino: https://github.com/arduino/arduino-cli/issues/2135
[6] Dancing Servos Project: https://wokwi.com/projects/305336312628511297
[7] Custom Chips API: https://docs.wokwi.com/chips-api/getting-started
[8] "Reverse-engineering the ESP32 Wi-Fi" video: https://youtu.be/XmaT8bMssyQ
[9] Wowki's Feature Voting Page: https://wokwi.com/features
[10] A Practical Guide to Designing PCB Art: https://blog.wokwi.com/a-practical-guide-to-designing-pcb-art/
[11] Floppy Disk Add-On: https://github.com/urish/floppy-disk-sao
[12] Tiny Tapeout: https://tinytapeout.com
[13] Uri's projects webpage: https://urish.org/#projects
[14] Uri's videos and interviews: https://urish.org/#talks

Join the
STM32 Wireless Innovation CONTEST

Innovate to win!
€5,000 in prizes!

The STM32 Wireless Innovation Design Contest 2023 gives you the opportunity to showcase your creativity and design skills by creating exciting wireless applications using the powerful development and evaluation boards from STMicroelectronics.

First Prize
€2,500

Second Prize
€1,500

Third Prize
€1,000

How to participate

1. Go to www.elektormagazine.com/st-contest.
2. Choose one of the featured boards.
3. Create an innovative project and document it.
4. Upload your project.

Submission Deadline: February 19, 2024

The Boards

NUCLEO-WBA52CG

NUCLEO-WBA52CG is a Bluetooth® Low Energy wireless and ultra-low-power board embedding a powerful and ultra-low-power radio compliant with the Bluetooth® Low Energy SIG specification v5.3. The ARDUINO® Uno V3 connectivity support and the ST morpho headers allow the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

Features

- Ultra-low-power wireless STM32WBA52CG microcontroller based on the Arm® Cortex®-M33 core, featuring 1 Mbyte of flash memory and 128 Kbytes of SRAM in a UFQFPN48 package
- MCU RF board (MB1863):
  - 2.4 GHz RF transceiver supporting Bluetooth® specification v5.3
  - Arm® Cortex® M33 CPU with TrustZone®, MPU, DSP, and FPU
  - Integrated PCB antenna
- Three user LEDs
- Three user and one reset push-buttons
- Board connectors:
  - USB Micro-B
  - ARDUINO® Uno V3 expansion connector
  - ST morpho headers for full access to all STM32 I/Os
- Flexible power-supply options: ST-LINK USB VBUS or external sources
- On-board STLINK-V3MODS debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port
- Comprehensive free software libraries and examples available with the STM32CubeWB MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE

STM32WB5MM-DK

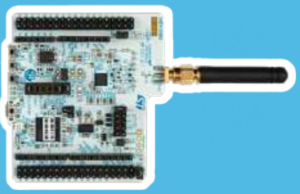
The STM32 device is a multi-protocol wireless and ultra-low-power device embedding a powerful and ultra-low-power radio compliant with the Bluetooth® Low Energy (BLE) SIG specification v5.2 and with IEEE 802.15.4-2011.

Features

- STM32WB5MMG (1-Mbyte Flash memory, 256-Kbyte SRAM, in Module RF package)
  - Dual-core 32-bit (Arm® Cortex®-M4 and dedicated M0+ CPU for real-time radio layer)
  - 2.4 GHz RF transceiver supporting Bluetooth® specification V5.2, 802.15.4 with Zigbee®, Thread®, and proprietary protocols
- 0.96-inch 128x64 OLED display
- 128-Mbit Quad-SPI NOR Flash Memory
- Temperature sensor
- Accelerometer/gyroscope sensor
- Time-of-Flight and gesture-detection sensor
- Digital microphone
- RGB LED
- Infrared LED
- 3 push-buttons (2 users and 1 reset) and 1 touch key button
- Board connectors:
  - STMod+
  - ARDUINO® Uno V3 expansion connector
  - USB user with Micro-B connector
  - TAG10 10-pin footprint
- Flexible power-supply options: ST-LINK/V2-1 USB connector, 5 V delivered by ARDUINO® or external connector, USB charger, or USB power
- On-board ST-LINK/V2-1 debugger/programmer with USB re-enumeration capability: Virtual COM port and debug port
- Comprehensive free software libraries and examples available with the STM32CubeWB MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE

NUCLEO-WL55JC

The NUCLEO-WL55JC STM32WL Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes with the STM32WL Series microcontroller, choosing from the various combinations of performance, power consumption, and features.



Features

- STM32WL55JC microcontroller multi-protocol LPWAN dual-core 32-bit (Arm® Cortex®-M4/M0+ at 48 MHz) in UFBGA73 package featuring:
  - Ultra-low-power MCU
  - RF transceiver (150 MHz to 960 MHz frequency range) supporting LoRa®, (G)FSK, (G)MSK, and BPSK modulations
  - 256-Kbyte Flash memory and 64-Kbyte SRAM
- 3 user LEDs
- 3 user buttons and 1 reset push-button
- 32.768 kHz LSE crystal oscillator
- 32 MHz HSE on-board oscillator
- Board connectors:
  - USB with Micro-B
  - MIPI® debug connector
  - ARDUINO® Uno V3 expansion connector
  - ST morpho extension pin headers for full access to all STM32WL I/Os
- Delivered with SMA antenna
- Flexible power-supply options: ST-LINK, USB VBUS, or external sources
- On-board STLINK-V3 debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port
- Comprehensive free software libraries and examples available with the STM32CubeWL MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE
- Suitable for rapid prototyping of end nodes based on LoRaWAN®, Sigfox™, wM-Bus, and many other proprietary protocols
- Fully open hardware platform

www.elektormagazine.com/st-contest

A Bare-Metal Programming Guide (Part 3)

CMSIS Headers, Automatic Testing, and a Web Server

By Sergey Lyubka (Ireland)

In the first two parts of that guide, we learned how to get access to controller pins, the system clock and the UART, and build our first firmware examples with linker scripts and Makefiles. In this last installment of the series, predefined headers and libraries will make our life much easier. We will program a web server and see how builds and tests of such more complex firmware can be automated.

Editor's note: This guide is a living document on GitHub [1] and it grows.

Vendor CMSIS Headers

In the previous articles [2][3], we have developed the firmware using only datasheets, editor, and GCC compiler. We have created peripheral structure definitions manually, using datasheets.

Now, as you know how it all works, it's time to introduce CMSIS headers. These are header files with all definitions, created and supplied by the MCU vendor. They contain definitions for the internal blocks and peripherals of that MCU, and are thus rather large.

CMSIS stands for Common Microcontroller Software Interface Standard, and is a common ground for MCU manufacturers to specify peripheral APIs. Since CMSIS is an ARM standard, and since CMSIS headers are supplied by the MCU vendor, they are the source of authority. Therefore, using vendor headers is preferable to writing definitions manually.

There are two sets of CMSIS headers:

- ARM Core CMSIS headers. They describe the ARM core, and are published by ARM on GitHub [4]
- MCU vendor CMSIS headers. They describe MCU peripherals, and are published by the MCU vendor. In our case, ST publishes them at [5]

We can pull those headers by a simple Makefile snippet:

```
cmsis_core:
  git clone --depth 1 -b 5.9.0
   https://github.com/ARM-software/CMSIS_5 $@
cmsis_f4:
  git clone --depth 1 -b v2.6.8
   https://github.com/STMicroelectronics/
   cmsis_device_f4 $@
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-5-cmsis/Makefile

The ST CMSIS package also provides startup files for all of their MCUs. We can use those instead of hand-writing the *startup.c*. The ST-provided startup file calls the `SystemInit()` function, so we define it in *main.c*.

Now, let's replace our API functions in *hal.h* using CMSIS definitions, and leave the rest of the firmware intact. From *hal.h*, remove all peripheral API and definitions, and leave only standard C includes, vendor CMSIS include, defines to PIN, BIT, FREQ, and the `timer_expired()` helper function.

If we try to rebuild the firmware — `make clean build`, then GCC will fail, complaining about missing `systick_init()`, `GPIO_MODE_OUTPUT`, `uart_init()`, and `UART3`. Let's add those, using STM32 CMSIS files.

Let's start from `systick_init()`. ARM core CMSIS headers provide a `SysTick_Config()` function that does the same, so we'll use it.

Next comes the `gpio_set_mode()` function. The *stm32f429xx.h* header has a `GPIO_TypeDef` structure, identical to our `struct gpio`. Let's use it:

```
#define GPIO(bank) ((GPIO_TypeDef *)
  (GPIOA_BASE + 0x400U * (bank)))
enum { GPIO_MODE_INPUT, GPIO_MODE_OUTPUT,
   GPIO_MODE_AF, GPIO_MODE_ANALOG };

static inline void gpio_set_mode
      (uint16_t pin, uint8_t mode) {
  GPIO_TypeDef *gpio =
  GPIO(PINBANK(pin));  // GPIO bank
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-5-cmsis/hal.h

The `gpio_set_af()` and `gpio_write()` functions are also trivial — simply replace `struct gpio` with `GPIO_TypeDef`, and that's it.

Next goes UART. There is a `USART_TypeDef`, and defines for USART1, USART2, USART3. Let's use them:

```
#define UART1 USART1
#define UART2 USART2
#define UART3 USART3
```

In the `uart_init()` and the rest of UART functions, change `struct uart` to `USART_TypeDef`. The rest stays the same!

And, we're done. Rebuild and reflash the firmware. The LED blinks, the UART shows the output. Congratulations, we have adapted our firmware code to use vendor CMSIS header files. Now ,let's reorganize the repository a bit by moving all standard files into the *include* directory and updating Makefile to let GCC know about it:

```
-I. -Iinclude -Icmsis_core/CMSIS/Core/Include -Icmsis_f4/Include \
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-5-cmsis/Makefile

Also, let's include CMSIS header pulling as a dependency for the binary:

```
firmware.elf: cmsis_core cmsis_f4 mcu.h
      link.ld Makefile $(SOURCES)
```

We're left with a project template that can be reused for the future projects. You can find the complete project source code in the *step-5-cmsis* project folder [6].

Setting Up Clocks

After boot, the Nucleo-F429ZI CPU runs at 16 MHz. The maximum frequency is 180 MHz. Note that system clock frequency is not the only factor we need to care about. Peripherals are attached to different buses, APB1 and APB2, which are clocked differently. Their clock speeds are configured by the frequency prescaler values, set in the RCC (the RCC controller manages system and peripheral clocks). The main CPU clock source can also be different — we can use either an external crystal oscillator (HSE) or an internal oscillator (HSI). In our case, we'll use HSI.

When the CPU executes instructions from flash, the flash read speed (which is around 25 MHz) becomes a bottleneck if the CPU clock gets higher. There are several tricks that can help. Instruction prefetch is one. Also, we can give a clue to the flash controller as to how fast the system clock is: That value is called flash latency. For a 180 MHz system clock, the `FLASH_LATENCY` value is 5. Bits 8 and 9 in the flash controller enable instruction and data caches:

```
FLASH->ACR |= FLASH_LATENCY | BIT(8) |
    BIT(9); // Flash latency, caches
```

The clock source (HSI or HSE) goes through a piece of hardware called the PLL, which multiplies the source frequency by a certain value. Then, a set of frequency dividers are used to set the system clock and the APB1, APB2 clocks. In order to obtain the maximum system clock of 180 MHz, multiple values of PLL dividers and APB prescalers are possible. Section 6.3.3 of the STM32F4xx controller reference manual [7] tells us the maximum values for APB1 clock: ≤ 45 MHz, and the APB2 clock: ≤ 90 MHz. That narrows down the list of possible combinations. Here, we chose the values manually. Note that tools like CubeMX can automate the process and make it easy and visual.

```
// 6.3.3: APB1 clock <= 45MHz;
//      APB2 clock <= 90MHz
// 3.5.1, Table 11: configure flash
// latency (WS) in accordance to clock freq
// 33.4: The AHB clock must be at least
// 25 MHz when Ethernet is used
enum { APB1_PRE = 5 /* AHB clock / 4 */,
  APB2_PRE = 4 /* AHB clock / 2 */ };
enum { PLL_HSI = 16, PLL_M = 8,
  PLL_N = 180, PLL_P = 2 };
     // Run at 180 Mhz
#define FLASH_LATENCY 5
#define SYS_FREQUENCY ((PLL_HSI * PLL_N /
      PLL_M / PLL_P) * 1000000)
#define APB2_FREQUENCY
   (SYS_FREQUENCY / (BIT(APB2_PRE - 3)))
#define APB1_FREQUENCY
   (SYS_FREQUENCY / (BIT(APB1_PRE - 3)))
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-6-clock/hal.h

Now, we're ready for a simple algorithm to set the clock up for CPU and peripheral buses. It may look like this:

- Optionally, enable FPU
- Set flash latency
- Decide on a clock source, and PLL, APB1 and APB2 prescalers
- Configure RCC by setting respective values

Listing 1. Clock Initialization.

[see https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-6-clock/sysinit.c]

```
uint32_t SystemCoreClock = SYS_FREQUENCY;

 void SystemInit(void) {  // Called automatically by startup code
   SCB->CPACR |= ((3UL << 10 * 2) | (3UL << 11 * 2));  // Enable FPU
   FLASH->ACR |= FLASH_LATENCY | BIT(8) | BIT(9);      // Flash latency, prefetch
   RCC->PLLCFGR &= ~((BIT(17) - 1));                   // Clear PLL multipliers
   RCC->PLLCFGR |= (((PLL_P - 2) / 2) & 3) << 16;      // Set PLL_P
   RCC->PLLCFGR |= PLL_M | (PLL_N << 6);               // Set PLL_M and PLL_N
   RCC->CR  |= BIT(24);                                // Enable PLL
   while ((RCC->CR & BIT(25)) == 0) spin(1);           // Wait until done
   RCC->CFGR = (APB1_PRE << 10) | (APB2_PRE << 13);    // Set prescalers
   RCC->CFGR  |= 2;                                    // Set clock source to PLL
   while ((RCC->CFGR & 12) == 0) spin(1);              // Wait until done

   RCC->APB2ENR |= RCC_APB2ENR_SYSCFGEN;    // Enable SYSCFG
   SysTick_Config(SystemCoreClock / 1000);  // Sys tick every 1ms
 }
```

- Move clock initialization into a separate file *sysinit.c* function `SystemInit()`, which is automatically called by the startup code

See **Listing 1**.

We need to also change *hal.h* — specifically, the UART initialization code. Different UART controllers are running on different buses: UART1 runs on a fast APB2, and the rest of the UARTs run on a slower APB1. When running on a default 16 MHz clock, that does not make a difference. But, when running at higher speeds, APB1 and APB2 may have different clocks, thus we need to adjust the baud rate calculation for the UART. See **Listing 2**.

Rebuild and reflash, and our board runs at its maximum speed, 180 MHz! A complete project source code can be found in the *step-6-clock* project folder [8].

Web Server With Device Dashboard

The Nucleo-F429ZI comes with Ethernet onboard. Ethernet hardware needs two components: a PHY (which transmits/receives electrical signals to the media, such as copper, optical cable, etc.) and MAC (which drives the PHY controller). On our Nucleo Board, the MAC controller is built-in, and the PHY is external (specifically, it is Microchip's LAN8720a).

MAC and PHY can talk several interfaces. We'll use RMII. For that, a bunch of pins must be configured to use their Alternative Function (AF). To implement a web server, we need three software components:

- a network driver, which sends/receives Ethernet frames to/from MAC controller
- a network stack, which parses frames and understands TCP/IP
- a network library that understands HTTP

We will use the *Mongoose* Network Library [9] which implements all of that in a single file. It's a dual-licensed library (GPLv2/commercial) that was designed to make network embedded development fast and easy.

So, copy *mongoose.c* [10] and *mongoose.h* [11] to our project. Now we have a driver, a network stack, and a library at hand. Mongoose also provides a large set of examples, and one of them is a device dashboard example [12]. It implements lots of things, such as dashboard login, real-time data exchange over WebSocket, an embedded file system, MQTT communication, etc. So, let's use that example. Copy two extra files:

- *net.c* [13] — implements dashboard functionality
- *packed_fs.c* [14] — contains HTML/CSS/JS GUI files

What we need is to tell Mongoose which functionality to enable. That can be done via compilation flags, by setting preprocessor constants. Alternatively, the same constants can be set in the *mongoose_custom.h* file. Let's go the second way. Create a *mongoose_custom.h* file with the following contents:

```
#pragma once
#define MG_ARCH MG_ARCH_NEWLIB
#define MG_ENABLE_MIP 1
#define MG_ENABLE_PACKED_FS 1
#define MG_IO_SIZE 512
#define MG_ENABLE_CUSTOM_MILLIS 1
```

Now it's time to add some networking code to *main.c*. We `#include "mongoose.c"`, initialize Ethernet RMII pins and enable Ethernet in the RCC. See **Listing 3**.

Listing 2. UART Initialization.

[see https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-6-clock/hal.h]

```
static inline bool uart_init(USART_TypeDef *uart, unsigned long baud) {

  // https://www.st.com/resource/en/datasheet/stm32f429zi.pdf
  uint8_t af = 7;           // Alternate function
  uint16_t rx = 0, tx = 0;  // pins
  uint32_t freq = 0;        // Bus frequency. UART1 is on APB2, rest on APB1

  if (uart == USART1) {
    freq = APB2_FREQUENCY, RCC->APB2ENR |= BIT(4);
    tx = PIN(’A’, 9), rx = PIN(’A’, 10);
  } else if (uart == USART2) {
    freq = APB1_FREQUENCY, RCC->APB1ENR |= BIT(17);
    tx = PIN(’A’, 2), rx = PIN(’A’, 3);
  } else if (uart == USART3) {
    freq = APB1_FREQUENCY, RCC->APB1ENR |= BIT(18);
    tx = PIN(’D’, 8), rx = PIN(’D’, 9);
  } else {
    return false;
  }
```

Listing 3. Initialize Ethernet, enable MAC GPIO pins.

[see https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-7-webserver/nucleo-f429zi/main.c]

```
uint16_t pins[] = ;
   for (size_t i = 0; i < sizeof(pins) / sizeof(pins[0]); i++) {
     gpio_init(pins[i], GPIO_MODE_AF, GPIO_OTYPE_PUSH_PULL, GPIO_SPEED_INSANE,
               GPIO_PULL_NONE, 11);
   }
   nvic_enable_irq(61);                         // Setup Ethernet IRQ handler
   RCC->APB2ENR |= BIT(14);                     // Enable SYSCFG
   SYSCFG->PMC |= BIT(23);                      // Use RMII. Goes first!
   RCC->AHB1ENR |= BIT(25) | BIT(26) | BIT(27);  // Enable Ethernet clocks
   RCC->AHB1RSTR |= BIT(25);                    // ETHMAC force reset
   RCC->AHB1RSTR &= ~BIT(25);                   // ETHMAC release reset
```

Mongoose’s driver uses Ethernet interrupt, thus we need to update *startup.c* and add *ETH_IRQHandler* to the vector table. Let’s reorganize the vector table definition in *startup.c* in a way that does not require any modification to add an interrupt handler function. The idea is to use a “weak symbol” concept.

A function can be marked “weak” and it works like a normal function. The difference comes when source code defines a function with the same name elsewhere. Normally, two functions with the same name make a build fail. However, if one function is marked weak, then a build succeeds and the linker selects a non-weak function. This gives the ability to provide a “default” function in a boilerplate, with the ability to override it by simply creating a function with the same name elsewhere in the code.

Here how it works in our case. We want to fill a vector table with default handlers, but give the user the ability to override any handler. For that, we create a `DefaultIRQHandler()` function and mark it as `weak`. Then, for every IRQ handler, we declare a handler name and make it an alias to `DefaultIRQHandler()`:

```
void __attribute__((weak)) DefaultIRQHandler(void) {
for (;;) (void) 0;
}
#define WEAK_ALIAS
  __attribute__((weak, alias(“DefaultIRQHandler”)))
WEAK_ALIAS void NMI_Handler(void);
WEAK_ALIAS void HardFault_Handler(void);
WEAK_ALIAS void MemManage_Handler(void);
```

Listing 4. Initialize Mongoose library.

```
struct mg_mgr mgr;        // Initialise Mongoose event manager
mg_mgr_init(&mgr);        // and attach it to the MIP interface
mg_log_set(MG_LL_DEBUG);  // Set log level
struct mip_driver_stm32 driver_data = {.mdc_cr = 4};  // See driver_stm32.h
struct mip_if mif = {
     .mac {2, 0, 1, 2, 3, 5}
     .use_dhcp = true,
     .driver = &mip_driver_stm32,
     .driver_data = &driver_data,
};
mip_init(&mgr, &mif);
extern void device_dashboard_fn(struct mg_connection *, int, void *, void *);
mg_http_listen(&mgr, "http://0.0.0.0", device_dashboard_fn, &mgr);
MG_INFO(("Init done, starting main loop"));
```

Listing 5. Makefile with libraries references.

```
847 3 mongoose.c:6784:arp_cache_add     ARP cache: added 0xc0a80001 @
90:5c:44:55:19:8b
 84e 2 mongoose.c:6817:onstatechange     READY, IP: 192.168.0.24
 854 2 mongoose.c:6818:onstatechange            GW: 192.168.0.1
 859 2 mongoose.c:6819:onstatechange            Lease: 86363 sec
 LED: 1, tick: 2262
 LED: 0, tick: 2512
```

```
...
__attribute__((section(“.vectors”)))
void (*tab[16 + 91])(void) =
{ 0, _reset, NMI_Handler,
HardFault_Handler, MemManage_Handler,
...
```

Now, we can define any IRQ handler in our code, and it will replace the default one. This is what happens in our case: There is an `ETH_IRQHandler()` function defined by Mongoose's STM32 driver, which replaces a default handler.

The next step is to initialize the Mongoose library: Create an event manager, setup the network driver, and start a listening HTTP connection. See **Listing 4**.

What's left is to add a `mg_mgr_poll()` call into the main loop.

Now, add *mongoose.c*, *net.c*, and *packed_fs.c* files to the Makefile. Rebuild and reflash the board. Attach a serial console to the debug output and observe that the board obtains an IP address over DHCP. See **Listing 5**.

Fire up a browser at that IP address, and get a working dashboard, with real-time graph over WebSocket, MQTT, authentication, and other things! See the full description for more details.

Complete project source code can be found in the *step-7-webserver* directory [15].

Automated Firmware Builds (Software CI)

It's good practice for a software project to have a continuous integration (CI) test. On every change pushed to the repository, CI automatically rebuilds and tests all components.

GitHub makes it easy to do. We can create a *.github/workflows/test.yml* file, which is a *CI configuration* file. In that file, we can install ARM GCC and run `make` in every example directory to build respective firmware files.

Long story short, this tells GitHub to run on every repo push:

```
name: build
on: [push, pull_request]
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/.github/workflows/test.yml

This installs the ARM GCC compiler:

```
- run: sudo apt -y install
  gcc-arm-none-eabi make stlink-tools
```

This builds the firmware in every example directory:

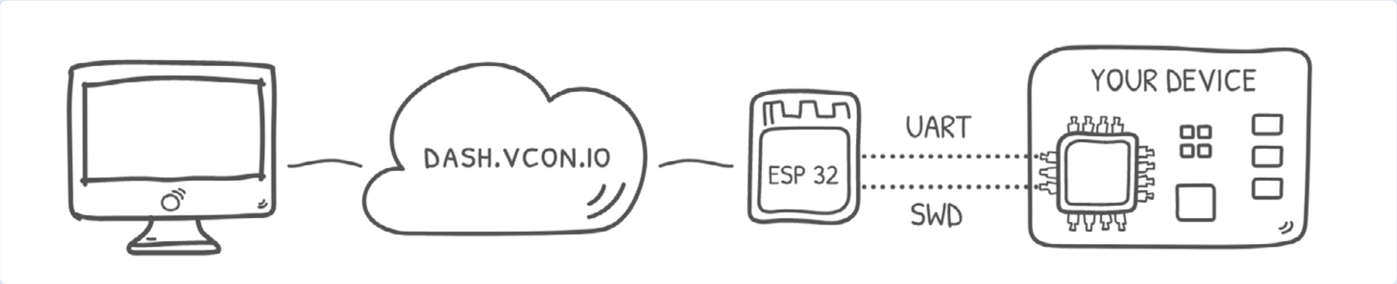


Figure 1: Setting up ESP32 as a remotely-controlled programmer and registering on vcon.io.

```
- run: make -C step-0-minimal
- run: make -C step-1-blinky
- run: make -C step-2-systick
- run: make -C step-3-uart
- run: make -C step-4-printf
- run: make -C step-5-cmsis
- run: make -C step-6-clock
- run: make -C step-7-webserver/nucleo-f429zi
- run: make -C step-7-webserver/pico-w
```

That's it! Extremely simple and extremely powerful. Now, if we push a change to the repo that breaks a build, GitHub will notify us. On success, GitHub will keep quiet. See an example successful run [16].

Automated Firmware Tests (Hardware CI)

Would it be great to also test built firmware binaries on a real hardware, to test not only the build process, but that the built firmware is correct and functional?

It is not trivial to build such a system ad hoc. For example, one can set up a dedicated test workstation, attach a tested device (e.g. Nucleo-F429ZI board) to it, and write a piece of software for remote firmware upload, and test using a built-in debugger. Possible, but fragile, takes a lot of effort and needs a lot of attention.

The alternative is to use one of the commercial hardware test systems embedded board farms (EBFs), though such commercial solutions are quite expensive. But, there is an easier way.

Solution: ESP32 + vcon.io

Using the *http://vcon.io* service, which implements remote firmware update and UART monitor, we can:

- Take any ESP32 or ESP32-C3 device (e.g. any inexpensive development board)
- Flash a pre-built firmware on it, turning ESP32 into a remotely-controlled programmer
- Wire ESP32 to the target device: SWD pins for flashing, UART pins for capturing output
- Configure ESP32 to register on the *vcon.io* [17] management dashboard

When done, your target device will have an authenticated, secure RESTful API for reflashing and capturing device output. It can be called from anywhere, for example from the software CI (see **Figure 1**).

Note: the vcon.io service is run by Cesanta — the company I work for. It is a paid service with a freebie quota: If you have just a few devices to manage, it is completely free.

Configuring and Wiring ESP32

Take any ESP32 or ESP32-C3 device — a devboard, a module, or your custom device. My recommendation is ESP32-C3 XIAO devboard because of its low price and small form factor.

We're going to assume that the target device is a Raspberry Pi W5500-EVB-Pico board [18] with a built-in Ethernet interface. If your device is different, adjust the *Wiring* step according to its pinout.

- Follow Flashing ESP32 [19] to flash your ESP32
- Follow Network Setup [20] to register ESP32 on dashboard
- Follow Wiring [21] to wire ESP32 to your device

Figure 2 shows how a configured device breadboard setup may look. **Figure 3** shows how a configured device dashboard looks.

Now, you can reflash your device with a single command:

```
curl -su :API_KEY https://dash.vcon.io/api/v3/devices/
ID/ota --data-binary @firmware.bin
```

Where `API_KEY` is the dash.vcon.io authentication key, `ID` is the registered device number, and `firmware.bin` is the name of the newly built firmware. You can get the `API_KEY` by clicking on the *api key* link on a dashboard. The device ID is listed in the table.

We can also capture device output with a single command:

```
curl -su :API_KEY https://dash.vcon.io/api/v3/devices/
ID/tx?t=5
```

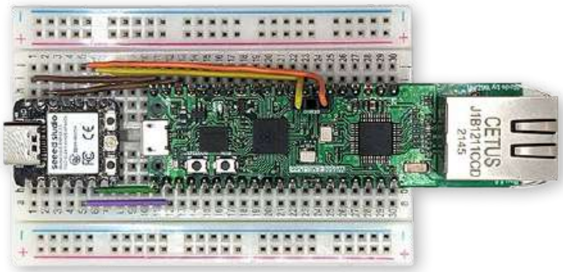
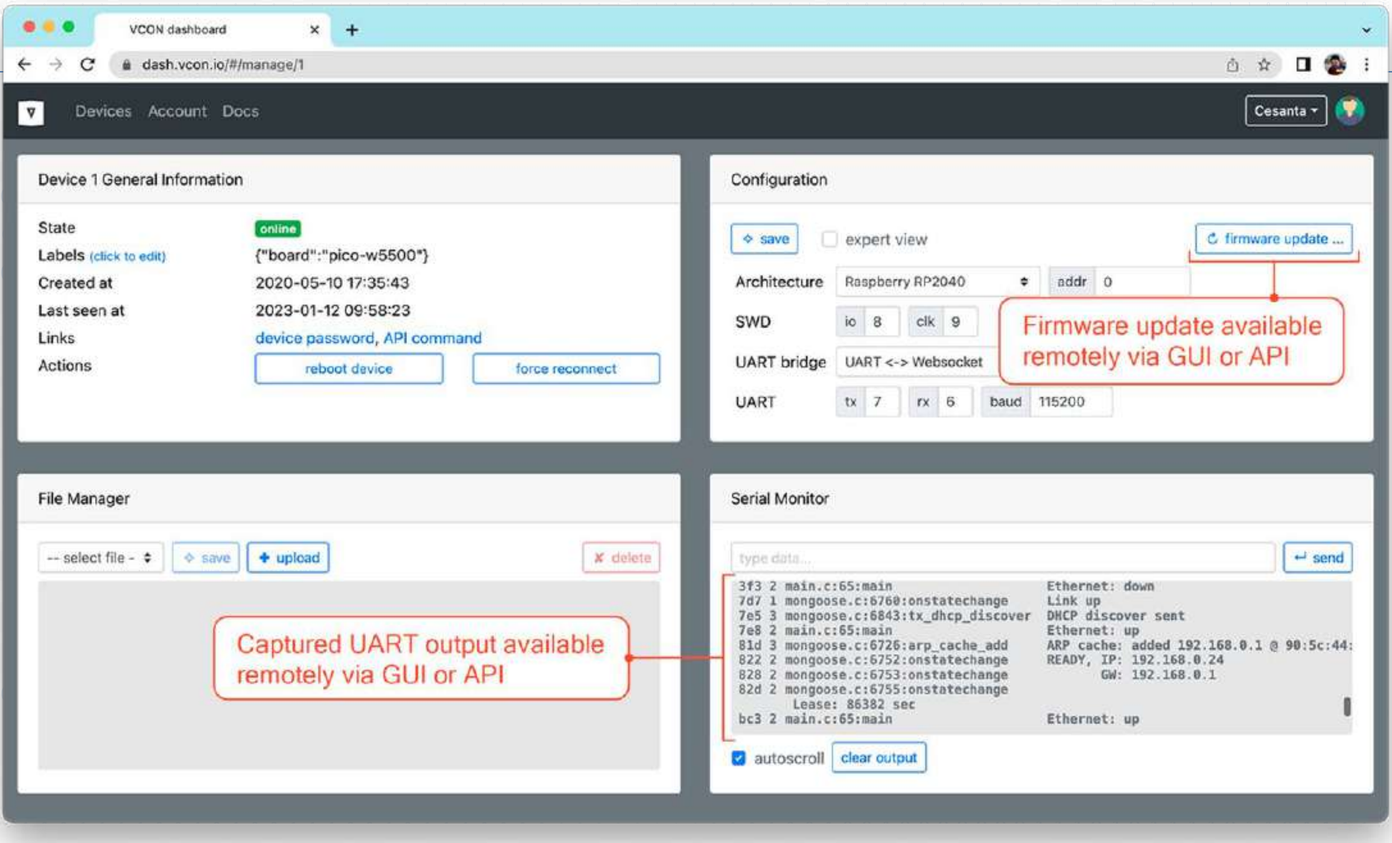


Figure 2: Configured device breadboard setup.



State: online
Labels (click to edit): {"board":"pico-w5500"}
Created at: 2020-05-10 17:35:43
Last seen at: 2023-01-12 09:58:23
Links: device password, API command
Actions: reboot device, force reconnect

Architecture: Raspberry RP2040, addr 0
SWD: io 8, clk 9
UART bridge: UART <-> Websocket
UART: tx 7, rx 6, baud 115200

☐ expert view

```
3f3 2 main.c:65:main                   Ethernet: down
7d7 1 mongoose.c:6760:onstatechange    Link up
7e5 3 mongoose.c:6843:tx_dhcp_discover DHCP discover sent
7e8 2 main.c:65:main                   Ethernet: up
81d 3 mongoose.c:6726:arp_cache_add    ARP cache: added 192.168.0.1 @ 90:5c:44:
822 2 mongoose.c:6752:onstatechange    READY, IP: 192.168.0.24
828 2 mongoose.c:6753:onstatechange           GW: 192.168.0.1
82d 2 mongoose.c:6755:onstatechange
       Lease: 86382 sec
bc3 2 main.c:65:main                   Ethernet: up
```

☑ autoscroll

Firmware update available remotely via GUI or API

Captured UART output available remotely via GUI or API

Figure 3: Configured device dashboard.

There, `t=5` means wait 5 seconds while capturing UART output.

Now, we can use those two commands in any software CI platform to test new firmware on a real device, and test the device's UART output against some expected keywords.

Integrating with GitHub Actions

Okay, our software CI builds a firmware image for us, and now we can even test that firmware image on a real hardware! We should add a few extra commands that use the `curl` utility to send built firmware to the test board, and then capture its debug output.

The `curl` command requires a secret API key, which we do not want to expose to the public. In general, the right way to go is:

- Go to the project settings of your project on GitHub (you could clone the repository with the webserver examples), then on / *Secrets / Actions*
- Click on *New repository secret* button
- Give it a name, `VCON_API_KEY`, paste the value into a *Secret* box, click *Add secret*

One of the example projects [22] builds firmware for the RP2040-W5500 board, so let's flash it using a `curl` command and a saved API key. The best way is to add a Makefile target for testing, and let GitHub Actions (our software CI) call it:

```
- run: make -C step-7-webserver/pico-w5500 test VCON_
API_KEY=${{secrets.VCON_API_KEY}}
```

https://github.com/cpq/bare-metal-programming-guide/blob/main/.github/workflows/test.yml

Note that we pass a `VCON_API_KEY` environment variable to `make`. Also note that we're invoking the `test` Makefile target, which should build and test our firmware. In **Listing 6**, you can see the `test` Makefile target.

Explanation:

- line 34: The `test` target depends on the `upload` target, so `upload` is executed first (see line 38)
- line 35: Capture UART log for 5 seconds and save it to */tmp/output.txt*
- line 36: Search for the string `Ethernet: up` in the output, and fail if it is not found
- line 38: The `upload` target depends on `build`, so we always build firmware before testing
- line 39: We flash firmware remotely. The `--fail` flag to the `curl` utility makes it fail if the response from the server is not successful (not HTTP 200 OK)

In **Listing 7**, you can find the example output of the `make test` command described above.

Done! Now, our automatic tests ensure that the firmware can be built, that is it bootable, that it initializes the network stack correctly. This mechanism can be easily extended: Just add more complex actions in your firmware binary, print the result to the UART, and check for the expected output in the test.

Happy testing! ◀

220665-C-01

Listing 6. Makefile for Webserver on the Pico.

[see https://github.com/cpq/bare-metal-programming-guide/blob/main/steps/step-7-webserver/pico-w5500/Makefile]

```
# Requires env variable VCON_API_KEY set
DEVICE_URL ?= https://dash.vcon.io/api/v3/devices/1
test: update
  curl --fail -su :$(VCON_API_KEY) $(DEVICE_URL)/tx?t=5 | tee /tmp/output.txt
  grep ’Ethernet: up’ /tmp/output.txt

update: build
  curl --fail -su :$(VCON_API_KEY) $(DEVICE_URL)/ota --data-binary @firmware.bin
```

Listing 7. Output of the Make Test command.

```
$ make test
 curl --fail ...
 {"success":true,"written":59904}
 curl --fail ...
 3f3 2 main.c:65:main                    Ethernet: down
 7d7 1 mongoose.c:6760:onstatechange     Link up
 7e5 3 mongoose.c:6843:tx_dhcp_discover  DHCP discover sent
 7e8 2 main.c:65:main                    Ethernet: up
 81d 3 mongoose.c:6726:arp_cache_add     ARP cache: added 192.168.0.1 @
90:5c:44:55:19:8b
 822 2 mongoose.c:6752:onstatechange     READY, IP: 192.168.0.24
 827 2 mongoose.c:6753:onstatechange            GW: 192.168.0.1
 82d 2 mongoose.c:6755:onstatechange            Lease: 86336 sec
 bc3 2 main.c:65:main                    Ethernet: up
 fab 2 main.c:65:main                    Ethernet: up
```

Questions or Comments?

Do you have technical questions or comments about this article? Email the author at sergey.lyubka@cesanta.com or contact Elektor at editor@elektor.com.

About the Author

Sergey Lyubka is an engineer and entrepreneur. He holds an MSc in Physics from Kyiv State University, Ukraine. Sergey is director and co-founder of Cesanta, a technology company based in Dublin, Ireland (Embedded Web Server for electronic devices: https://mongoose.ws). His passion is bare-metal embedded network programming.

Nucleo Boards Programming with the STM32CubeIDE
Hands-on in more than 50 projects
Dogan Ibrahim
elektor

Related Products

- **Dogan Ibrahim, *Nucleo Boards Programming with the STM32CubeIDE* (Elektor 2020)**
  www.elektor.com/19530
- **Dogan Ibrahim, *Programming with STM32 Nucleo Boards* (Elektor 2015)**
  www.elektor.com/18585

WEB LINKS

[1] This guide on GitHub: https://github.com/cpq/bare-metal-programming-guide
[2] Sergey Lyubka, "A Bare-Metal Programming Guide (Part 1)," Elektor 7-8/2023: https://elektormagazine.com/220665-01
[3] Sergey Lyubka, "A Bare-Metal Programming Guide (Part 2)," Elektor 9-10/2023: https://elektormagazine.com/220665-B-01
[4] CMSIS Version 5: https://github.com/ARM-software/CMSIS_5
[5] STM32 CMSIS headers for F4 family: https://github.com/STMicroelectronics/cmsis_device_f4
[6] Step 5 CMSIS folder: https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-5-cmsis
[7] Reference manual RM0090 for STM32F4xx Controllers [PDF]: https://tinyurl.com/stm32f4man
[8] Step 6 Clock Folder: https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-6-clock
[9] Mongoose Network Library: https://github.com/cesanta/mongoose
[10] mongoose.c: https://raw.githubusercontent.com/cesanta/mongoose/master/mongoose.c
[11] mongoose.h: https://raw.githubusercontent.com/cesanta/mongoose/master/mongoose.h
[12] Set of examples from Mongoose: https://github.com/cesanta/mongoose/tree/master/examples/device-dashboard
[13] Webserver Example, net.c: https://raw.githubusercontent.com/cesanta/mongoose/master/examples/device-dashboard/net.c
[14] Webserver Example, packed_fs.c: https://tinyurl.com/packedfsc
[15] Step 7 Webserver directory: https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-7-webserver
[16] Example: Successful run of Continuous Integration: https://tinyurl.com/bmgoodrun
[17] vcon.io: https://dash.vcon.io/
[18] Raspberry Pi W5500-EVB-Pico board: https://docs.wiznet.io/Product/iEthernet/W5500/w5500-evb-pico
[19] Flashing ESP32: https://vcon.io/docs/#module-flashing
[20] Network Setup: https://vcon.io/docs/#module-registration
[21] Wiring: https://vcon.io/docs/#module-to-device-wiring
[22] Webserver Example for Pico: https://tinyurl.com/picowebeg

elektor TV

Elektor LabTalk

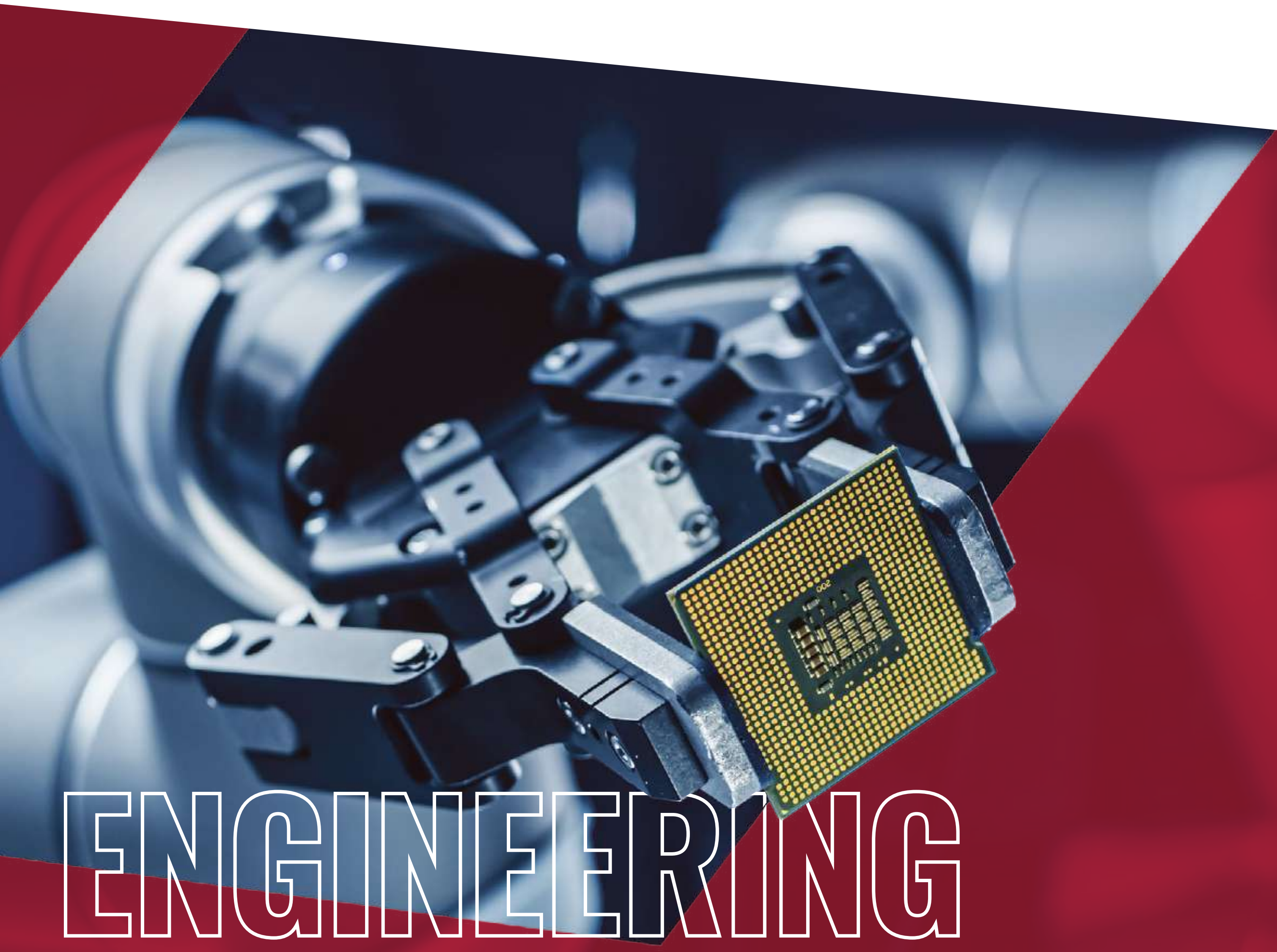
In our latest livestream, our experts explored the Raspberry Pi 5 and discussed the possibilities it offers your projects!

The new features of the Raspberry Pi 5 were discussed in this livestream on the Elektor TV YouTube channel. With its quad-core Arm Cortex-A76 processor running at 2.4 GHz, it's up to three times faster than its predecessor.

You can rewatch the Elektor LabTalk here
www.elektormagazine.com/labtalk13

Missed the livestream? Don't worry, we've got you covered! Watch the highlights of the livestream and more content about the Raspberry Pi in our YouTube playlist on Elektor TV.

www.elektormagazine.com/pi-playlist



ENGINEERING
A BETTER
WORLD

IMAGINE THAT.
WE CAN.

Let us take you on a journey of discovery. Whether you design, build or maintain, we have ingenious technologies and solutions to help engineer a better world for everyone.

LoRa, a Swiss Army Knife (2)

The Hardware and Software

By Gilles Brocard (France)

Having covered the LoRa protocol in the first part, we will get into the details of the construction of a LoRa transmitter/receiver using a small commercial module. Controlled by an Arduino Nano and a program completely dedicated to LoRa configuration and transmission/reception, the final assembly becomes a true LoRa Swiss Army knife.

The small printed circuit board described here consists of a LoRa module, an Arduino Nano board, a 5 V regulator, some LEDs, and screw termnals for easily connecting any type of sensor, controls, and an OLED I^{2}C screen or additional LEDs. This assembly is very flexible, partly due to the program. It constitutes a real LoRa Swiss Army knife.

In short, the schematic (**Figure 1**) covers the connections between the Arduino Nano and the E220-900M30S module, with a voltage regulator and an LED confirming the presence of +5 V, and two other LEDs indicating a transmission (TX) or reception (RX).

Other components can be connected via connector K2. Ports A0 to A5 are analog inputs or digital I/O, according to the configuration. In addition, A4 and A5 may be configured as an I^{2}C bus to connect I^{2}C sensors or even a small 128×64-pixel OLED display. A6 and A7 are two dedicated analog inputs. The Nano's analog inputs have a resolution of 10 bits. The +5 V terminal can function as either input or output.

The LoRa module's DIO2 port is connected to the antenna switch's Tx input and to T1's gate, which serves as a logic inverter. The drain controls the Rx input of the same antenna switch.

LED1 (white) signals a transmission (TX). LED2 (green) indicates the reception (RX) of a valid LoRa signal, but with no information on the content. In addition, the program activates port A2 for one second if the expected message is detected. This makes it very easy to use the assembly as a long-range remote control.

Power

The input voltage on K3 should be no more than 24 V. It's also possible to power the assembly directly with 5 V via K4. In both cases, the power supply should be capable of delivering up to 750 mA.

A heatsink for IC1 is optional, as, in LoRa mode, the transmit/sleep duty cycle is very low (1% maximum), and the average dissipation remains very small. In reality, one could go up to 35 V on the input. In effect, even with a 750 mA draw for an output signal of 30 dB, the dissipation is only around 0.25 W (assuming that the circuit is "asleep" for the rest of the time).

The Printed Circuit Board

The printed circuit board (**Figure 2**) measures 87×50 mm and can be mounted in most types of enclosures (optional). The antenna output is made with an SMA connector, perfectly suitable for a near-gigahertz frequencies (this is not the case with BNC).

The wiring is easy. The LEDs, resistors, and 100 nF and 1 µF capacitors are all SMD 1206 types — easy to solder. The other components are through-hole. If your PCB does not have plated through-holes (**Figure 3**), don't forget the straps, the ground plane links, and those for the connections of D4 and D5 between the two sides of the board.

If you want to use long-range links (10 km or more), it is advisable to house the circuit in a waterproof enclosure, directly below the antenna, which should be mounted high up (2 m from ground level is optimum) in order to minimise HF cable losses. Another solution would be to replace the Arduino Nano with an ESP32. In this case, the messages will be relayed via Wi-Fi before being sent on via LoRa. The modifications to the sketch are minor, and an adapted version can be supplied.

A TX/RX switch may be connected between the D2 and Ground points of terminal K3. This allows for operating mode changes at any time between transmission (open) and reception (closed). If the assembly is only used in RX mode, you should bridge K1's contacts 1 and 2. Alternatively, to keep it fixed in TX mode, leave the contacts open.

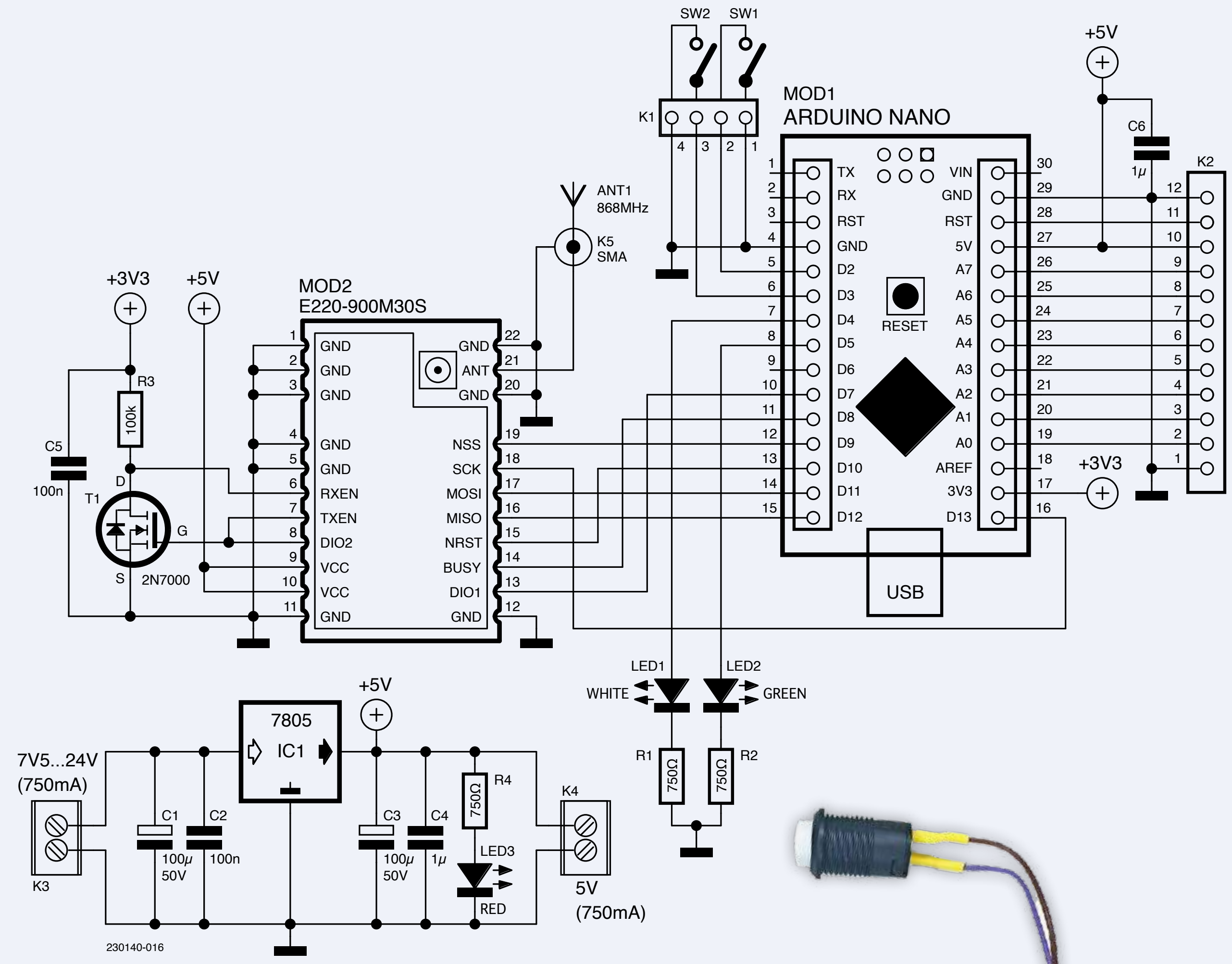


Figure 1: LoRa Swiss Army knife schematic.

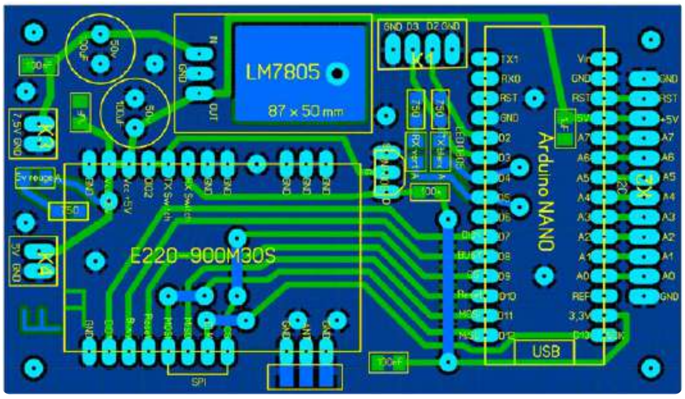


Figure 2: The PCB and source code files are available on the webpage for this article [1].

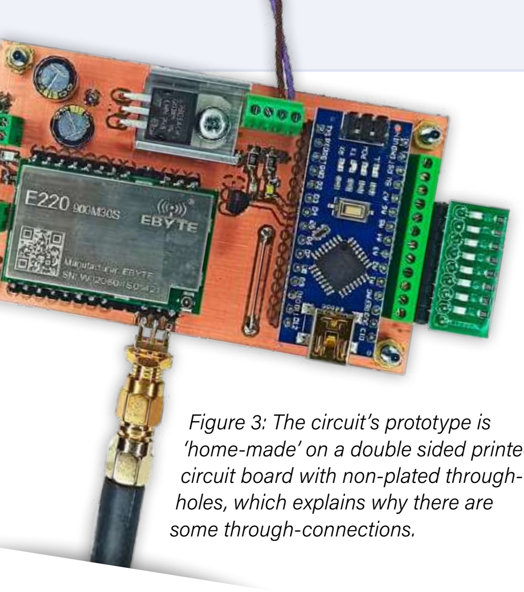


Figure 3: The circuit's prototype is 'home-made' on a double sided printed circuit board with non-plated through-holes, which explains why there are some through-connections.

The Program

The sketch [1] allows you to modify the LoRa configuration easily for any kind of experimentation and attempts at long-range LoRa transmission. There is a record to break here!

The software also allows hot-switching between sending (TX) and receiving (RX) messages. The connection of many analog or digital (I²C) sensors is also possible. See the sidebar, **Quick Guide to the LLCC68 Control Program** to see how the sketch is organized.

LoRa Optimization

The LLCC68 can inform the sender of the signal reception quality so that it can adapt the transmission parameters (power, SF, CR, etc.). This is one of the means of adjustment used by the LoRaWAN layer.

The RSSI (received signal strength indicator) is the value of the signal power that the receiver can return to the transmitter. This value can be used to modify the transmission parameters, increasing or decreasing the transmit power in order to maximise the autonomy while maintaining a sufficient level of transmission.
The LLCC68 can measure the minimum receive level that allows it to extract a signal without errors. This value is very low in the case of the LLCC68; it can be as low as -129 dBm as a function of the chosen parameters.
The SNR (signal-to-noise ratio) is the ratio between the power of signal received and the ambient noise level that the LLCC68 measures outside the period of data reception. In the case of LoRa, the noise level can be considerably greater than that of the signal (negative SNR).
The *GetStats* instruction returns the total number of packets received, the number of those received with errors and the number of those received with header errors. The *ResetStats* instruction resets all the statistics.

Reading the 106-page datasheet is not essential, but it is very useful for giving more in-depth information on some of the subjects referred to in this article. We advise you also to read the remarkable document written by the teachers of the *Lycée Dorian* [3] [Note: This is in French.]. It covers the physical LoRa layer, but also that of LoRaWAN, and LoRa demodulation, for which Semtech does not give any information.

The Main LoRa Configuration Parameters

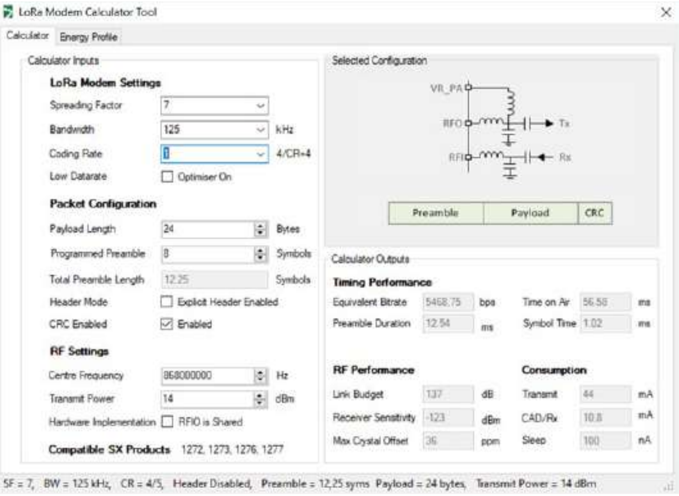
All of the LoRa configuration parameters are found between lines 83 and 111:

Line 83: The toggle for LoRa operation.
Line 84: Configuration of FS, BW, CR, and LDRO.
Line 99: The parameters of the send packet are configured in nine bytes; the last three are common with FSK modulation.

For each of these three program lines, the first byte is not a configuration byte, but the opcode (the command) that tells the LLCC68 what it must do with the following bytes. There are 41 opcodes in total. See the datasheet for more details. You will also find the addresses of the 36 configuration registers there.

SF — The Spreading Factor

The analog reception path's sensitivity can only be modified by the gain of the input preamplifier, which is configurable (see line 154 of the sketch). But, as we have seen, the more we increase the value of SF, the more the data rate decreases. Also, as the efficiency of the message's digital reconstruction is inversely proportional to the data rate, the digital sensitivity thus increases with the value of SF (as does the transmission time). This increases the sensitivity of the analog part, permitting higher sensitivity figures, and hence greater range.



Calculator Inputs

LoRa Modem Settings
Spreading Factor: 7
Bandwidth: 125 kHz
Coding Rate: 1 4/CR+4
Low Datarate: ☐ Optimiser On

Packet Configuration
Payload Length: 24 Bytes
Programmed Preamble: 8 Symbols
Total Preamble Length: 12.25 Symbols
Header Mode: ☐ Explicit Header Enabled
CRC Enabled: ☑ Enabled

RF Settings
Centre Frequency: 868000000 Hz
Transmit Power: 14 dBm
Hardware Implementation: ☐ RFIO is Shared

Compatible SX Products 1272, 1273, 1276, 1277

Selected Configuration

Calculator Outputs

Timing Performance
Equivalent Bitrate: 5468.75 bps
Preamble Duration: 12.54 ms
Time on Air: 56.58 ms
Symbol Time: 1.02 ms

RF Performance
Link Budget: 137 dB
Receiver Sensitivity: -123 dBm
Max Crystal Offset: 36 ppm

Consumption
Transmit: 44 mA
CAD/Rx: 10.8 mA
Sleep: 100 nA

SF = 7, BW = 125 kHz, CR = 4/5, Header Disabled, Preamble = 12,25 syms Payload = 24 bytes, Transmit Power = 14 dBm

Figure 4: There's also an online version of Semtech's LoRa modem calculator.

SF also corresponds to the number of bits transmitted during the period T_S, i.e. per symbol.

The transmission time of a symbol T_s and of a message of 24 bytes is shown in **Table 1**. Semtech has a calculator on their website [2] (**Figure 4**) that allows a precise evaluation of transmission time, power consumption, and link budget. Don't hesitate to use this, because, although it is not made specifically for the LLCC68 (rather for the SX1272 to SX1277), the results are almost identical. Increasing the value of SF to the next value almost doubles the transmission time, as can be seen in Table 1.

The value of the data rate (bit rate) is R_B and is calculated by the formula

$$R_b = SF \cdot (BW / 2^{SF}) \cdot 4/(4+CR) \text{ (bits/s)} \qquad (1)$$

Because SF is equal to the number of bits per symbol, one can calculate the baud rate R_s (symbol rate):

$$R_s = R_b / SF \text{ (baud)}$$

As the three values, BW, SF, and CR are available, we can easily configure the transmission speed that we want.

CR — The Code Rate

LoRa allows the reliability of the transmission to be increased by increasing the signal's redundancy. For CR = 1, the redundancy rate is 4/5, which means five bits are used to code every four bits of data. For CR = 4, the rate is 4/8, where eight bits are used to code four data bits. As every advantage has a downside, increasing CR decreases R_B (see equation 1) and, in consequence, increases the message transmission time.

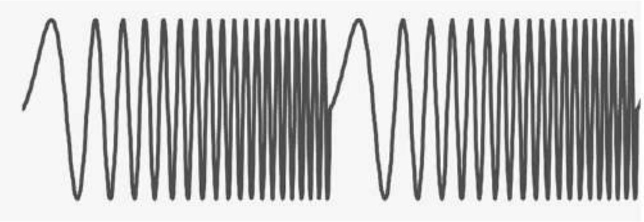


Figure 5: Representation of two **chirps** *(with modulation exaggerated for illustration).*

Table 1: Transmission Times as a Function of Spreading Factor SF.

SF	Rb (bit/s)	Rs (baud)	Preamble time (ms)	Ts (ms)	Transmission time (ms)
5	15,625	3,125	3.135	0.256	16.37
6	9,375	1,562.5	6.27	0.51	30.85
7	5,468.75	781.3	12.54	1.02	56.58
8	3,125	390.6	25.09	2.05	102.91
9	1,757.8	195.3	50.18	4.10	185.34
10	976.5	97.7	100.35	8.19	370.69
11	537.11	48.8	200.70	16.38	659.46

These values are without header. The presence of a header will increase these durations. Here, CR = 1, BW = 125 kHz, a message (payload) contains 24 bytes, and a preamble consists of 8 + 4.25 = 12.25 symbols. $Ts = 2^{SF}/BW$.

Spreading factor SF also corresponds to the number of bits transmitted during time T_s. *Values* R_b *and* R_s *in the table are those obtained with the Semtech data calculator compatible with products SX1272 to 1277. Although it's not specifically stated, they are also compatible with the LLCC68. Note: Transmission delays* R_b *of a bit of the payload (user message) and of a bit of the preamble are not identical.*

BW — The Bandwidth

The bandwidth is also referred to as the spectral congestion of the modulated signal. The variation of a *chirp* (**Figure 5**), which extends from F – df to F + df, determines the occupied bandwidth. Because, R_B is proportional to BW, according to equation 1, the message's transmission time can be reduced by choosing a larger value for BW. In contrast to the other parameters, there is no practical disadvantage to such an increase, apart from the congestion of the frequency band.

Optimization of the Data for Low Data Rates

Taking account of the potentially long packet time for high values of SF (10 or 11), the low data rate optimization (LDRO) option must be activated. It increases the reliability of transmission vis-à-vis any frequency variations during the time of transmission and reception of the packet. Its use is required if the duration of a symbol, T_S, is greater than 16 ms. For example, one can see in Table 1 that, if SF = 11, LDRO must be used, because $T_S > 16$ ms. The transmitter and the receiver must use the same LDRO value.

Quick Guide to the LLCC68 Control Program

The 48 KB sketch consists of 571 lines, of which the greater part is detailed commentary (in French). They refer to the page numbers of the LLCC68 datasheet. The program is simply structured so it may be adapted to any kind of application.

Line 1: The program includes the *SPI.h* library. It is part of the Arduino built-in libraries. It does not need to be installed.
Lines 5 to 15: Two tables, one to store the the message to be transmitted (TX), the other to store the received message (RX).
Lines 17 to 28: Listing of the initialization of the input and output terminals to enable adaptation for any type of application.
Lines 31 to 48: Declaration of the necessary variables for the sketch's internal calculations, and the addresses of the LLCC68's registers used by the program.
Lines 50 to 56: Declaration of the LLCC68's general configuration variables.
Lines 57 to 81: Declaration of the variables specific to the FSK modulation (the last three packet parameters are common, for LoRa).
Lines 83 to 111: Declaration of the LoRa modulation variables.
Lines 113 to 170: Declaration of the configuration variables for all of the LLCC68's RF functions.
Line 172: The required Arduino `setup()` function.
Line 197: The other required Arduino function, `loop()`. It is divided into two parts. The RX/TX mode switch determines which part is executed.
Lines 232 to 277: Five functions for correction of certain limitations of the LLCC68 (see chapter 15 of the datasheet).
Lines 278 to 311: The `Message()`, `busy()`, and `RX_Wait()` functions, used inRX mode. In the `RX_Wait()` function, an infinite loop waits for the reception of a valid LoRa message. The mode switch allows exiting from the loop and passing from RX to TX mode.
Lines 313 to 355: `TXRX_Setup()`, the function common to both RX and TX, then `TX_Setup()` for the configuration of the TX mode and `RX_Setup()` for RX mode.
Line 356: The `TX_Send()` function, which starts the TX mode and the transmission of a message.
Line 371: The `RX_Read()` function, which starts RX mode and puts the program in an infinite loop waiting for a valid LoRa message.
Line 411: The `Fsk_Setup()` configures the FSK mode.
Line 433: `Lora_Setup()` configures the LoRa mode.
Lines 454 to 483: Groups the SPI functions, then the `Status()` function for accessing the LLCC68'S status.
Lines 484 to 571: Groups `Affichage_GetRssi()`, `Affichage_GetRssiInst()`, and `Affichage_GetStats()`, functions for measuring RSSI, SNR, and retrieving statistics, such as transmission errors and number of messages), respectively. They can be resent to optimize the link.

Transmission Speed

LoRa message transmission speed and range is dependent on the values of SF, CR, BW, and LDRO, but also on the choice to have a header (explicit option) or not (default option), and the length of the header, if present.

Back to the E220-900T Modules

The software to control the E220-900T22 or 30 modules presented in the first part [5] does not allow you to configure the LoRa parameters as you can do now. They automatically set parameters that are best for getting the desired transmission speed. This is a great pity, because each configuration has its own advantages and disadvantages. For this reason, we chose the E220-900M30S module, which, via its SPI interface, allows total flexibility in its configuration.

Using the Software

All of the commands and configuration, except those thar are specific to your sensor, are gathered at the start of the sketch and are abundantly commented. During the execution of the sketch, on the other hand, you can follow its operation step-by-step in Serial Monitor on your PC, because it includes a lot of `serial.print()` calls. You can also see the sidebar, **Quick Guide to the LLCC68 Control Program**, for more detail.

Sending a Message (TX)

There is a write buffer at line 6 that must contain your message every time the sketch starts. For this, replace the ASCII codes in lines 7 and 8 with your message (127 bytes maximum, in ASCII). If your message is a numeric value (for example the sensor measurement output) you will have to convert it to ASCII before writing it to the table.

Stick to the format of the table. Do not change the first two bytes — 0x0E is the command (opcode) that the LLCC68 needs in order to write to the buffer; 0x80 is the address of the buffer's TX section. However, you need to set the third byte, which is your message's total number of bytes. It's used from the recovery of the message until the moment of receiving. Note: All values are in hexadecimal.

Depending on the length of your message, you may have to adjust a few more parameters: The maximum length of the message (line 99, bytes 4 and 7), and also lines 125 and 126 for reception. On line 122, you may change the location of the reception and transmission buffers, and the parameters specific to LoRa are on lines 83 to 112.

Put switch SW_1 in position TX (open) and then restart the sketch. Your message will then be sent periodically, separated by a programmable delay (line 229). You will probably have to readjust the ratio of the busy/sleep time so it does not go above 1% (legal limit for LoRa transmissions in France).

Receiving a Message (RX)

Put switch SW_1 in the RX position (closed) and then restart the program. It will then wait for the reception of a valid LoRa message, which will automatically be placed in the receive buffer (line 14). You then just have to read it.

Another way of doing this is to modify the RX function, line 37, which makes it possible to recover a received message in the loop, lines 381 to 396. It has an `if` comparison for recognizing the byte 0x0A, which triggers the `Message_Valide` flag variable to be set to 1. The example illuminates an LED on the board, but could be used as a remote signal.

Some Application Ideas

A GPS beacon for an animal, human or another object is an application example of our LoRa Swiss Army knife. But, the program that we have developed is very versatile and suggests a large number of other applications, such as:

- Gathering of data (soil humidity, sunlight levels, etc.) for a farming application or a weather station.
- A home automation system extended to a garden, a park, several houses, a whole street, or even a village.
- Surveillance (fire, security) or telemetry of a wide or isolated area (swimming pool, sports ground, industrial unit, etc.).
- Central distribution of the time, phases of the moon and those of the main planets, to receiving clocks situated around a house or larger building (museum, town hall, etc.) or throughout a district in a city.

There are many more, but it's probable that some other ideas will have taken root in your imagination... ◀

230140-B-01

The European Standard

You may transmit using the LoRa protocol in the 868 MHz band, respecting the Euorpean rules. Here is a summary:

1. Do not transmit outside the 868 MHz band.
2. Do not use a bandwidth greater than BW = 500 kHz.
3. Do not exceed 14 dBm (25 mW) of radiated power in the air. This may correspond to a lot more at the output of the module, depending on the losses between the module and the antenna.
4. Do not transmit for more than 1% of the time (duty cycle). That is to say that if the time of transmission of a message is 0.1 second, we may only send a new transmission every 0.1 × 100 = 10 seconds, or six times per minute (which is adequate for many applications).

Let's look at an example. If, like the author, your companion is a beagle dog that's prone to running off, and you want to know its GPS coordinates so you can find it on its wanderings, a collar with a GPS receiver, a LoRa transmitter, and a small battery will solve your problems. Six times a minute, it will send the position of your dog to within a radius of 10 km in open country. The gadget consumes very little power and we have a light, small system with great autonomy.

About the Author

Gilles Brocard is a retired consulting engineer and trainer of analog electronic simulation software (LTspice). He is also a specialist in on-board converter design (SMPS).

Questions or Comments?

Do you have any technical questions or comments about this article? Contact the author at brocard.gilles.b26@gmail.com, or the Elektor team at editor@elektor.com.

Related Products

- **Great Scott Gadgets HackRF One Software Defined Radio (1 MHz to 6 GHz) (SKU 18306)**
  www.elektor.com/18306
- **CircuitMess Chatter – DIY LoRa Communicator (SKU 20407)**
  www.elektor.com/20407
- **Claus Kühnel, *Develop and Operate Your LoRaWAN IoT Nodes*, Elektor 2023 (SKU 20147)**
  www.elektor.com/20147

WEB LINKS

[1] Source code and printed circuit board: https://elektormagazine.com/230140-B-01
[2] LoRa modem calculator tool: https://lora-developers.semtech.com/build/tools/calculator
[3] Lycée Dorian: "Caractérisation de l'interface radio LoRa d'un réseau de communication LoRaWAN," eduscol.education.fr: https://elektormagazine.fr/lycee-dorian
[4] Official radio rules for France [French]: https://arcep.fr/uploads/tx_gsavis/21-1589.pdf
[5] Part 1 of this series: https://elektormagazine.com/230140-01

R&S®ESSENTIALS
NEXT-GENERATION OSCILLOSCOPE: EVOLVED FOR MORECHALLENGES
Precision made easy.

NEW R&S®MXO5-Oscilloscope

The R&S®MXO 5 series delivers breakthrough oscilloscope technology to speed up your understanding and testing of electronic systems.

With both four- and eight-channel models, the specifications of the R&S®MXO 5 series impress, making the instrument stand out above other industry choices.

In addition, the R&S®MXO 5 series oscilloscopes are the epitome of cutting-edge technology by delivering quick and accurate results. With custom technology and game-changing features, these oscilloscopes are the perfect tools for understanding circuit behaviors.

More at:
www.rohde-schwarz.com/product/mxo5

ROHDE&SCHWARZ
Make ideas real

MEMS Microphone

Design and Construction

By Peter Riccardi (United States)

You can spend thousands of dollars on a free-field microphone. But, before you do, consider this MEMS-based design, which will give you superb performance at a fraction of the cost.

Acousticians and acoustical engineers often rely on measurement microphones to acquire physical signals during experiments. Typical condenser microphones from companies such as PCB Piezotronics run a bill of over 1000 USD for a free-field microphone. The quality of such microphones is superb, but there are alternatives available to the hobbyist and enthusiast which will get you the performance you need for a fraction of the cost. Here, a MEMS-based design is explored with the goal of having as low a noise floor as practically possible and a flat frequency response in the audio band of 20 Hz to 20 kHz.

The prototype needs to be inexpensive, simple to design and assemble, and integrate seamlessly into the typical workflow of an experimental acoustician. Typical acoustical experiments are powered from IEPE DAQ inputs, so this microphone will run off IEPE excitation [1]. This will drive the power requirements necessary for the preamplifier and the MEMS elements. MEMS microphone technology has improved in recent years, so their noise floor is pretty low. Analog averaging by utilizing a paralleled array will further decrease noise; this tactic will be leveraged here.

Electrical Design

There are many types of MEMS microphones which could be specified for use in this project. Here, the TDK ICS-40300 has been chosen due to its extended low-frequency response (by way of an extended back volume): 6–20,000 Hz, reasonable sensitivity: -45 dBV at 94 dB SPL, and low noise floor: -108 dBV. Mechanically, this will be packaged in a 0.5″ nominal diameter housing. This will give it a small form factor and allow easy use in an acoustic lab with existing calibrators, mic stands, etc. A BNC connector on the rear side of the housing will bring in the IEPE power and output the voltage-modulated waveform. An array will be designed to lower the noise floor. The target is four elements in the array, providing a maximum noise improvement of 6 dB. To achieve the widest dynamic range possible, the preamplifier will provide gain to bring the maximum expected output of the microphone to the supply rail. Here, that will be 3.3 V. Per the datasheet, the maximum voltage output is 0.355 V_{rms} @ 130 dB SPL. Some quick calculations provide more intuitive values.

First, the sensitivity should be expressed with units of mV/Pa. Sound pressure level (SPL) has the form:

$$\text{dB SPL} = 20 \log_{10} (P_{rms} / P_{ref})$$

The SPL is a decibel value. The P_{rms} is the root-mean-square of the measured pressure. P_{ref} is the reference pressure (20 µPa in air). A simple plug-and-chug produces 1.417 Pa_{pk} at the rated sensitivity of -45 dBV. Now, the voltage (in mV) needs to be derived in a similar fashion from the -45 dBV given value. Here, the "reference" voltage would be assumed to be 1 V. The governing equation would be:

$$\text{dBV} = 20 \log_{10} (V / 1\,V)$$

producing a peak voltage of 5.623 mV_{pk}. Note: SPL, by definition, assumes a root-mean-square pressure waveform. The voltage is a simple decibel, so both the numerator and the denominator of the logarithm function are assumed to be peak values. Dividing the peak voltage by the peak pressure produces the sensitivity as:

$$\text{sensitivity} = 1.417\ Pa_{pk} / 5.623\ mV_{pk} = 3.968\ mV / Pa$$

The sensitivity will be multiplied by any gain in the preamplifier, so the sensitivity can be rewritten as:

$$\text{sensitivity} = 3.968\ {}^{mV} / {}_{Pa} * A$$

where:

A = gain

This is a far cry from the more typical measurement grade 50 mV/Pa, but a necessary compromise to use the low-voltage MEMS elements

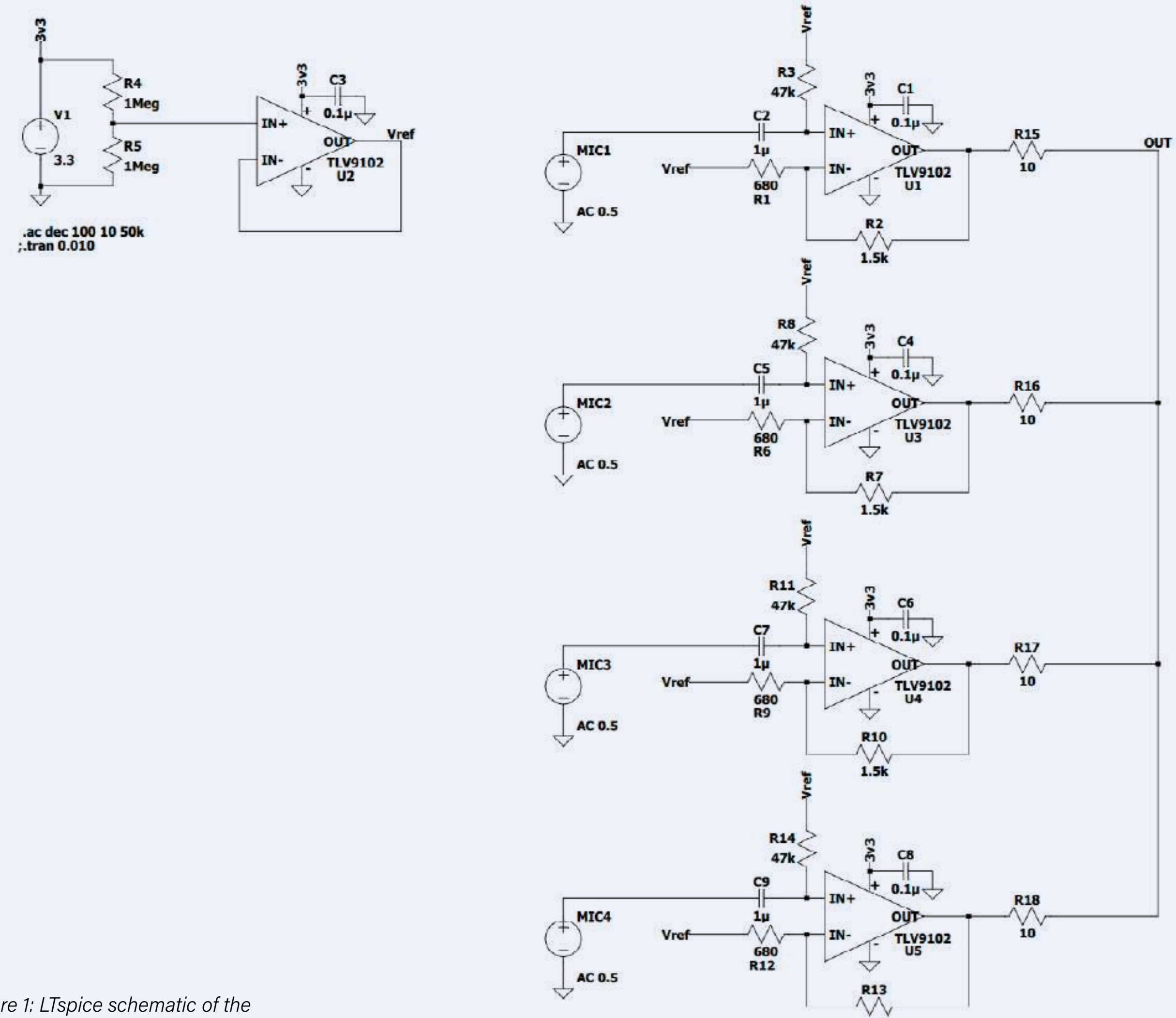


Figure 1: LTspice schematic of the proposed, simplified circuit.

with a wide dynamic range of pressure amplitudes — up to 130 dB SPL. A typical condenser microphone can swing as high as 20 $V_{pk\text{-}pk}$ for lower voltage models. This sensitivity can be increased by adding gain to the preamplifier. To maximize the dynamic range of our system, the headroom must be considered. Low-voltage, 3.3 V-rail-operational amplifiers will be chosen. This guarantees low power consumption, while maximizing the amount of voltage swing the system can produce. The datasheet specifies the maximum voltage output of the microphone at the highest rated pressure it can measure; this can be used to calculate the gain. Per the datasheet, the maximum output voltage is:

$V_{out} = 0.355\ V_{rms}$

The peak-to-peak output voltage is approximately:

$V_{out} = 0.355 * 1.414 = 1\ V_{pk\text{-}pk}$

The total swing of a rail-to-rail operational amplifier running on a 3.3 V unipolar rail is about 3.3 V. The gain, to achieve maximum swing, would therefore be:

$$A = \frac{output}{input} = \frac{3.3\,V_{pk-pk}}{1.00\,V_{pk-pk}} = 3.3$$

This also implies the use of a virtual ground at 3.3 V / 2 = 1.65 V and will therefore be the reference voltage, V_{ref}, used in this design. The microphone elements can have a substantial DC voltage offset, so the outputs from each microphone need to be AC coupled via a simple RC filter. Here, a 1 µF capacitor and a 47 kΩ resistor are used, producing a cutoff frequency of:

$$f_c = \frac{1}{2\pi RC} = \frac{1}{6.28*47{,}000*0.000001} = 3.4\ Hz$$

The relatively high cutoff frequency was chosen so that a small 0603 capacitor with suitable voltage ratings could be used.

Circuit

The circuit in **Figure 1** is a simplified schematic illustrating the proposed topology. The output node consisting of 10 Ω resistors forms an analog averaging circuit. Theoretically, the noise generated by all the op-amps and the noise generated (electrically) in the MEMS microphones are uncorrelated. Thus, it is possible to achieve a maximum improvement in the noise floor of 6 dB. This issue is complicated by the fact that the noise that could appear on the output is not solely due to the electronics. The MEMS microphones are subject to acoustical noise due to the molecular interactions of air in contact with its diaphragms.

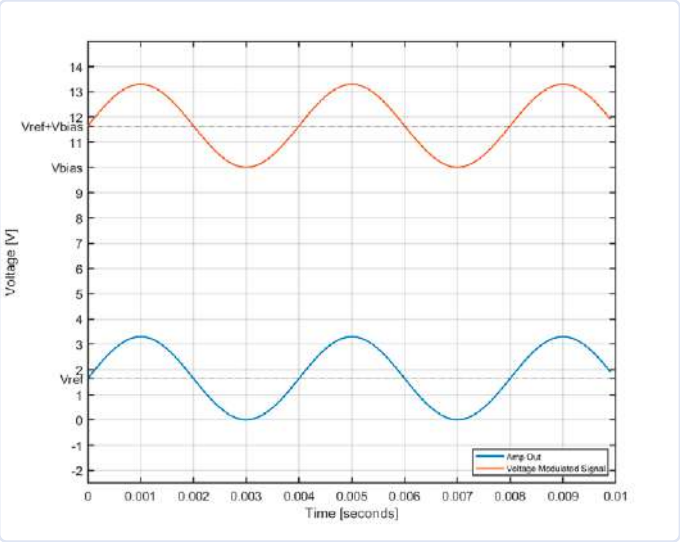


Figure 2: Typical waveforms and biasing in an IEPE-powered system. Here, a 250 Hz sinusoidal waveform is the injected signal. The blue trace would be the output as taken from the averaging node. The output is biased to the reference voltage, V_{ref}, and can be seen oscillating around that DC voltage. By shifting that voltage by another voltage, V_{bias}, the output can be programmed to operate at a DC voltage equal to about half of the anticipated compliance voltage of an IEPE DAQ. The NI 4431 was used as a reference, which produces 2.1 mA at a voltage to exceed 20 VDC. Therefore, something around 10-15 VDC was chosen as the system's operating point.

The subject of noise in the context of both electrical circuits and acoustical systems is vastly complex, and far beyond the scope of this article. It is assumed that, electrically, the noise sources are uncorrelated, and the hope is that the majority of the acoustical noise is uncorrelated as well. At the very least, the acoustical noise that is uncorrelated will be averaged out. Interested readers can find additional material on the subject of noise here [2][3].

A note regarding Figure 1: A 3.3 VDC rail is generated, utilizing a buffered high-impedance voltage divider to produce a reference voltage equal to half the supply. Four MEMS elements are in parallel, each with their own preamplifier with a nominal gain of A = 3.2. The RC filter removes the DC offset from the MEMs elements while simultaneously injecting the reference voltage into the non-inverting terminal. The outputs of each op-amp are tied together using low-impedance resistors. The ohmic separation allows the voltages to be effectively averaged at their output node without creating a short circuit across the terminals and stressing the output transistors.

IEPE Power and Power Distribution

Integrated Electronics Piezo-Electric (IEPE) is a clever power delivery standard utilized in many data acquisition (DAQ) systems. It ingeniously solves the problem of wanting to source power and receive a signal over a two-wire interface. It does so by leveraging the flexible properties of a constant-current source. Recall that a constant-current source has — in the ideal case — infinite output impedance. Practically, this manifests as a very high slope on the V-I curve. This means that the current source will push a current into the load (the preamplifier and microphone) regardless of the voltage it sees. Thus, the preamplifier can modulate the voltage — called the compliance voltage — and constant current will still flow into the preamplifier for its power rails and distribution. **Figure 2** shows an example waveform seen in an IEPE-based system.

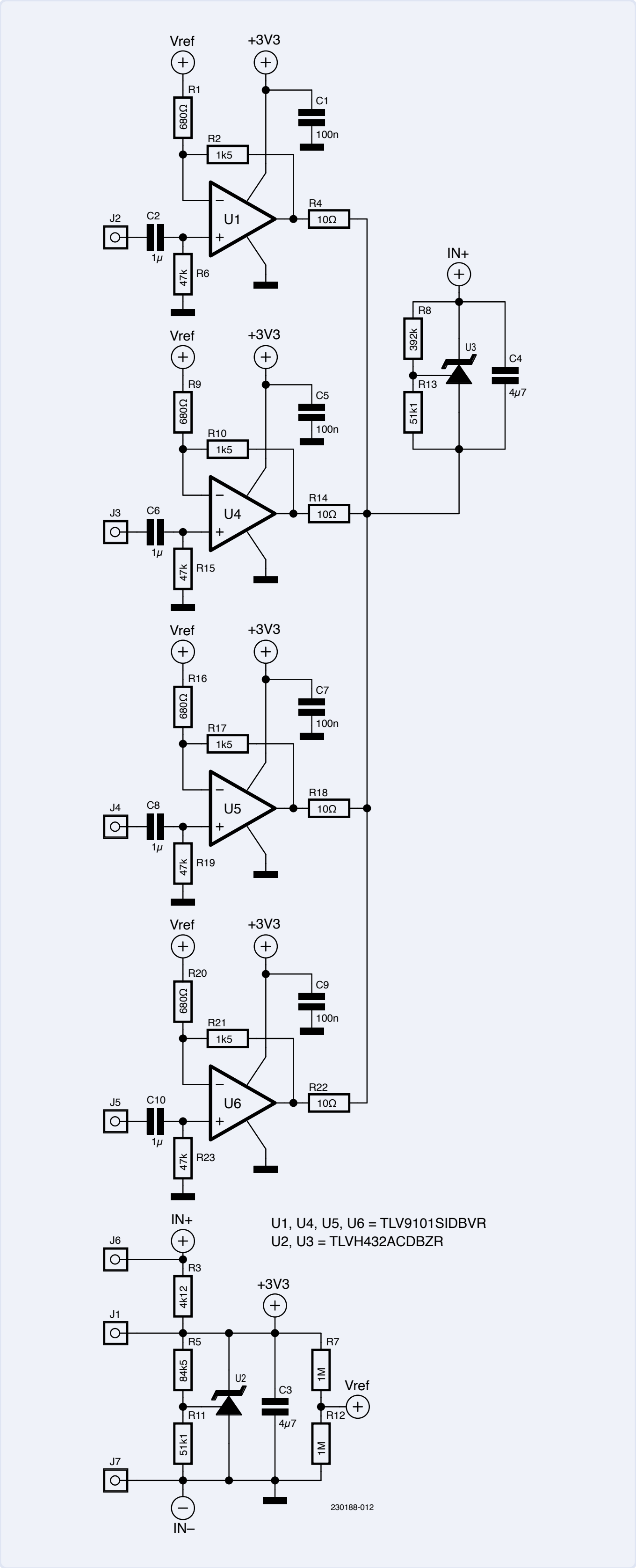


Figure 3: The schematic for the designed, built, and tested circuit. The buffer on the reference voltage (see Figure 1) was initially omitted in the first prototype. This is an error in the design and causes issues. So, do not omit this if replicating the work. The "J" reference designators are through-hole pads to solder "flying leads" to. The BNC connector in the mechanical housing is wired point-to-point to the preamplifier. The positive cable on the connector powers U2, the 3.3 V regulator, and sits at a voltage equal to VIEPE — the set compliance voltage. The output of the summing node drives another Zener regulator set to about 10 VDC, thus producing the total compliance voltage as illustrated in Figure 2.

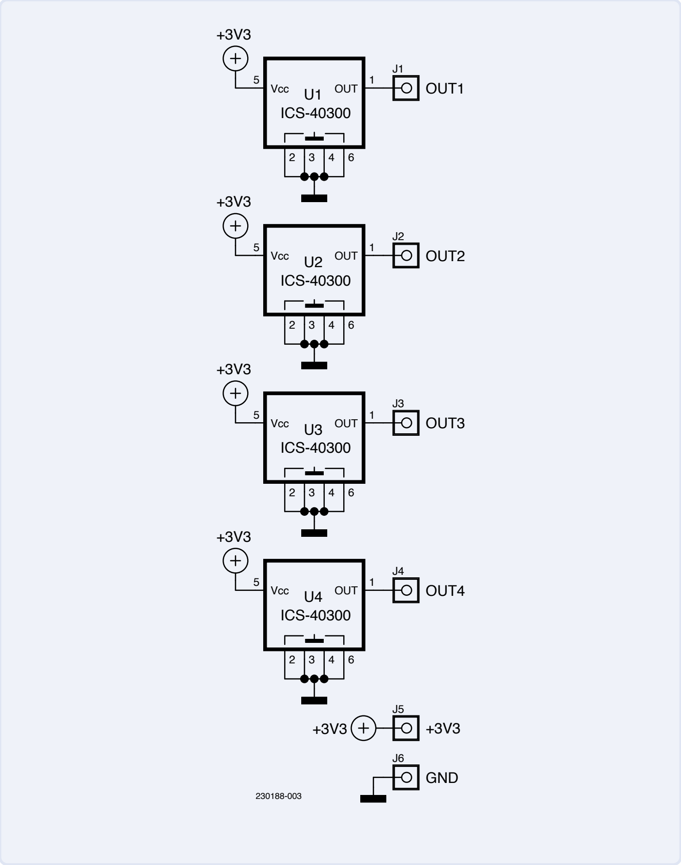


Figure 4: Schematic for the MEMS "carrier" board — a small, circular PCB which houses the four microphones and has a number of through-hole pads to wire the output signals (inputs to the preamplifier) and accept power and ground. As can be seen, there are six total "J" reference designators, four output signals, plus 3.3 V and GND.

The NI 4431 was used as a suitable "reference" DAQ due to the fact that National Instruments is widely used in acoustical research and their IEPE sources are all similar. The 4431 outputs 2.1 mA and has a compliance voltage that exceeds 20 VDC. Provided that the system's DC operating point is within these bounds, the circuit will work. Adequate headroom should be accounted for so that the maximum AC voltage biased by the DC compliance voltage does not exceed maximum operating conditions.

The constant current of 2.1 mA can now be used to design our power distribution scheme. Zener shunt regulators are used due to their low required cathode current and ease of use. The op-amps and MEMS microphones consume the most power. Per their datasheets, they have the following, respectively:

$I_q = 150\ \mu A$ and $I_q = 250\ \mu A$

Assuming five op-amps and four MEMs, the current draw for their power equals 1.75 mA. The remaining 350 μA will power the shunt regulators. The Zener regulator powering the 3.3 V rail will have a resistor setting its cathode current. The TLVH432 was chosen as the Zener reference since it is adjustable. It can operate on 100 μA of cathode current. The bias voltage regulator will not have any series resistance to set its cathode current. It will simply "absorb" whatever the difference is of current consumed by the circuit versus the 2.1 mA of supplied current.

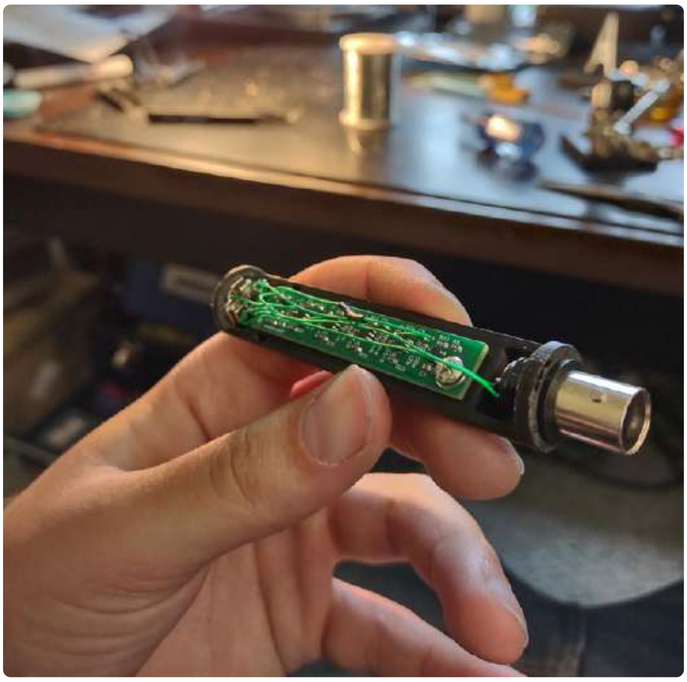


Figure 5: A built microphone, packaged in a 3D-printed enclosure with the lid removed. The final packaging is oversized for a nominally 0.5" design, but it's fairly close for a quick proof of concept.

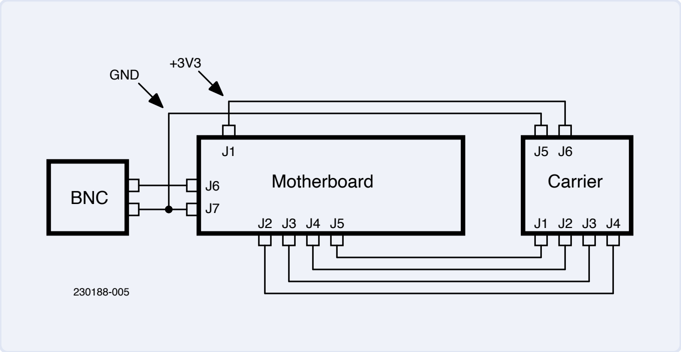


Figure 6: The hookup guide for wiring the circuits as designed here. The blue traces are the signals, black is ground, and red is the supply rail. The "motherboard" is the preamplifier, and the "carrier" houses the four microphones.

A thorough example of all calculations can be found in the exhaustive application note provided by Analog Devices [4]. Due to supply shortages and possible sourcing issues of high demand Zener references, a suitable substitution would be to leverage a floating voltage reference using two resistors and a 2N2222A NPN BJT (hint: check out the "741 op-amp discrete circuitry").

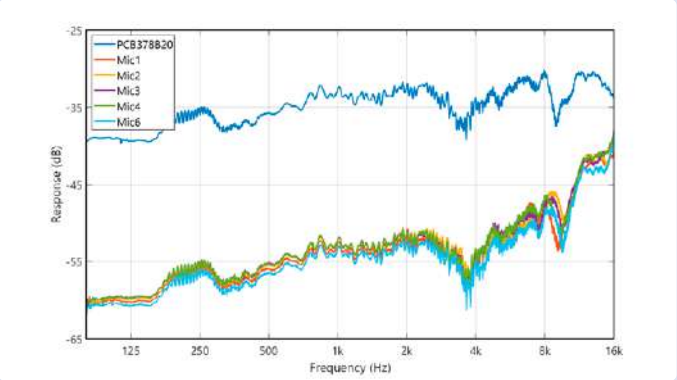


Figure 7: The microphones' (five of them) frequency response compared to a PCB378B20 free-field microphone. These were measured in a semi-anechoic chamber above 200 Hz. The overall trends between the MEMS microphone and the commercial are similar. Most of those "squiggles" are not electrical in nature, but rather due to the complex acoustical interactions of reflections and unwanted bounce off the boom stand and other hard surfaces in the chamber. The notable difference is the upward trend in the MEMS elements above 10 kHz, which follows the FRF given in its datasheet.

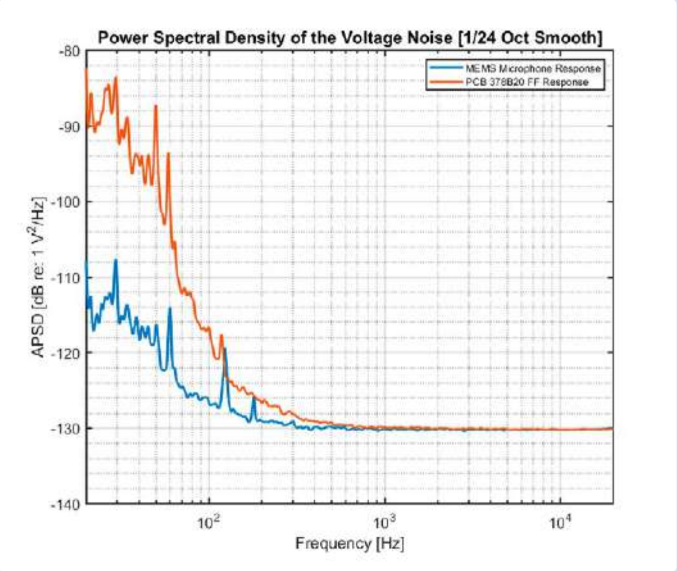


Figure 8: Voltage noise measured in a semi-anechoic chamber with no sounds played in the room and no sensitivity correction made (raw voltage waveforms). Above 1 kHz, both the MEMS microphone and commercial microphone approach the DAQ's noise floor.

Prototype Preamplifier and Microphone

Figures 3 to 6 illustrate the final design. Two PCBs were designed, one small circular PCB to house the four microphone elements, and one small rectangular PCB to house the electronics for the preamplifier. The circuit was wired point-to-point and assembled into a 3D-printed enclosure with dimensions close to that of a nominally 0.5″ measurement microphone. For measurement results, see **Figure 7 and 8**.

Conclusions and Corrections

The first thing to note is the obvious error in the schematic shown in Figure 3. The design shown in that schematic was captured in rapid time in order to meet funding requirements for the original project. There was no time for simulation in any capacity. As such, the obvious error of the overloaded rail was missed. While the output current from that rail is very small, so the normal "check" to see if the voltage divider is "stiff" passes — there is another issue. The op-amps are actively driving their outputs via the low-impedance feedback to make the inverting terminal equal to the non-inverting terminal. There is only 680 Ω of resistance between the non-inverting terminal and the output of the divider network. The divider network has 1 MΩ separation between the tap point and the regulated rail. So, effectively, the op-amp via its low impedance feedback actively drives that node to be equal to its own output. The effect is that the reference node follows the output and essentially collapses to a unity gain follower, but with additional interference possible. All these measurements were performed on the original circuit as shown in Figure 3. So, the noise measurements may be improved from here. Any interested reader working out the math might notice the low sensitivity of the measured Frequency Response Function — this is because the amplifier was operating in unity-gain mode, so it was down by about 10 dB. When this board gets re-spun, the buffer will absolutely be required for proper operation.

Assembly

Future work entails resesigning the mechanical enclosure to be more robust. Assembly was quite difficult as there were several small features approaching the printer's wall thickness limits with a 0.4 mm nozzle. The best solution is likely to go to a 1″ nominal diameter microphone design. The PCBs were always intended to be made utilizing a rigid flex design for ease of assembly. Due to timing constraints for the early prototype, this was not possible. Combining the carrier board and preamplifier on a single design without the point-to-point wiring solution will greatly simplify complexity and reduce assembly time. The total cost to build one of these units was around 30 USD; the measured noise figure and frequency response is quite good considering the cost.

All files can be found in the project's Hackaday post [5].

230188-01

Author's Note

I would like to thank the SPRAL team at The Pennsylvania State University for their help in acquiring the anechoic chamber measurements, and thank Mr. Zane Rusk especially for his tireless support. Similarly, I would like to thank the Acoustical Society of America Chapter at The Pennsylvania State University for their support in continuing the project and presenting the work. PCBway sponsored the build of five MEMS microphones and sourced all parts for free. I would like to thank them for supporting the project, empowering graduate students to design, build, and test these measurement microphones at absolutely zero cost.

Questions or Comments?

Do you need to know anything more about this project? You can reach the author at pjriccardi@pjroses.co and the Elektor team at editor@elektor.com.

About the Author

Peter Riccardi is currently working in aerospace as an avionics (electronics) engineer. His formal education is Mechanical Engineering, having received a degree in 2018 from Rowan University. In 2022, Peter received his M.S. in Acoustics from The Pennsylvania State University. Peter is interested primarily in audio electronics and electroacoustics; his research proposed a linear equivalent circuit transduction model of the air motion transformer, a novel electroacoustic device. His personal projects have included custom rack-mounted active crossover networks with integrated power amplifiers, low-noise turntable preamplifiers, custom-built turntables (mechanical gimbal, motor drivers, packaging, etc.).

Related Products

- **Elektor Audio Collection (USB Stick)**
  https://elektor.com/19892

WEB LINKS

[1] IEPE [Wikipedia]: https://en.wikipedia.org/wiki/Integrated_Electronics_Piezo-Electric
[2] D. Self, "Chapter 1: Basics," Small Signal Audio Design, Focal Press, New York, NY, 2015, pp. 1–16.: https://amzn.to/43dVyTa
[3] T. B. Gabrielson, "Mechanical-Thermal Noise in Micromachined Acoustic and Vibration Sensors," in IEEE Transactions on Electron Devices, vol. 40, no. 5, pp. 903-909, May 1993, doi: 10.1109/16.210197: https://ieeexplore.ieee.org/document/210197
[4] Analog Devices, "IEPE-Compatible Interface for Wideband MEMS Accelerometer Sensors," CN-0532, 2020: https://bit.ly/CN-0532
[5] P. Riccardi, "MEMS Based IEPE Powered Instrumentation Microphone," Hackaday.io: https://hackaday.io/project/185762-mems-based-iepe-powered-instrumentation-microphone

BALVER ZINN®

SN100CV® The alternative to high-priced, silver-containing alloys for high-end applications.

Available as ingots, solid wire, cored solder wire with StarCore®234-E and the JEAN-151 solder paste, supplemented by the latest REGI-RED flux.

Tools to Try Before You Solder

Simulation and 3D Modeling Tools That Can Be Used for Free

By Stuart Cording (Elektor)

You've got a great idea for a circuit or product but want to test a few core ideas before you warm up the soldering iron or switch on the 3D printer? Today, it's simpler than ever to figure out what's possible, and what's probably not. Simply power up your PC or laptop and give it a go!

Back in the late 1990s, while studying engineering, the power of the PC as a tool alongside the soldering iron was just beginning to show. Circuits could be described in text format and simulated using SPICE from a DOS command, outputting blocky VGA graphs of signal amplitude and phase. Another tool, Windows-based, enabled simulation of digital circuits, provided the machine didn't crash during our allotted lab session. However, outside the university's facilities, access to such tools was reduced to time or feature-limited versions. If you were lucky, you might have been able to sit a little longer in the office during a work placement and use whatever development software the employer had available.

Fast forward to today, and the industry's attitude to software has changed significantly. Many free tools are on the market, providing more than enough capability for most makers to test an idea before ordering parts online. Others are free for makers and non-commercial activity, giving you time to learn whether it meets your needs before taking the plunge and paying for a license. And, thanks to the power of modern web browsers, some tools don't even require software to be downloaded. Just log in and get designing!

Wokwi

When initially navigating to the online tool, Wokwi [1], it's a little unclear whether the site can offer a tool with enough power to be of use. But, as we engineers know, looks can be deceiving. Behind the simple entry point lies a wealth of hardware and software that makes it possible to simulate complex applications based on Arduino, Raspberry Pi Pico, STM32, and ESP32 hardware. Access to Wokwi's capabilities is free, although any projects you create will be public. For more privacy and access to advanced features, there is the reasonably priced "Wokwi Club." Projects start with selecting a microcontroller board, and then users are encouraged to explore one of the many existing projects as a starting point.

The browser splits into two, with the left side dedicated to the traditional code development approach of the selected board, while the right side offers a visual display of the hardware (**Figure 1**). There are a range of parts in the library, from simple passive components and logic elements to complete modules from SparkFun, seven-segment LEDs, LCDs, servos, and IR remotes and receivers. You can even start a debugging session if you are comfortable with the GDB [2] source code debugger. Otherwise, simply click the play button to execute your code and simulate your hardware. Displays update, LEDs blink, and sounders buzz, enabling you to put everything through its paces.

The environment is well documented [3], has a blog providing tips, and it's supported by an active Discord channel. Arduino projects are saved in the *.ino* format, enabling them to be copied to the Arduino IDE with ease, and there's a library manager, too. The circuit diagrams are saved in the easy-to-read JSON format. To document your project and allow others to learn from your experience, you can write notes in the markdown format in a *readme.md* file. Efforts are also underway to support the tool through the popular Microsoft Visual Studio Code IDE and offer a cloud-based continuous integration (CI) environment to automate testing.

Get Started

- Arduino alarm clock with real-time clock module: *https://tinyurl.com/alrmclkino*
- ESP32 MQTT weather logger in MicroPython: *https://tinyurl.com/espmqttwl*
- Raspberry Pi Pico matrix keypad with LEDs: *https://tinyurl.com/picokeypad*

QSPICE

SPICE, Simulation Program with Integrated Circuit Emphasis, is a tool that came out of the University of California, Berkeley, in the 1970s. It helped engineers simulate their analog circuits before committing them to silicon using complex mathematic analysis, leveraging the FORTRAN language's power for numeric computation. Today, variants of SPICE can be found as standalone tools or hidden in the back-end of circuit design packages.

Now, there's a new version available. QSPICE [4] has been launched by Qorvo, an American semiconductor company growing its portfolio of power devices, which recently added silicon carbide (SiC) products through its acquisition of UnitedSiC in 2021. Their simulation tool has been developed by Mike Engelhardt, the author of the LTspice tool promoted by Analog Devices (previously Linear Technology).

On the surface, QSPICE looks much like any other simulation tool, providing a workspace to draw circuits (**Figure 2**) and start a simulation, displaying results in the time domain (**Figure 3**). However, compared to LTspice, the manner of interaction follows standard Windows user interface rules, such as pushing forward on the mouse scroll wheel to zoom in. However, that's where the similarity ends. In an interview, Engelhardt explained that this version of SPICE is what he would have created 25 years ago if he'd known what he knows today. Perhaps the most significant change is the support for mixed-mode simulation with a compiler for Verilog (which is written inside a C++ file — check out the *VerilogCounter* demo listed below), allowing the implementation of digital devices. He also states that this tool simulates the circuit given rather than solving the complex equations as best as possible, unlike other SPICE tools.

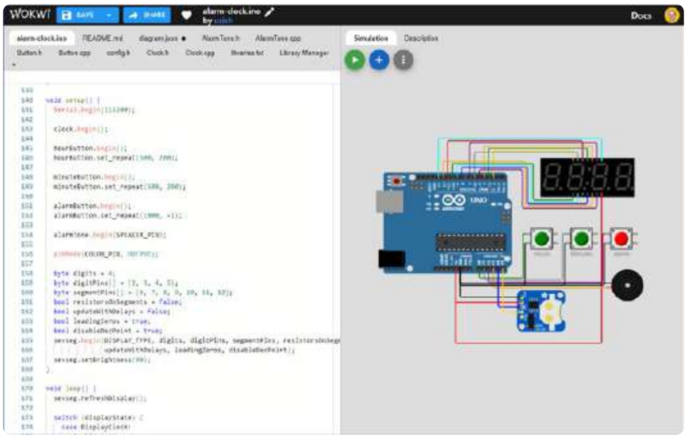
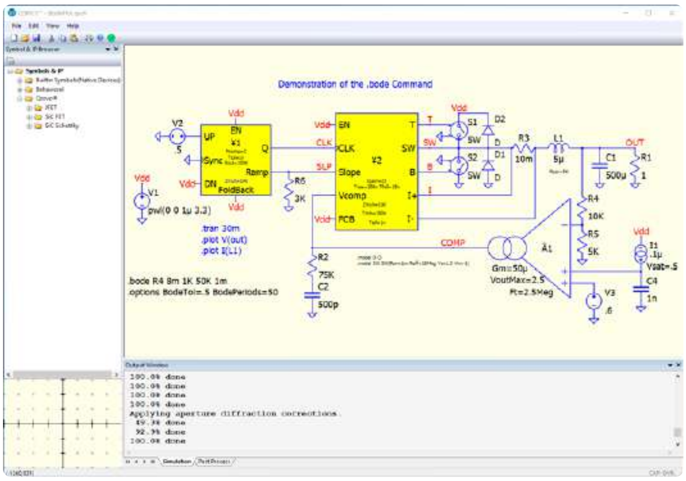


Figure 1: Wokwi's user interface is kept simple, enabling you to concentrate on the code (left) and the circuit (right).



```
100.0% done
100.0% done
100.0% done
100.0% done
Applying aperture diffraction corrections
 49.3% done
 92.3% done
100.0% done
```

Figure 2: QSPICE offers a familiar circuit design interface while your PC's GPU boosts its simulation performance.

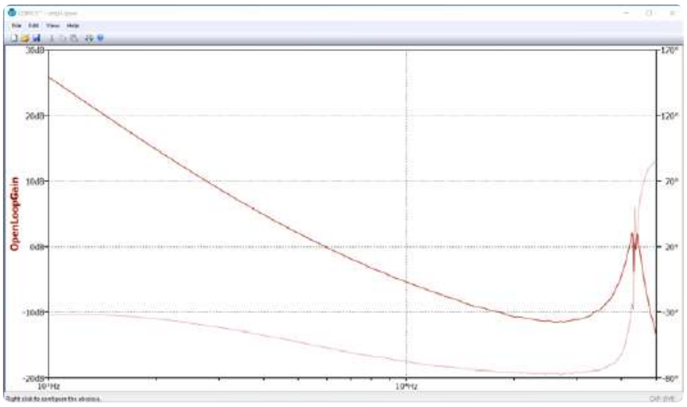


Figure 3: The simulation output, here a Bode plot, is familiar but with a higher resolution than other SPICE tools.

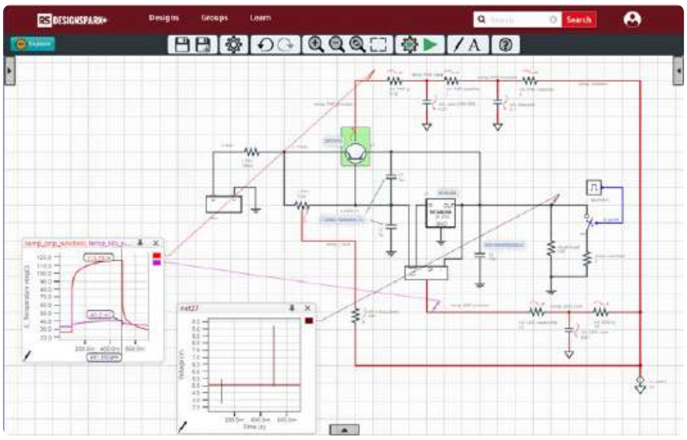


Figure 4: Thermal aspects of a circuit can also be simulated in DesignSpark's Circuit Simulator, an online tool.

QSPICE also uses the latest features of the Windows operating system that provide simplified access to the computer's GPU through a standardized software API. During simulation, this means that waveform data can now be used as presented, avoiding the need for compression and the associated artifacts that result. Furthermore, the graphics are much improved both when displaying circuits and simulation results. Another benefit is the speed, proving to be faster on benchmark circuits and passing all tests — a simulation failure rate of around 15% is typical for such tools.

One significant simulation improvement concerns the saturation of components like SiC JFETs and gallium nitride (GaN) switches in the "on" region. Qorvo hopes that this will help engineers develop advanced, next-generation power management systems and become an industry tool that other semiconductor vendors contribute to. Currently, around 1,000 parts are supported, along with 20 example projects.

Get Started

Go to the menu and select *File → Open Demo...*, then:

- Try digital simulation of a counter with *VerilogCounter.qsch*
- Examine a resonant switch mode power supply (SMPS) with *SMPS.qsch*
- Create a bode plot with *BodePlot.qsch*

DesignSpark Circuit Simulator

The DesignSpark program from RS Components has been going for over a decade, evolving into a community that empowers engineers and engineering activists. Supporting them is a range of free PCB design and 3D-modeling tools, together with subscriptions that open up access to advanced features. Added to this tool list is Circuit Simulator [5], a browser-based tool by Siemens PartQuest Explore. It supports analog, digital, and mixed-signal circuits, but perhaps the most exciting aspect of this offering is its multi-domain simulation.

Most circuit simulators simulate according to the circuit given. You can quite happily pass several amps through a small-signal diode, oblivious to the fact that the magic smoke would have been released in the first second of operation. Multi-domain simulation means that electro-thermal and electro-mechanical simulation is also available.

This is most easily explained using one of the available example circuits. The Current Boost Regulator project using an Onsemi NCV4264 LDO generates a 5 V output into a 100 Ω load. An additional test circuit switches a 2 Ω load in and out, providing an expected droop and peak in the voltage output. Around the circuit, an additional thermal circuit has been constructed (**Figure 4**). This consists of heat generated by a transistor plus an Electrical Power to Heat Flow element that wraps around two-pin electrical components such as the LDO.

Obviously, such a simulation requires an understanding of the thermal properties of the circuit when constructed, such as heat transfer through the PCB, which is not something everyone has experience with. The tool includes a range of thermal and electro-thermal components such as transistors and diodes with electro-thermal interfaces, a Cauer thermal network, and a Peltier thermo-electric cooler.

Also included are magnetic and electro-mechanical components, such as electromagnets, linear DC motors, and three-phase motors. If needed, new components can be added to your library. One approach is to input the essential details from the component's datasheet. Otherwise, they can be described using SPICE or VHDL-AMS [6].

Like all of the Design Spark range of tools, registration is required to access Circuit Simulator, and with the free Explorer plan, simulation is limited to one minute. Upgrading increases this to 60 minutes. Another clever feature is the ability to embed your design into your own website, allowing visitors to explore your design, moving waveform probes to view simulation results.

Get Started

Create an account, log in, and check out:

- An open-loop stepper motor controller: *https://tinyurl.com/pqolstct*
- An electric power steering system using a PMSM motor: *https://tinyurl.com/pqpowerst*
- A hydraulic car lift based on a three-phase motor pump: *https://tinyurl.com/pqcarlift*

Shapr3D

As 3D printers emerged, tools were needed to design the 3D objects they would print. One popular tool was SketchUp [7] from a software startup purchased by Google that enabled buildings to be added to Google Earth. Two decades later, the free version of SketchUp is only available as a browser-based tool, feels clunky, and doesn't look like it's changed much.

The problem with 3D tools is that they are challenging to use on the 2D monitors most of us have available. This is something that Shapr3D [8] solves with aplomb. Available as an app on Windows or Mac, or through the App Store for iPad, the software can be downloaded after registering. The user interface makes it easy to understand what 3D drawing options are available at any time. But, the clever piece is that you can seamlessly drop into a 2D drawing mode at any time by pressing the space bar, thereby avoiding the confusion often resulting from working with perspective rendering. So, if you're adept at technical drawing on paper, you'll have a comfortable starting point.

You'll be equally satisfied if you prefer to jump straight in with a 3D drawing (**Figure 5**). Dimensions and angles are easily modified after an object has been drawn or a face pulled, as are fillets and chamfers. Circles are placed centrally using the concentric tool, even if only an arc is available as the reference line. And, to ensure accuracy, the grid and snapping are easy to configure to the needs of your design.

Shapr3D also allows its advanced features to be trialed. For example, an existing 3D model can be turned into a mechanical drawing, or materials can be applied to the surface to get a feel for the final product's appearance (**Figure 6**). The free version allows models to be exported as STL or 3MF files for 3D printing, although the resolution is limited to the low setting. Otherwise, your files are stored in the cloud.

Get Started

For testing, we downloaded a simple bracket in STEP file format [9] from GrabCad [10], which offers a range of models in its library.

Try First, Build Later

National Semiconductor's infamous analog Staff Engineer, Bob Pease, always claimed that his favorite programming language was solder. And, he's correct — building something will always be better and more fun than simulating its existence. However, if parts aren't to hand, or you want to explore an idea for feasibility, all the tools listed here can be used for free before you place your component order.

230512-01

Questions or Comments?

Do you have technical questions or comments about this article? Email the author at stuart.cording@elektor.com or contact Elektor at editor@elektor.com.

About the Author

Stuart Cording is an engineer and journalist with more than 25 years of experience in the electronics industry. You can find many of his recent Elektor articles at *www.elektormagazine.com/cording*. In addition to writing for Elektor, he hosts a regularly livestreamed interview show, *Elektor Engineering Insights* (*www.elektormagazine.com/eei*).

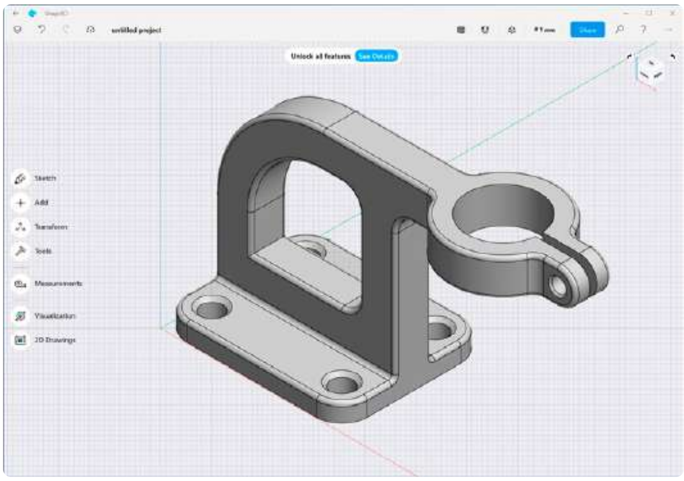


Figure 5: Shapr3D's user interface is very intuitive and supports a seamless transition between 3D and 2D drawing.

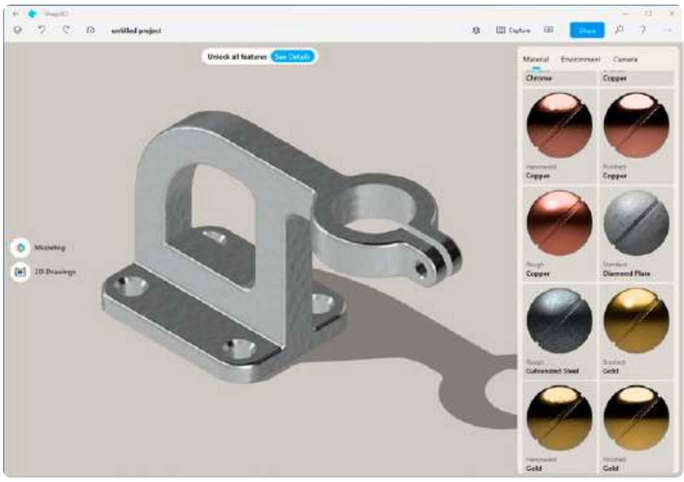


Figure 6: Finished 3D models can have materials applied. The free version doesn't save the materials selected.

WEB LINKS

[1] Wokwi: https://wokwi.com
[2] GDB Debugger Project: https://sourceware.org/gdb
[3] Wokwi Documentation: https://docs.wokwi.com
[4] Qorvo QSPICE: https://tinyurl.com/qspicesimulator
[5] DesignSpark Circuit Simulator: https://tinyurl.com/designsparkcs
[6] VHDL-AMS Wikipedia entry: https://tinyurl.com/wikivhdlams
[7] SketchUp: https://sketchup.com
[8] Shapr3D: https://shapr3d.com
[9] Example 3D model of bracket: https://tinyurl.com/bracketmount
[10] GrabCAD: https://grabcad.com

New Tools from Microchip!

PICkit 5 and MPLAB ICD 5 Available Now!

By Tam Hanna, for Microchip

Long gone are the days when electronics enthusiasts cobbled together homebrew burners to program PIC and similar microcontrollers. Microchip has recently introduced their latest PICkit 5 In-Circuit Debugger/Programmer which won't break the bank and is compatible with many of the Microchip controller families. For professional troubleshooting, the powerful MPLAB ICD 5 In-Circuit Debugger/Programmer is also available. This article will explore their new features and demonstrate how to integrate the new versions into an existing development ecosystem.

Now with USB-C!

Microchip has always been committed to reducing the "cable clutter" in labs of MPLAB developers. The earlier ICD 3 variant used a USB-A connector, while the PICkit 4 came with a Micro-USB connector.

Standardization in the USB connector world now means that both PICkit 5 [1] and the ICD 5 [2] come with a USB-C connector, which is becoming the standard for all modern smartphones. Universal uptake will eventually make USB cable variant anxiety a thing of the past.

For owners of these devices, this comes with several advantages. Firstly, for a developer, it's now easy to use the same cable you use to recharge your smartphone to connect to your development environment. This saves space — and sometimes money — on business trips because you can borrow USB-C cables almost anywhere due to their widespread use in the smartphone industry.

The second innovation specifically concerns owners of the "larger" programming device variant, the ICD 5. Due to its access to more power, the ICD 5 can provide up to 1 A to the application circuit. In many cases, this eliminates the need to carry a second power supply for the device under test. The smaller device is still limited to a current budget of around 150 mA — likely a concession to its significantly smaller hardware.

Regarding actual programming speed, there are no significant advantages with USB-C — both programmers still operate at "only" the USB 2.0 High Speed bitrate, but this should not be a major issue in practice, given the small file size of most microcontroller images. Hardware optimizations, on the other hand, enhance the "perceived speed."

And PoE!

When developing applications that include high voltage sections in the design, it's important to ensure good electrical separation between the high voltage stage and the rest of the circuit, including the processor, low-voltage signals, and sensor signals. One advantage of using a communication link such as an Ethernet network is that it usually uses isolation transformers at each network socket to provide galvanic isolation of the data signals. This provides good isolation between your expensive workstation or PC and any external device.

The latest ICD 5 includes a network port on the back (**Figure 1**). What's new is that it can now supply power to the "entire application" through Power over Ethernet (PoE) if configured appropriately in your PoE network. This can provide substantial current to the application circuit and means you won't need to provide an additional mains adapter to power it.

Fortunately, MPLAB X is also ready to communicate directly with the ICD 5 via the internet, so there's no need to wire an additional USB-C cable between your PC and the programming device.

It's worth noting that this setup also encourages experimentation with Continuous Integration and Continuous Delivery, although older versions of the programming device offered similar functionality when used with an Ethernet connection and a 9 V mains adapter.

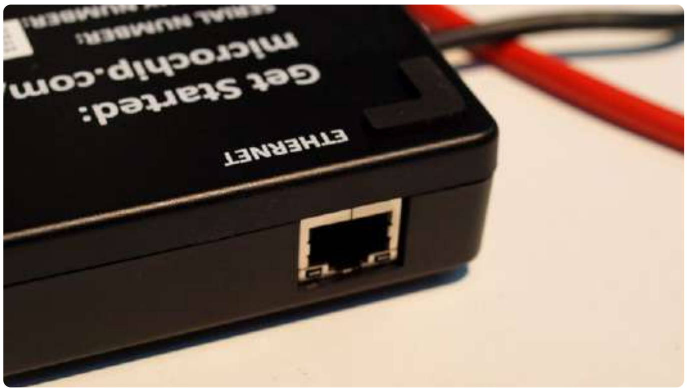


Figure 1: This port also provides Power over Ethernet capability.

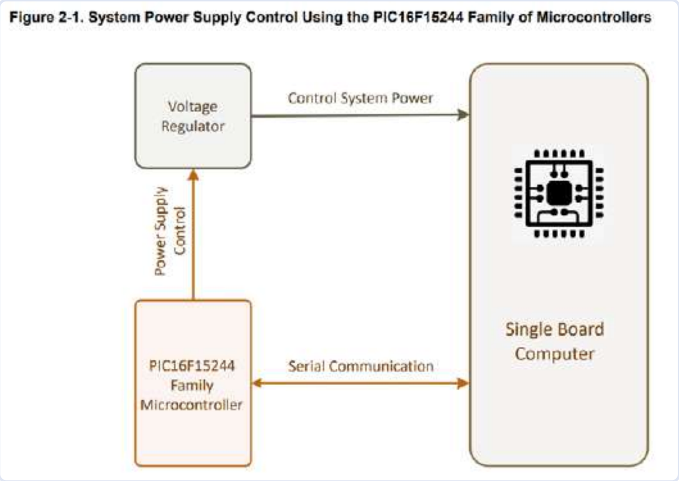


Figure 2-1. System Power Supply Control Using the PIC16F15244 Family of Microcontrollers
Voltage Regulator
Control System Power
Power Supply Control
Single Board Computer
PIC16F15244 Family Microcontroller
Serial Communication

Figure 2: Even a management microcontroller requires firmware! (Source: [3])

The wall wart PSU supplied with earlier versions of the development kit is no longer part of the package in Version 5. This indicates that power and also be acquired through the higher power capability that USB-C or USB 3.0 ports have when compared to earlier USB incarnations.

PICkit 5: Improved Functionality

The actual development of an electronic system is often just the first part of a complex value chain. Those who don't consider "Design for Manufacturing" — and similar aspects — from the start may encounter unexpectedly high costs or problems during manufacturing.

A good example is systems that combine a "management microcontroller" with a process computer, as described in Application Note AN4121 [3] and schematically summarized in **Figure 2**. Failing to plan properly, such as poorly positioning the microcontroller programming port, can significantly impact assembly and product maintenance costs.

While developers can handle delivering firmware to each individual board of a prototype system using MPLAB, this method will become time-consuming and unworkable when the device enters production.

With the PICkit 5 and its *Programmer-To-Go* feature, Microchip offers a solution. Essentially, this feature has been available for several versions, and the secret lies in the microSD slot shown in **Figure 3**, which accommodates a FAT32-formatted memory card.

Using MPLAB, developers can define a firmware image to be deployed automatically, and the programming adapter can then deliver it to connected target devices with minimal effort.

What's also new with the PICkit 5 is the ability to copy more than one firmware image into the *Programmer-To-Go* storage area. This allows a programming device to be used for the "parameterization or final assembly" of multiple assemblies. Initiation of the programming process occurs via the familiar button hidden beneath the logo on the front of the unit, so image selection must be done via a different communication interface.

...and Friends to the Bluetooth Radio Standard

To set up the new programming devices, you'll need development environment MPLAB version 6.10 or higher — the actual installation process is pretty much standard, as you'd expect from MPLAB. Besides the typical installation of the XC8 compiler, which Microchip traditionally handles separately, make sure that, during the initial startup of MPLAB, you grant it access to both local and remote networks in Windows Firewall. This is especially important if you plan to use the larger version of the programming device with its Ethernet connection.

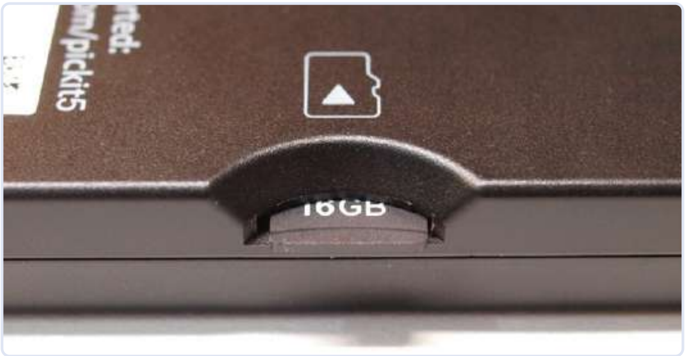


Figure 3: The memory card slot.

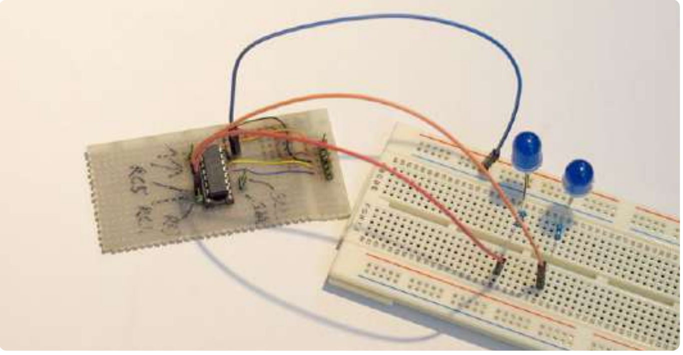


Figure 4: This old board will serve as a test system.

For the next step, you'll need an SD card. I used a spare 16 GB card and formatted it with a FAT32 partition under Ubuntu. Look for a special feature of the Disk Management snap-in; one quirk of the app is that it doesn't create a partition by default, so you'll need to do that separately.

For this setup, the hardware test bed I will use is a PIC16F1503-based controller board left over from a recent automotive consulting project (**Figure 4**).

In the first step, as usual, choose the *File* *New Project* option to generate a project for the XC8 compiler. Next, start the MCC to generate a basic project skeleton. In this description, I assume you are already familiar with PIC programming, so I won't get into general handling in detail.

Not for Windows 7

It's important to be aware that Microchip explicitly advises against using the fifth version of their programming devices with Windows 7. I exclusively used machines running Windows 10 or 11 for this article. There are, however, some rumors circulating on the internet that generally claim it "works," at least when the programming device can regularly be connected to a Windows 10 or Windows 11 computer, where MPLAB performs various firmware upgrades and housekeeping tasks as part of a test delivery

The only crucial point is that two GPIO pins need to be declared as outputs. In the following steps, I will define pins RC4 and RC5 as outputs and then generate the code.

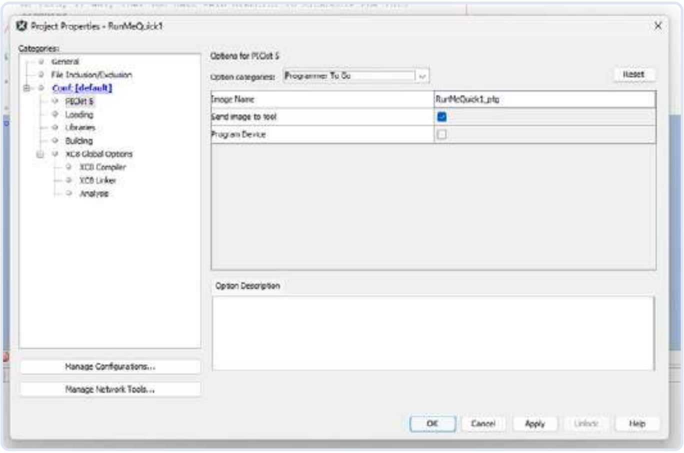
To make it easier to see the differences, I will write a program following a specific pattern within this example project skeleton, referred to as *RunMeQuick1*. This program will switch the two LEDs together at a relatively slow rate.

```
void main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // When using interrupts, you need to
  // set the Global and Peripheral Interrupt Enable bits
    // Use the following macros to:

    // Enable the Global Interrupts
    //INTERRUPT_GlobalInterruptEnable();

    . . .
    IO_RC4_SetHigh()  ;
    IO_RC5_SetHigh()  ;
    while (1)
    {
            IO_RC4_Toggle()  ;
            IO_RC5_Toggle()  ;
            __delay_ms(1000);
    }
}
```

In the next step, open the *Project Properties* view, where you'll find the settings for configuring the *Programmer-To-Go* operation, as shown in **Figure 5**.

The most crucial aspect here is the *Image Name* section, which is the name MPLAB assigns to the written-out file. The *Send image to Tool* option instructs the IDE to load the generated image onto the microSD card of the connected programming device.

Checking the *Program Device* checkbox is optional; it determines whether, in addition to the "actual" *Programmer-To-Go* function, a connected PIC microcontroller should be programmed with the provided image.



Option categories: Programmer To Go

Image Name: RunMeQuick1_ptg
Send image to tool ☑
Program Device ☐

Figure 5: MPLAB assists developers by configuring the **Programmer-To-Go** *mode.*

Using this option can make sense, for example, when you want to update both the *Programmer-To-Go* image and load current firmware onto an evaluation board for a final test.

The actual deployment must then be carried out — absolutely — through the menu shown in **Figure 6**, which initiates an "update" to the *Programmer-To-Go* image.

During deployment, sometimes an error message appears along the lines of `Transmission on endpoint 2 failed (err = -109)`. In my case, this issue was almost always resolved by restarting the PC. By the way, if you've just installed MPLAB, it's also recommended to reboot at this point. The deployment is complete when the status console displays the message `The debug tool is in programmer to go mode`.

Now we will return to the MPLAB start page in this next step to create another — almost identical — project, named *RunMeQuick2*. This will help illustrate the improvements in the second generation of PICkit 5. First, we will need to start the MCC again. The code intended to control the two signal LEDs now looks like this:

```
void main(void)
{
    . . .

    IO_RC4_SetLow()  ;
    IO_RC5_SetHigh()  ;
    while (1)
```

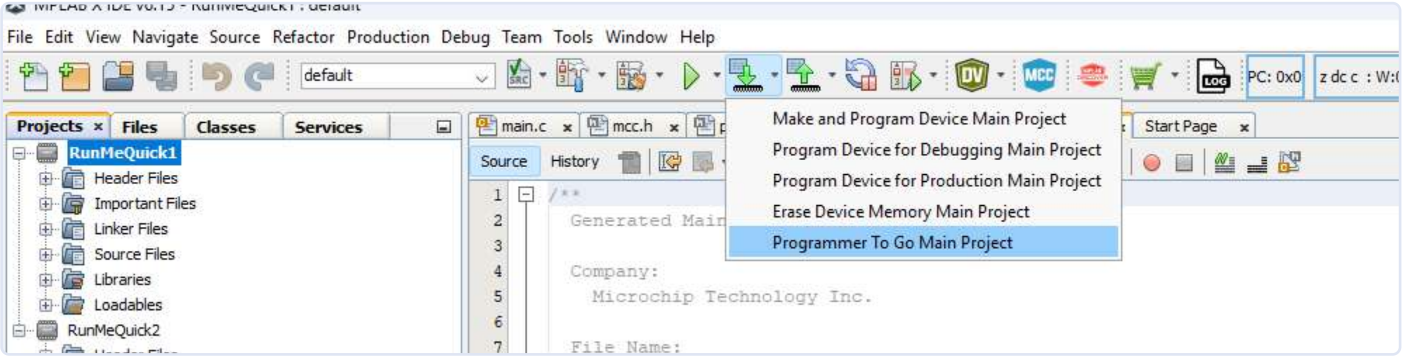
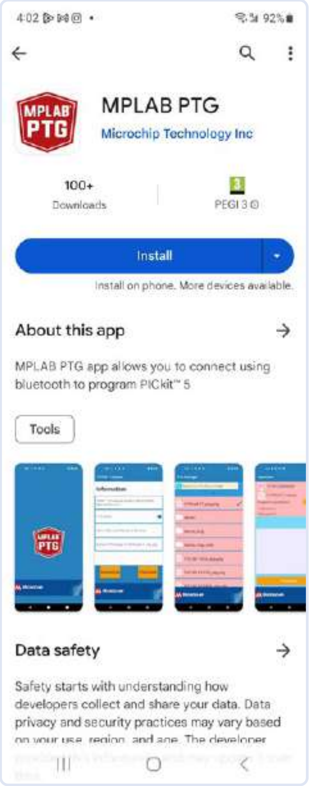

MPLAB X IDE v6.15 - RunMeQuick1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
default
PC: 0x0
z dc c : W:0
Projects
Files
Classes
Services
RunMeQuick1
Header Files
Important Files
Linker Files
Source Files
Libraries
Loadables
RunMeQuick2
main.c
mcc.h
Start Page
Source
History
Make and Program Device Main Project
Program Device for Debugging Main Project
Program Device for Production Main Project
Erase Device Memory Main Project
Programmer To Go Main Project
Generated Main
Company:
Microchip Technology Inc.
File Name:

Figure 6: This entry transfers the image to the SD card.

```
    {
           IO_RC4_Toggle()  ;
           IO_RC5_Toggle()  ;
           __delay_ms(1000);
    }
}
```

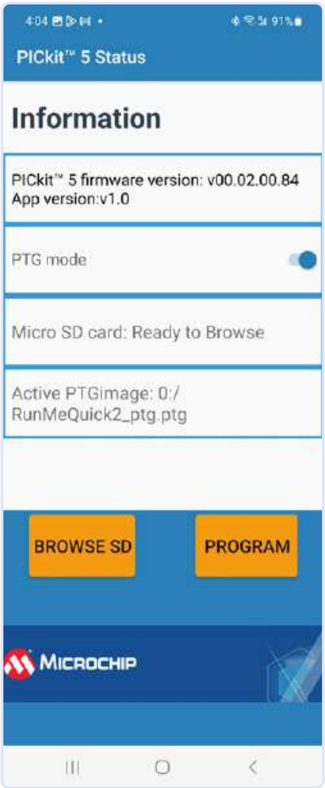
To configure the connected PICkit, logically, you'll need to perform another execution, which should also conclude with the status message `The debug tool is in programmer to go mode`.

In the next step, you can disconnect the programmer from your workstation. I used a regular phone charger to provide power to the PICkit at this point. The flashing bright-green LED indicates that the device is in *Programmer-To-Go* mode, waiting for instructions.



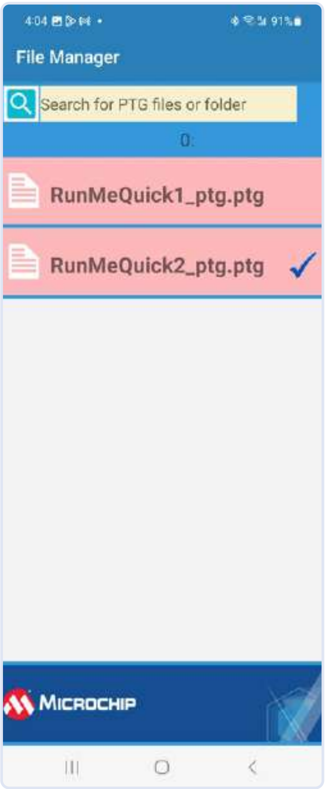
MPLAB PTG
Microchip Technology Inc
100+ Downloads
PEGI 3
Install on phone. More devices available.
About this app
MPLAB PTG app allows you to connect using bluetooth to program PICkit™ 5
Data safety
Safety starts with understanding how developers collect and share your data. Data privacy and security practices may vary based on your use, region, and age. The developer

Figure 7: This app simplifies remote programming.



PICkit™ 5 Status
Information
PICkit™ 5 firmware version: v00.02.00.84
App version:v1.0
PTG mode
Micro SD card: Ready to Browse
Active PTGimage: 0:/
RunMeQuick2_ptg.ptg

Figure 8: The start menu…



File Manager
0:
RunMeQuick1_ptg.ptg
RunMeQuick2_ptg.ptg ✓

Figure 9: …and a selection of deployable images.

Now, we can open Android's Play Store or Apple's App Store to download the new control application, which is only compatible with the Bluetooth module of the PICkit 5. In the following steps, I will use a Samsung smartphone; the application appears in the Play Store as shown in **Figure 7**.

When using Bluetooth LE, note that the program will request various permissions during its initial launch — this is necessary due to Google's requirements, and is not critical or avoidable in any way.

In the next step, the scanning dialog appears, listing all the devices found in the vicinity — in my case, while carrying out the tests, I found it was necessary to sometimes click the *Cancel* button before I could access the list of found programming devices.

After completing the tasks, the system presents the screens shown in **Figures 8** and **9**, allowing you to directly deploy the desired firmware to connected target devices.

These days, the smartphone is the most common device we use that runs a familiar GUI. Running the application on such a device gives newbies to the application a head start compared to those starting from scratch working with the often more complex Project Explorer in MPLAB on a desktop machine. It's worth noting that a smartphone is almost always within reach, so that the (relatively more powerful) workstation is really no longer needed — especially in large deployments, this can significantly reduce costs.

Upgrading My Development Environment

The next experiment involves returning to my Windows 10 workstation, which I used, among other things, to write the textbook referenced in the box below. By default, MPLAB version 5.45 is installed on this machine, which I normally use for my commercial activities.

Now, it's time to pay another visit to the MPLAB website, where I have downloaded version 6.15. The latest version of MPLAB can be installed directly over any existing version.

After successfully completing the installation process, MPLAB 6.15 offers to use the settings from the previous version as the basis for configuring the fresh installation. As always, you'll need to confirm various settings in Windows Firewall. The actual IDE starts after downloading the *Microchip Offline Help* component.

Upon the initial launch of the IDE, an extensive parsing process follows, which seeks to bring all project files up to date, among other tasks. Additionally, it updates various caches created in MPLAB X and populates them with information from the projects.

For the next step, I chose to install demo example *CH9-Demo1* and connected the board, familiar from my textbook, to the programmer and the PC. Compilation was successful, but actual firmware deployment sometimes fails on Windows 10 with the previously mentioned error, `Transmission on endpoint 2 failed (err = -109)`. Apart from this, the new version of PICkit 5 seamlessly integrates into my existing development workflow and, thanks to optimizations, code deployment is, in fact, often faster in many cases.

Measure the Flow Using the ICD5

Microchip has been enhancing its MPLAB IDE with various convenient additional features for some time, aiming to make it easier for developers to visualize measurement and tracking data generated by the embedded application.

One of the interesting features of the ICD 5 is that it can provide this data-visualization engine from MPLAB with information about the power consumption of the connected application circuit.

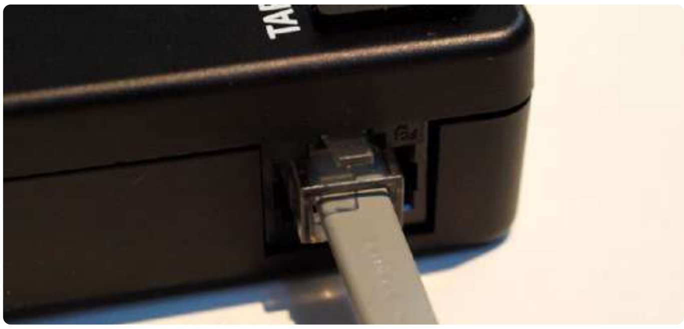


Figure 10: A minor inconvenience when using the ICD 5.

Table 1: The power monitor accuracy, according to Microchip.

Current and Voltage	Resolution	Full Scale
Current	29 µA / step	1.0 A
Voltage	0.2087 mV / step	6.8 V

The specification indicates a current measurement resolution of 0.29 µA — more measurement information for the power monitor is given in **Table 1**.

Now we can click on *Tools Plug-in Downloads* and choose *Microchip Plug-in Manager* in the window that pops up. Check if *MPLAB Data Visualizer* is already listed in the *Installed* section. This will typically be the case with a "fresh" installation.

Logically, you'll need to power the target hardware using the ICD5's built-in power source. The connector used in the ICD 5 is an RJ45 type, while older adapters such as the ICD 3 have an RJ11 cable. As shown in **Figure 10**, the connector makes good contact in principle, but it's not the most robust or secure of connectors

To save time and effort, we can use the LED examples created earlier for the PICkit to test the current measurement feature of the ICD 5. Disconnect the PICkit and connect the ICD 5 — for the sake of convenience, I will avoid the network cable together with its TCP/IP features and will instead connect via a USB-C cable.

Note that during the "initial" setup of an ICD, MPLAB needs to perform a firmware update for the FPGA — this one-time process takes some time.

Interestingly, this process often stalls at 93% and then continues "as usual" (**Figure 11**) — why MPLAB never reports reaching a 100% update status here is not entirely clear.

In the next step, click on *Window Debugging Data Visualizer* to start the *Data Visualizer* welcome screen. Then, in the *Power* section, click the *Play* button to begin data acquisition.

MPLAB responds by displaying the window shown in **Figure 12**, indicating there that no visualization form has been specified.

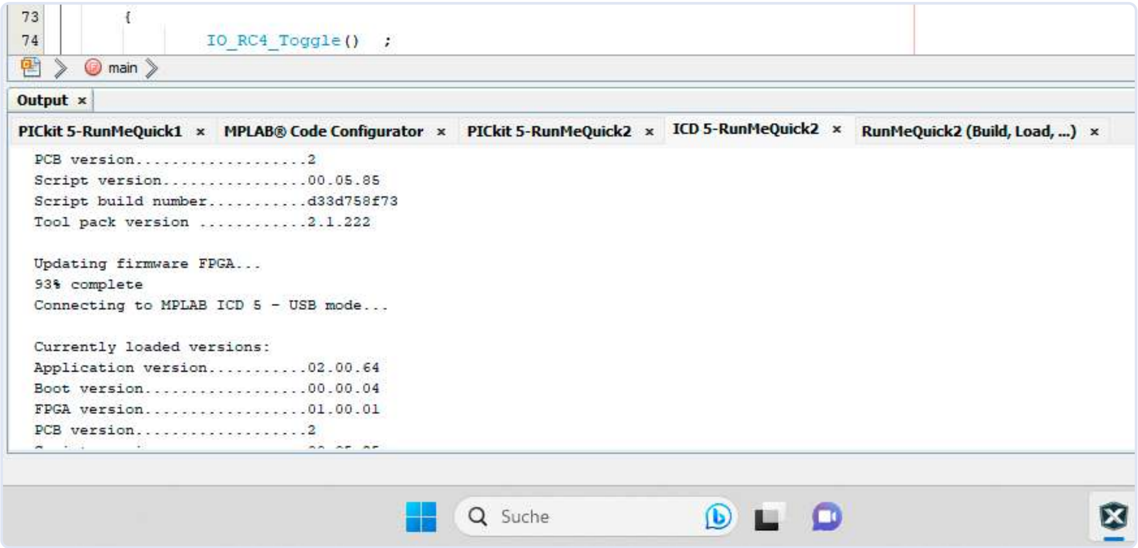
In response, click the *Plot Raw* function, which now shows a chart. If you run the program that alternates blinking between both LEDs, you'll see information about power consumption, as shown in **Figure 13**.

More interesting results can be obtained by using *RunMeQuick1* — this example turns both LEDs on and off alternately (**Figure 14**)

To Sum Up

Microchip has introduced a number of detailed improvements with the PICkit 5 and ICD 5, which will assist developers in both the debugging and "serial production" phase of a project's life cycle. Search engines for electronic parts such as *oemsecrets.com* are able to identify the best real-time pricing & stock levels of components from various distributors. The MPLAB PICkit 5 In-Circuit Debugger/Programmer retails at € 86, while the MPLAB ICD 5 In-Circuit Debugger/Programmer is available for € 360 — in no way too expensive for what is offered. Users of the PICkit 4 who are getting frustrated with the "old" interface may wish to upgrade to the new variant to cut down on the cable count. The power analysis feature is also very valuable and can potentially save significant costs when compared with an SMU or GPIB card. ⏮

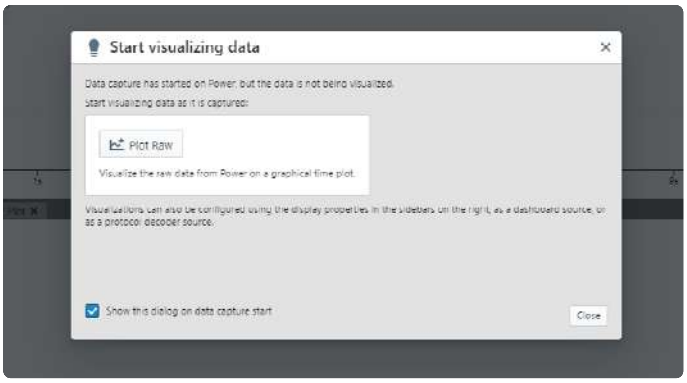
230571-01



```
        {
                IO_RC4_Toggle()   ;
```

```
PCB version...................2
Script version................00.05.85
Script build number...........d33d758f73
Tool pack version ............2.1.222

Updating firmware FPGA...
93% complete
Connecting to MPLAB ICD 5 - USB mode...

Currently loaded versions:
Application version...........02.00.64
Boot version..................00.00.04
FPGA version..................01.00.01
PCB version...................2
```

Figure 11: 93% indicates that the task is fully completed, here!



Start visualizing data

Data capture has started on Power, but the data is not being visualized.

Start visualizing data as it is captured:

Plot Raw

Visualize the raw data from Power on a graphical time plot.

Visualizations can also be configured using the display properties in the sidebars on the right, as a dashboard source, or as a protocol decoder source.

☑ Show this dialog on data capture start

Figure 12: MPLAB requests how the captured data should be represented or visualized.

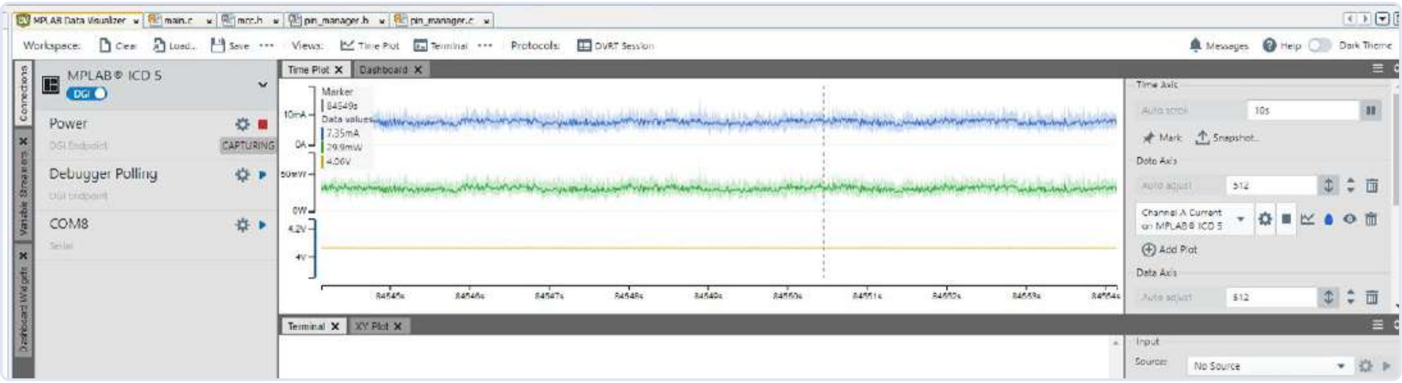


Figure 13: Here, there is always one LED on.

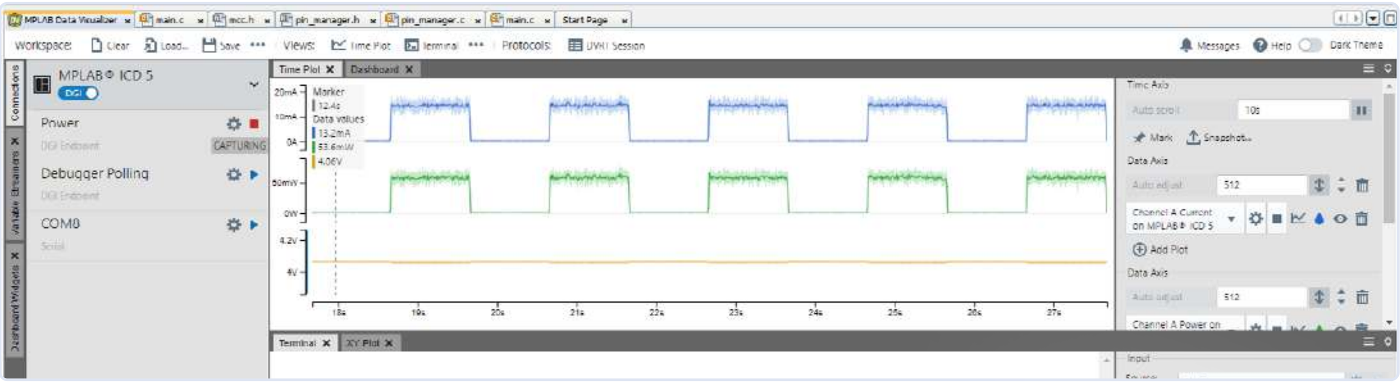


Figure 14: The example clearly shows the changes in current consumption.

Request your 15% discount code for these tools today!

Submit your contact details, select which tool you would like to get a 15% off discount voucher for and you'll receive a unique coupon code to redeem at MicrochipDirect.

https://page.microchip.com/pic5_icd5

WEB LINKS

[1] MPLAB PICkit 5 from Microchip: https://microchip.com/en-us/development-tool/pg164150

[2] MPLAB ICD5 from Microchip: https://microchip.com/en-us/development-tool/dv164055

[3] Application Note AN4121: https://microchip.com/en-us/application-notes/an4121

Rapid Prototyping of Flexible, Stretchable Electronics

How the Voltera NOVA Speeds Up Innovation in Wearable Electronic Systems

By Stuart Cording, for Voltera

Most of your experience with printed circuit boards (PCB) will be with green FR4 substrates. This copper-clad industry classic is turned into the desired design by removing the excess copper. But, like in the world of 3D printing, there are those working on systems that print PCBs. This enables researchers to replace the rigid FR4 in favor of flexible, stretchable substrates.

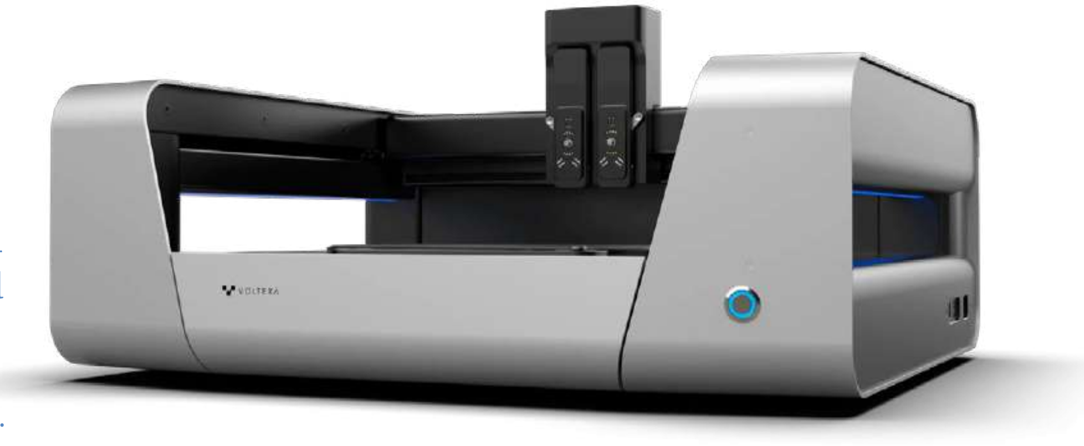


Figure 1: The Voltera NOVA is a serious piece of kit, weighing in at 35 kg and large enough to accommodate substrates up to A4 in size.

Electronics has burrowed its way into almost every aspect of our lives. In fact, finding anything that doesn't rely on electronics in some way is difficult. And while traditional circuit manufacturing materials and processes are optimized for their needs, medical electronics researchers often find them to be sub-optimal, especially when building systems designed to be worn. Human and animal bodies are non-uniform, with skin stretching when in motion that needs to be accommodated by an electronic application. Flexible PCBs have been around for a long time, using FR4 boards bonded to one another via copper tracks on polyimide, also known as Kapton. But while plenty of flex motion is available, the material doesn't stretch.

Flexible electronics is possible; it's just challenging to manufacture and out of the scope of most prototyping teams. For example, thermoplastic polyurethane (TPU) is both flexible and stretchable, and conductors can be applied using a metal deposition process known as sputtering. In a vacuum, the material's surface is bombarded with gaseous ions accelerated by a high voltage. Upon arrival, they form a tight bond with the target material. This method is often used to apply aluminum or alloys such as tungsten-titanium, or aluminum-copper. Otherwise, it's back to screen printing, something you probably last tried at school. Like printing posters or t-shirts, conductive inks are applied to the substrate through a mask applied to a pre-tensioned steel or nylon mesh before heating the ink to cure it.

Additive Approach to Circuit Manufacture

This makes the prototyping process either slow and expensive or slow and messy. However, this is changing thanks to the development of PCB printers, replicating the growth in 3D printers for plastic. Using Direct Ink Writing (DIW), these machines use an additive, rather than a subtractive, approach to apply conductive material where it is needed.

There are two types of rapid prototyping circuit printing approaches on the market. The first uses positive displacement, similar to a medical syringe. The ink is placed inside the barrel with a plastic or rubber plunger head pressed against it. Discounting the complexities of the fluid dynamics of conductive inks, this method is relatively reliable. A specific displacement on the plunger consistently delivers the same amount of ink through the nozzle. Clogging can be an issue, but this is a challenge for any dispenser working with thixotropic materials [1] (fluids whose viscosity changes when sheared). Additionally, for self-filled barrels, trapped air can alter your flow rate. Storage between use can also throw up challenges. The ink may have to be refrigerated or frozen, although some inks can be stored at room temperature.

The other approach is thermal inkjet, where cartridges containing conductive ink print onto the target substrate. As a technology, it can achieve fine-printed features when adequately tuned and can be used with both rigid and flexible substrates. However, compatible inks have relatively low viscosities, making fewer options available. There are also other challenges. Low ink viscosities can lead to beading up or spreading out on some substrates, resulting in poor adhesion unless special coatings are applied. And, because much of the ink is solvent, cured printed features often leave little conductive material behind. Issues with solderability and higher-than-desired track resistance may result. Multi-pass printing can combat this, but may impact minimum feature size due to alignment errors. Nozzle clogging is another frustration. Inks destabilize if incorrectly stored, causing blockages due to the formation of agglomerates. And even if storage recommendations are followed, heat from the thermal inkjetting process will cure some ink in the nozzles.

NOVA — A Modular Dispensing Platform

Tackling the challenges of rapid prototyping of flexible circuits is Voltera, the Canadian scale-up that launched the V-One [2] around seven years ago. That printer primarily targeted rigid substrates, although users were creating flexible circuits too. With NOVA [3], their newest printer launched late last year, they've gone all-in on supporting flexible, stretchable electronics (**Figure 1**).

Leveraging their knowledge of positive displacement printing, they've retained the syringe-style dispensers. However, the hardware is significantly upgraded, with a pressure feedback loop integrated into the printing head. There is also a 12 W heater that holds the ink's temperature at a constant value, ensuring a more consistent dispensing result during application. Incredibly, even air conditioning variations between rooms can result in differences in how the ink will dispense. Bearing in mind the wide variety of substrates and inks they are trying to support, users can quickly trial materials with viscosities ranging from 1,000 to 1,000,000 cP (centipoise). So, while silver-based inks such as the Conductor 3 [4] are readily available and suitable for both rigid and flexible applications, you can also fill a barrel with the same conductive inks used for screen-printed electronics to meet your own needs.

NOVA weighs in at 35 kg, meaning you'll need somewhere permanent to house it. And, at 675×605×345 mm, that's going to be most of a table. At the front, a drawer provides space to store materials and tools. Inside, a foam inlay ensures the print head and surface probe are securely stored alongside the other tools you'll need at hand during operation.

While the unit sports an Ethernet socket, the user guide recommends plugging in the Wi-Fi dongle included and using this to access the printer. Instead of providing software to install on your PC, this device is browser-based. That means, regardless of your chosen operating system and computer manufacturer, you won't have problems operating the printer. Furthermore, in a lab environment, everyone can access the unit as needed. However, if Wi-Fi isn't your thing, there is that provision for a wired Ethernet cable.

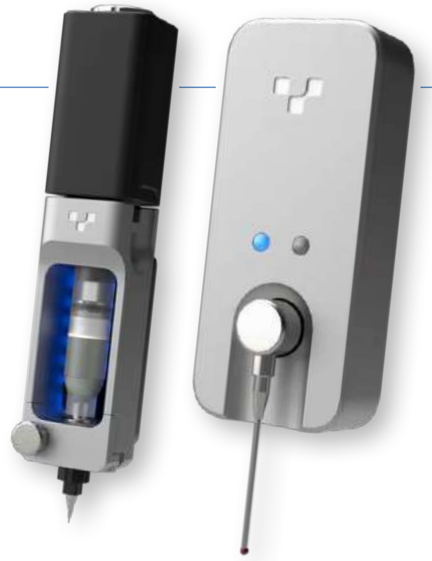
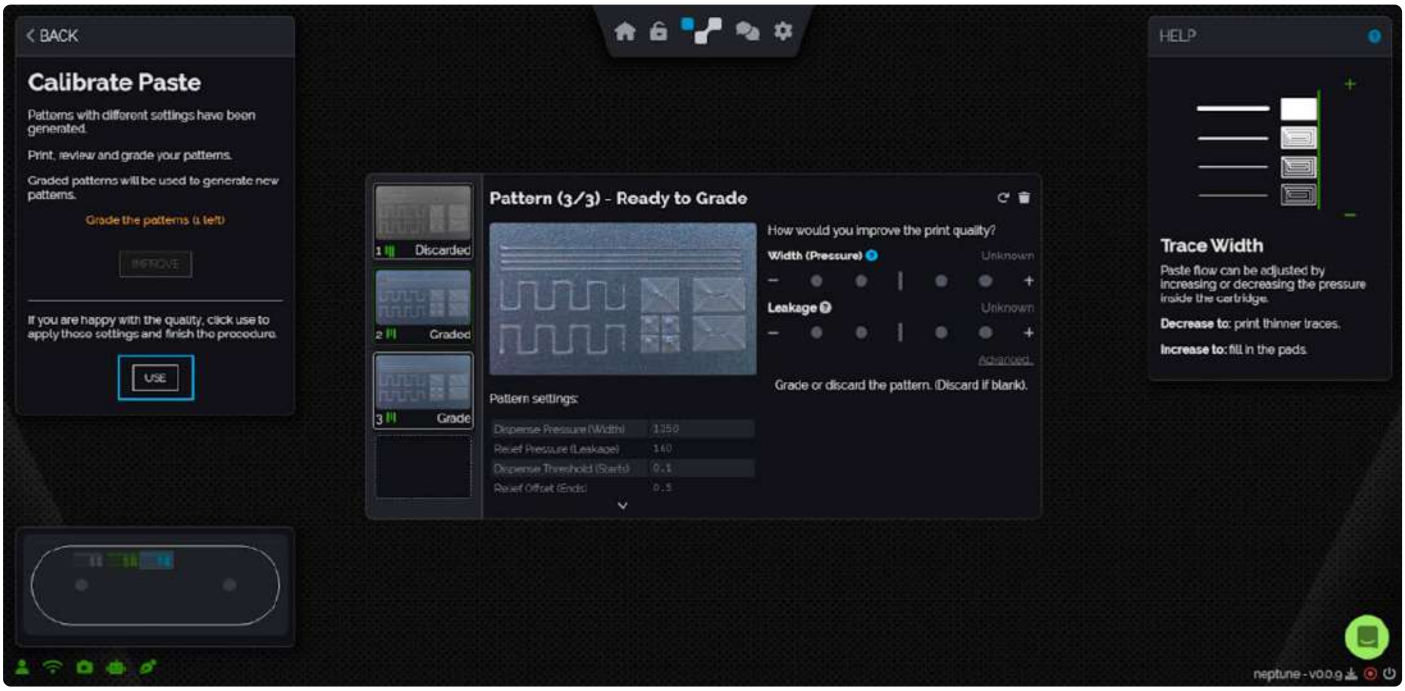


Figure 2: The Smart Dispenser features an integrated heater and AI-enabled flow calibration. The Smart Probe has a ruby tip for precise mapping of substrate surfaces.

Keeping Flexible Substrates in Place

Your flexible substrate is held in place by the porous titanium print bed with an integrated vacuum pump, large enough to handle anything up to A4 size. This stops the material from curling during printing and provides a better fixture solution than clamping. The module port is affixed to an X-Y-Z gantry that simultaneously accommodates two tools. These consist of the Smart Dispenser and the Smart Probe (**Figure 2**). The latter features a ruby-tipped probe that precisely maps the substrate's surface. Tools are easily affixed and detached thanks to the kinematic coupling and locking levers. Integrated LEDs provide the user with the tool's status.

The Smart Dispenser has a pressure sensor used for closed-loop feedback during operation to provide a consistent ink flow. The syringe barrels are standard 5 cc types that can be filled with 2.5 ml of ink, with replacements available from Voltera and its partners. Tracks down to 0.1 mm can be printed, while pads for surface mount devices with a pitch of 0.4 mm are supported with the right combination of nozzle and ink. A selection of nozzles, 100 µm, 150 µm, and 225 µm, are available.



Calibrate Paste

Patterns with different settings have been generated.

Print, review and grade your patterns.

Graded patterns will be used to generate new patterns.

Grade the patterns (1 left)

If you are happy with the quality, click use to apply those settings and finish the procedure.

Pattern (3/3) - Ready to Grade

1 Discarded
2 Graded
3 Grade

How would you improve the print quality?

Width (Pressure) Unknown

Leakage Unknown

Advanced...

Grade or discard the pattern. (Discard if blank).

Pattern settings:

Dispense Pressure (Width)	1350
Relief Pressure (Leakage)	160
Dispense Threshold (Starts)	0.1
Relief Offset (Ends)	0.5

Trace Width

Paste flow can be adjusted by increasing or decreasing the pressure inside the cartridge.

Decrease to: print thinner traces.

Increase to: fill in the pads.

Figure 3: The software for the NOVA is browser-based. Here, we see the AI-enabled calibration process in action.

However, thanks to the use of luer lock syringe barrels, any standard nozzle can be used, ensuring that the optimal combination can be found for the selected ink to achieve the track dimensions you require.

By far, the most significant benefit of the NOVA is its flexibility when testing inks with substrates. Depending on the composition, the ink may only be suitable for rigid substrates. Others may be usable on flexible but not stretchable materials. Polyimide (flexible) and TPU (stretchable) are popular materials, but you'll also need to ensure that the ink wets to the surface. There are also those experimenting with paper, which is porous. If the ink soaks in, there is a risk that conductive pathways fail to form. In such cases, coated papers can be the answer.

Camera-Supported Calibration

Before printing starts, an AI-enabled calibration process is executed (**Figure 3**). Patterns are printed onto a dedicated, removable calibration plate, rather than the target substrate that is integrated in the unit. This avoids the need to clean the target substrate, an alternative calibration target, before printing. NOVA also integrates an 8-megapixel camera that takes photos of the calibration print. After each attempt, the image is shown in the user interface, making it much easier to determine if dispensing has been tuned to your needs.

Curing and Soldering

After printing, the next step is curing, and, for this, a separate oven will be needed. As an example, Conductor 3 ink for flexible circuits has been designed to cure following a profile of 90°C for five minutes, followed by 120°C for 20 minutes, a little more than half the time needed for the previous ink generation. It's important to understand that the curing process is there to evaporate the ink's solvents, leaving the remaining binder polymers to hold the conductive filler particles in place. Because of this, the thickness of your tracks may be reduced during this process.

Some applications only require the conductive tracks to be applied to a flexible or stretchable substrate. But others may need some components added. Depending on the ink used, you may need to burnish the solder pads prior to soldering. Additionally, soldering temperatures often need to be lowered to protect both the electrical tracks and the substrate. As an example, Conductor 3 ink doesn't require burnishing. However, a tin-bismuth-silver (SnBiAg) solder or solder paste will be needed, together with a soldering iron tip or reflow soldering profile temperature of just 180 °C.

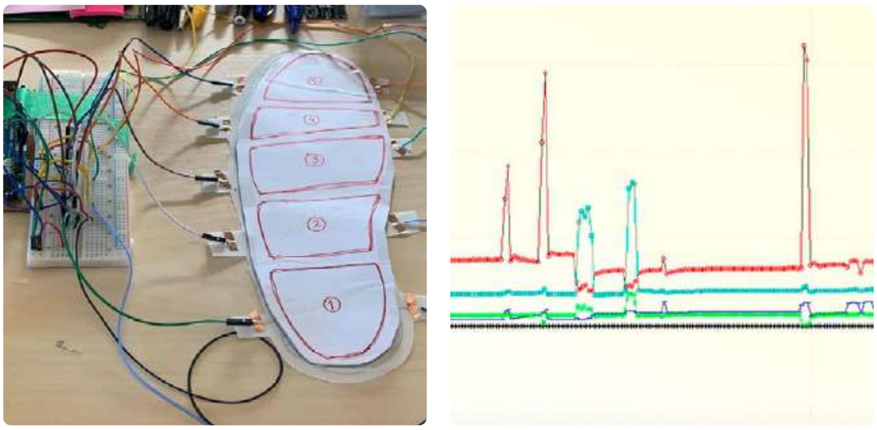


Figure 4: This shoe insole sensor was created using a TPU substrate and stretchable silver ink. Together with an Arduino Mega, pressure across the foot sole can be measured.

Other methods for attaching flexible or stretchable materials to your electronic circuit include compression fixtures, conductive glues, or the creation of a tail for a zero insertion force connector, which may also need the inclusion of a stiffener.

First Attempts With Stretchable PCBs

The Voltera team has been working hard to research the optimal combinations of substrates and inks by trialing wearable applications. One of these was an insole pressure sensor for shoes to help measure weight distribution and monitor posture and gait problems (**Figure 4**). The prototype design uses Celanese Intexar TPU film and Intexar PE874 stretchable silver conductive ink from the same supplier. The printed sensor and a carbon film layer were connected to an Arduino Mega, enabling pressure results to be graphed.

Another project used the same materials to create a heater for integration into clothing (**Figure 5**). The heater is laminated onto the cloth using a T-shirt press commonly used for transferring pictures, while a mechanical compression joint allows power to be applied. Stretchable substrates can also be printed with conductive ink to create strain gauges. This phenomenon was used to good effect in a control glove for a robotic hand (**Figure 6**). Thicker tracks proved to provide a more consistent output, while MG Chemicals 9400 conductive adhesive allowed wires to be connected to the sensors.

Additive PCB printer technology allows researchers and engineering teams to shorten development times and iterate their ideas much faster. Conductive inks can even be used to fix mistakes in a PCB design by simply printing missing or additional tracks, then curing. Wearable electronics have long been a challenge. This is because the truly stretchable electronics for use close to the skin were only manufacturable using expensive metal sputtering techniques or messy screen-printing processes. Now, with the Voltera NOVA, researchers can innovate and create prototypes in-house in numbers large enough for meaningful trials at a fraction of the cost.

230633-01

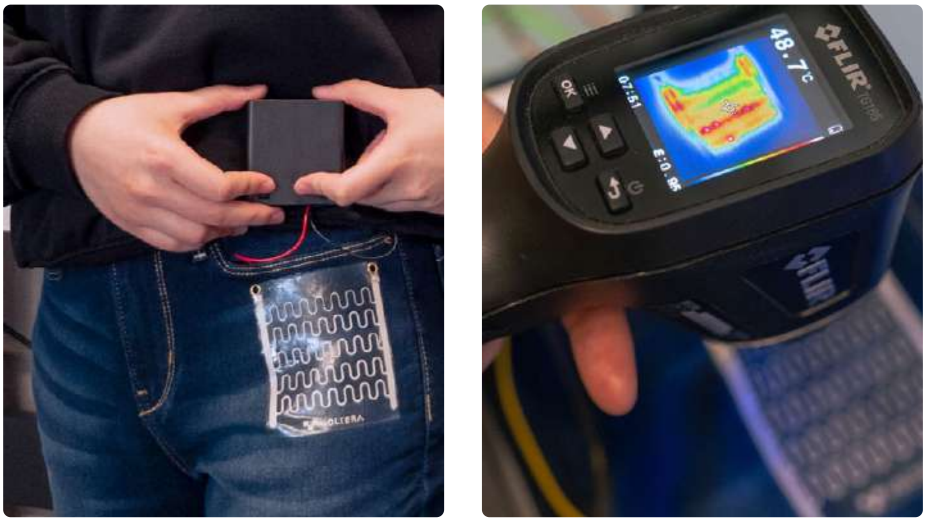


Figure 5: A flexible clothing heater such as this could be used to promote blood flow by dilating blood vessels.

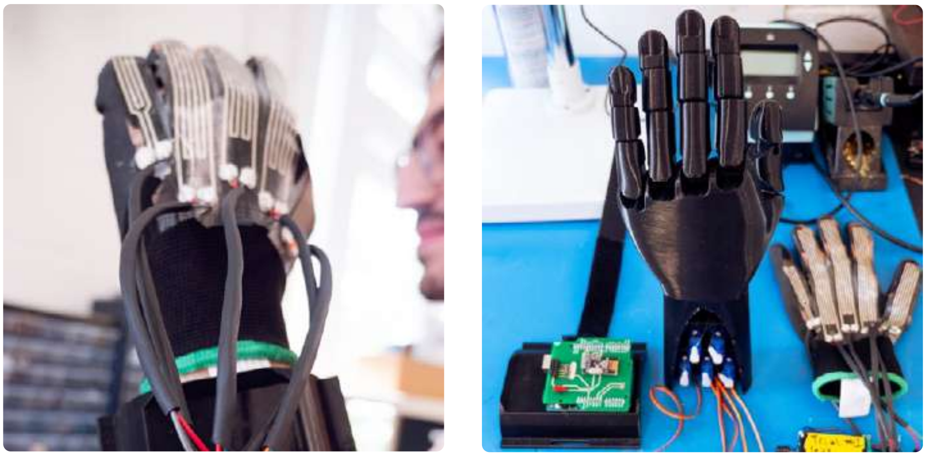


Figure 6: By leveraging changing resistance of conductive ink on a stretchable substrate, strain gauges in the fingers of this glove could control a robotic hand.

WEB LINKS

[1] About thixotropic liquids: https://ptm-mechatronics.com/en/topics/wiki/thixotropic-liquids
[2] V-One — A desktop PCB printer: https://voltera.io/v-one
[3] NOVA — A modular dispensing platform: https://voltera.io/nova
[4] Conductor 3 Ink: https://store.voltera.io/products/conductor-3-ink-cartridge-2ml
[5] Elektor Expert Paper: The buyer's guide to PCB printers: https://elektormagazine.com/elektor-pcb-printer

Galvanic Isolation

Using Phototransistor Optocouplers Successfully

By Eleazar Falco (Würth Elektronik eiSos)

Optocouplers transmit electrical signals optically across an isolation barrier so that two circuit parts can be galvanically isolated from each other. What aspects play an important role in their use and what are their limitations?

An optocoupler is a component that can transmit an electrical signal by optical coupling across two galvanically isolated circuits. Unlike transformers or capacitors, which can only transmit AC signals across the isolation barrier, optocouplers transmit both DC and AC signals equally. This makes them very popular for numerous applications such as isolated power supplies and communication interfaces.

To design a working, robust, and reliable application with optocouplers, it is essential to understand not only the main parameters and parasitic elements of the device, but also variations and dependence on other factors such as temperature and operating point. In this context, the most important factors are the current transfer ratio (CTR) and the parasitic output capacitance of the optocoupler, which limit the operating frequency range and switching power.

This article covers the basics of the operation of the WL-OCPT [1] family of phototransistor output optocouplers from Würth Elektronik, including its parameter characterization for a given operating condition and important design considerations.

Structure and Operation

A phototransistor optocoupler consists of a gallium arsenide infrared light emitter (GaAs IR LED) and a light detector (phototransistor), both optically coupled and typically encapsulated in a 4-pin package. This assembly comes in a variety of mechanical dimensions and solder variations to meet the varying size and isolation requirements of target applications. The WL-OCPT series is based on a coplanar dual-cast structure with a copper alloy metal frame and a clear epoxy dome serving as the optical medium between the IR LED and the phototransistor (**Figure 1**). Such a structure provides a mechanically robust isolation barrier at very high voltage levels (e.g., 5 kV) while maintaining a compact size.

The principle of operation of the optocoupler: A conversion of current to light takes place in the emitter, as the IR LED emits infrared radiation (i.e., photons) with an intensity proportional to the current flowing through it. These photons pass through the optical medium and, after reflection from the inner surface of the dome, reach the light-sensitive base-collector region of the phototransistor. Here, a reconversion of the light energy into current takes place: The phototransistor's valence electrons "absorb" the photon energy and "jump" into the conduction band, generating a current between the collector and emitter — similar to a conventional NPN bipolar transistor. The ratio between the collector current of the phototransistor (I_C) and the current of the IR LED (I_F) is the optocoupler's most important parameter: the direct current transfer ratio (CTR). The electrical symbol of a phototransistor optocoupler is shown in **Figure 2**.

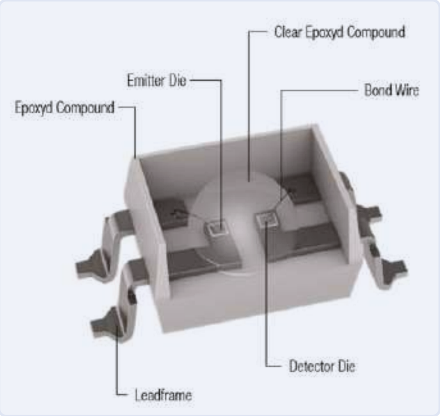


Figure 1: Typical structure of a WL-OCPT-family optocoupler.

Direct Current Transfer Ratio (CTR)

CTR is the current gain parameter of the optocoupler, expressed as the ratio of collector current (I_C) to LED current (I_F) in **Equation 1**:

$$CTR = \frac{I_C}{I_F}$$

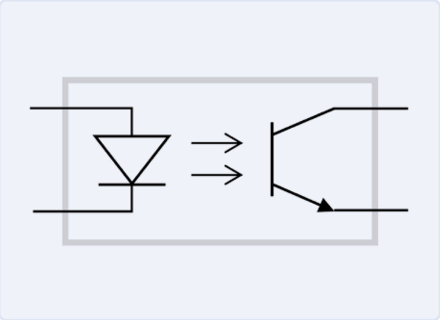


Figure 2: Phototransistor optocoupler electrical schematic symbol.

Table 1: Example of optocoupler binning based on CTR (WL-OCPT series 816/817).

CTR-Binning: WL-OCPT Series 816 and 817
Test conditions I_F = 5 mA, V_{CE} = 5 V, T = 25 °C

Binning	Min.	Max.	Unit
none	50	600	%
A	80	160	%
B	130	260	%
C	200	400	%
D	300	600	%

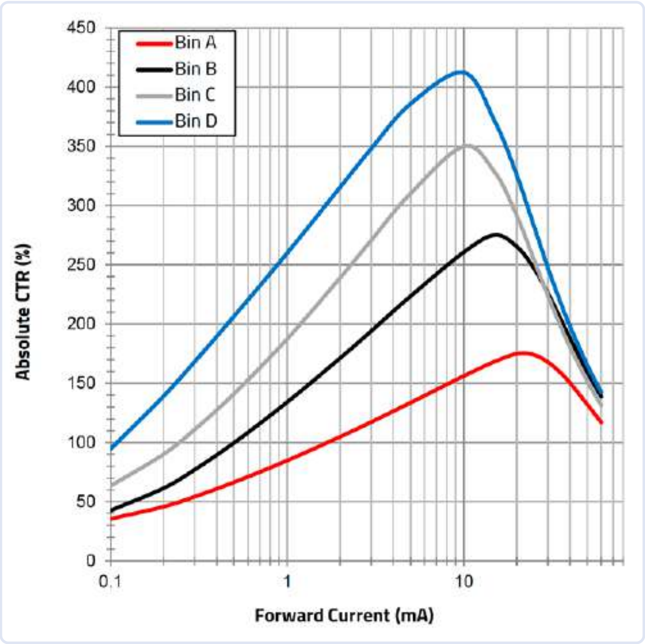


Figure 3: Absolute CTR in relation to I_F (at V_{CE} = 5 V, T = 25 °C; WL-OCPT 817).

Note that Equation 1 is valid when the phototransistor is biased within its active operating region. By analogy, the CTR corresponds to the static h_{FE} or β-parameter (beta) of a bipolar transistor, while the LED current would correspond to its base current. In functional terms, therefore, the optocoupler can be viewed as an "isolated base" NPN bipolar transistor. It should also be noted that the CTR is usually expressed as a percentage (%) (**Equation 2**):

$$CTR(\%) = \frac{I_C}{I_F} \cdot 100$$

Among the major challenges in optocoupler design are the large tolerances and variations in its CTR value caused by various factors.

CTR Production Tolerance and Binning

The limitations faced by semiconductor manufacturing processes make it difficult to produce devices with identical characteristics and parameter values. IR LEDs are grown on GaAs-substrate wafers, where achieving completely uniform semiconductor doping and layer thickness is challenging. This results in different current-to-light conversion efficiencies for each LED device on the wafer (i.e., the light energy emitted by each LED at a given current level is different). Similar constraints apply to the phototransistor, resulting in devices with different light sensitivity (i.e., they produce different collector currents for identical light energy received). These two factors directly affect CTR tolerance in production, but they are not the only ones. For example, variations in the transmittance and reflectance parameters of the clear epoxy dome that optically connects the LED and phototransistor also contribute to the increase in CTR production tolerance of the device.

Since too much tolerance is impractical for most designs, the CTR of each optocoupler is measured in production at a specified DC operating point and the device is classified based on the measured value as part of a binning process. Each binning guarantees a minimum and maximum CTR value under the specified test conditions and is indicated with an additional letter, as shown in **Table 1** for the WL-OCPT 816/817 series. While this reduces the initial tolerance range to be considered in the design, the CTR also depends on operating conditions such as DC bias voltage and temperature, and these variations must also be considered.

LED Current-Induced Variation of the CTR

The CTR range within a binning is valid only for a given LED current (I_F) and collector-emitter voltage (V_{CE}). For the WL-OCPT series, these are 5 mA and 5 V, respectively. For a different LED current, the resulting CTR range would be different, as shown in **Figure 3**: Here, the absolute CTR is plotted as a function of the LED current of a pattern from each binning (for V_{CE} = 5 V). It can be seen how the CTR increases with LED current up to a certain point, above which the behavior reverses, i.e., a further increase in LED current causes a decrease in CTR. This is due to the nonlinear efficiency of the current-to-light conversion of the LED combined with the variation in phototransistor gain. For I_F = 5 mA, the CTR of each device is within the range of the corresponding binning, as expected.

When the phototransistor enters the saturation region, the CTR value and the characteristic curve also change, as shown in **Figure 4** for V_{CE} = 0.4 V: The CTR is lower, and the characteristic peaks at a lower LED current.

Note that the characteristic curves in Figures 3 and 4 correspond to only a single sample part from each binning. Characteristic curves for relative CTR are also given on the WL-OCPT series data sheet, where the CTR is normalized to the value measured for I_F = 5 mA. For an LED current of less than 10 mA and with the phototransistor operating in the active region (V_{CE} = 5 V), all normalized characteristics are very close, regardless of binning [2]. Since the relative CTR characteristic of a binning in this range practically does not change with the absolute CTR value, the absolute CTR of any device within the binning can be estimated using **Equation 3**:

$$CTR_{\mathrm{IF}} \approx CTR_{\mathrm{rel(IF)}} \cdot CTR_{\mathrm{5mA}}$$

It should be noted that, above 10 mA, the relative characteristics show a greater deviation from the absolute CTR value, so

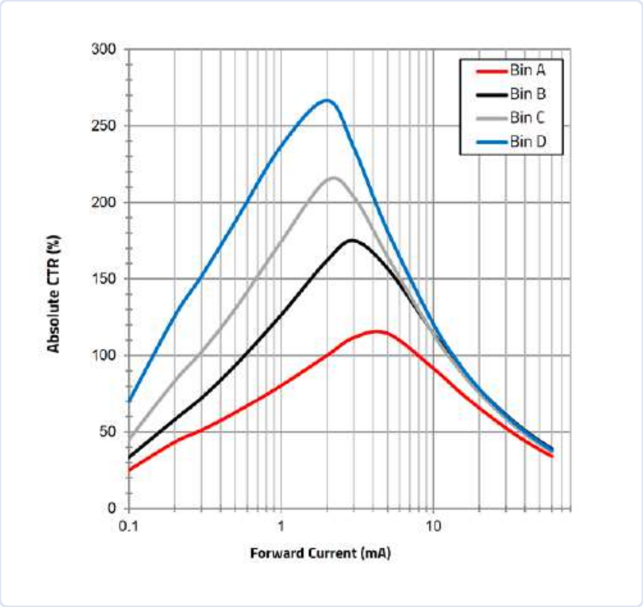


Figure 4: Absolute CTR in relation to I_F (at V_{CE} = 0.4 V, T = 25 °C).

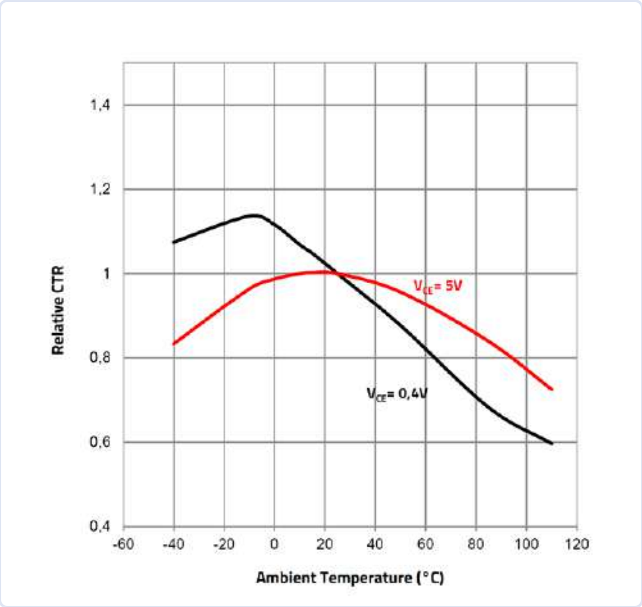


Figure 5: CTR in relation to temperature (at I_F = 5 mA).

Equation 3 is less accurate in that range. When the phototransistor is operating in saturation, the curve characteristics are inverted to some extent, and Equation 3 is accurate only above about 5 mA.

Temperature-Induced Variation of the CTR

The operating temperature of the device affects both the efficiency of the LED emission and the light-current gain of the phototransistor, and in this respect affects the CTR of the optocoupler in a non-linear manner — the characteristic curves for the relative CTR shown in **Figure 5** demonstrate this. Both characteristic curves are valid for all binnings and for an LED current below about 5 mA. Note that the temperature of the operating environment is considered here, to which the junction temperature of the device is directly proportional.

CTR Measurement

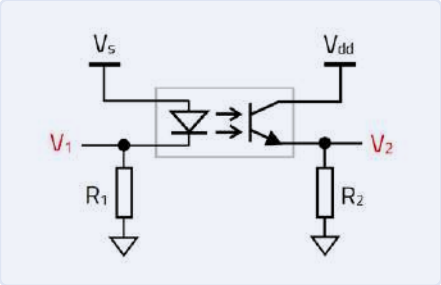


Figure 6: Recommended setup for CTR measurement.

Measuring the CTR for the specific DC bias conditions of the application is straightforward if the currents are measured based on Equation 1. This is also true for the recommended setup in **Figure 6**, where only voltage sensors are used. A typical collector configuration is used here. Both output resistor R_2 and bias voltage V_{dd} are set as in the final application circuit, where the optocoupler is used, and the same value is chosen for the LED resistor R_1 as for R_2. Then, an increase in the DC input voltage (V_s) takes place until the collector-emitter voltage (V_{CE}) reaches the target value. The CTR for this DC bias condition is given by **Equation 4**:

$$CTR = \frac{V_2}{V_1}$$

Other Parameters in Detail

The application note ANO007 [2] offers much more detailed information about optocouplers, which would go beyond the scope of this document. Specifically, there are:

- an example for estimating the CTR range as well as AC/small-signal CTR
- consideration of frequency response and bandwidth with measured values
- switching times and switching time fluctuations
- optocoupler DC bias voltage
- SPICE modeling

230546-01

About the Author

Eleazar Falco holds a degree in Electrical Engineering from the University of Elche in Spain. In 2014, he joined Dyson in the United Kingdom, where he worked in electronics hardware development of home appliances, focusing on offline power supply and motor control. Since 2018, he is an Applications Engineer for switching power supplies at Würth Elektronik in Germany.

WEB LINKS

[1] WL-OCPT Optocoupler Phototransistor from Würth Elektronik: https://tinyurl.com/weoptocouplers

[2] Falco, Eleazar: Understanding Phototransistor Optocouplers. Application Note ANO007 from Würth Elektronik: http://www.we-online.com/ANO007

The Complex Solution or the Anybus Solution?

Embedded Industrial Ethernet in 2 Days Rather Than Many Months

Contributed by HMS Networks

Industrial networking is taking an increasingly important role in controlling machines, factories, and warehouses. What once began with the simple management of remote inputs and outputs, has grown into hugely scalable networks that can provide even the most complex control of entire factories with thousands of nodes. Often, most of the nodes are ready-made products such as drives, I/O stations or sensors. But what happens if you want to connect an industrial network with the new device you are developing?

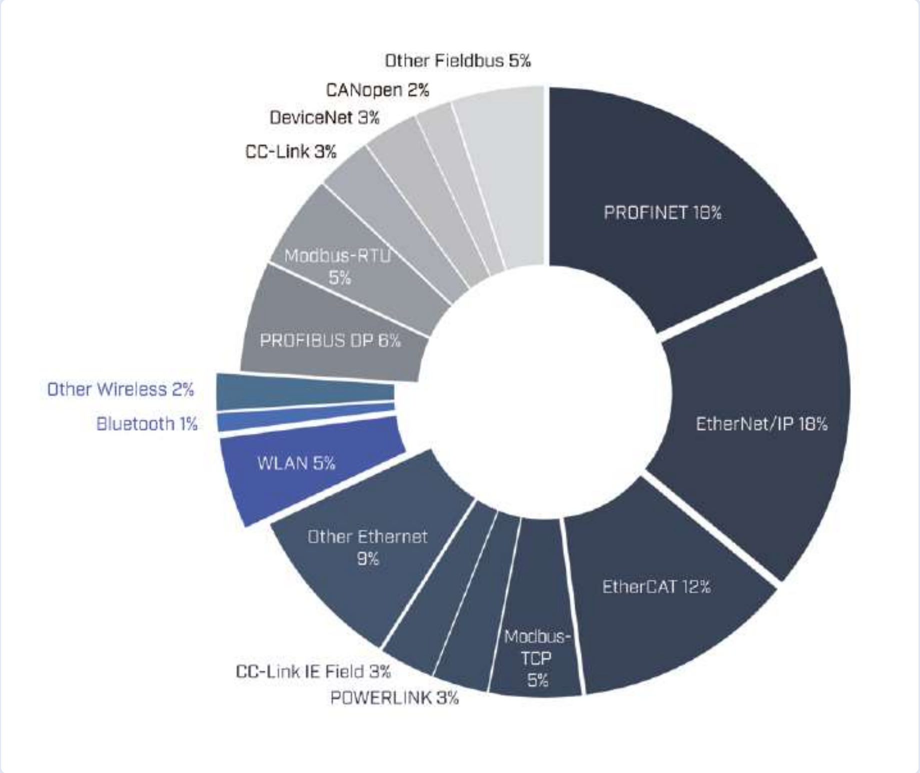


Figure 1: The study done in 2023 by HMS Networks shows a total growth in the number of installed nodes of 7%.

Industrial Networks

It is important to realise that the term "industrial networks" encompasses a broad range of industrial protocols. There are fieldbuses such as Profibus, Modbus RTU or CANopen, which are mainly used in environments that involve long distances. This could be, for example, large fuel tanks where sensors and other nodes can be installed hundreds of metres apart.

Other networks are Ethernet-based, such as Profinet, EtherNet/IP or EtherCAT (**Figure 1**). They offer high-quality, possibly even real-time properties where accuracy plays a major role. The fastest protocols can achieve cycle times of 50 µs and a deviation of < 1 µs. This allows for engines to be synchronised very accurately. In addition to real-time aspects, cloud connections are being used increasingly within the industry. Various protocols are available for this as well. Currently, about 25% of the market is using fieldbuses and another 70% use variants of Ethernet. The other 5% is wireless.

However, the protocols to be used are not entirely a free choice. They are often linked to a particular producer of PLCs, and they, in turn, have their own typical share in the global market. Therefore, in Europe, protocols such as Profibus, Profinet or EtherCAT are very popular, whereas in North America, EtherNet/IP is used the most.

Let's say that you have developed a bar code scanner to be used in logistics applications. The system integrators who will use the scanner in their systems will determine which network they use, which will often also be in consultation with the end customer. The choice of protocol will ultimately depend on the customer, and you have no control over where they are located or which PLC they might be using. This is why your scanner must be able to support several protocols.

Designing Industrial Network Interfaces

What is involved in designing a product that includes multiple industrial network interfaces? Obviously, there is different hardware and software required. Fieldbuses all use different connectors. CAN bus has its own physical driver (ISO11898), Profibus (DP) uses RS485, etc. but even the Ethernet variants have differences. Most are 100 Mb/s but, once you go behind the RJ45 and the PHY, you start to see the variations. They may use MACs that have been designed differently, as well as buffers, at times real-time clocks (IEEE 1588), or their very own Ethernet controller (EtherCAT). CC link IE, used often in Japan, is a gigabit connection. Additionally, some industries use fibre optic connections by default rather than the traditional copper ones.

The software side is even more complex. For each protocol, high-speed data stream processors and stacks are available on the market. But these are not simply libraries that you can link to in your embedded software. They demand thorough knowledge of how a specific protocol works and how you implement parts of it. Moreover, each stack will come with a separate driver that must be adjusted to the target hardware. Stacks are often from different suppliers, each with their own API, so the application software must, in turn, be adjusted to each protocol used. All protocols implemented must be certified so that they meet all the technical network requirements. And of course, protocols continue to be further developed, which means that regular updates are required.

As already mentioned, the processing of protocols takes place on three layers: the hardware layer (with the MAC), the high-speed data stream processing, and the stack that processes the protocol. These are often separate parts in the design, for example, an ASIC for the first layer and various layers of software in the microcontroller for the other two. It is not surprising if this does not always work together seamlessly. Designing a multiple-network interface is therefore also a complex and time-consuming occupation that can take months.

So Is There a Simpler Way?

Since 1995, the team working under the Anybus brand of Swedish company HMS Networks, has been developing complex network interfaces that take on the responsibility for all tasks concerning communication. The Anybus CompactCom–40 is the fourth generation in a continuing development to be able to support the newest and fastest networks, such as the up-and-coming gigabit networks or TSN protocols.

Anybus CompactCom combines all layers — hardware, data processing and stack(s) — together in its NP40 chip, with the first two parts integrated in an FPGA and the stack running on the built-in microcontroller, both of which are optimised for communication. It works with one and the same API for all protocols, which is unique.

The strength of the Anybus CompactCom is that it processes the information rather than passing it on. In addition to the well-known local and global variants, a new type of variable is created: the ***network variable***. This is stored in the CompactCom and is the mechanism whereby data is exchanged with your application.

If you replace a CompactCom module with a version for a different protocol, the same variables are still shared and, as a result, nothing changes for your application. With just one single development, you have access to all known fieldbus and Ethernet protocols.

The Anybus CompactCom is available in various form factors. There is a Module (Anybus M40) that can be easily installed, sliding into a connector on a PCB (**Figure 2**). The module comes complete with the correct network connector; e.g. RJ45 for the Ethernet

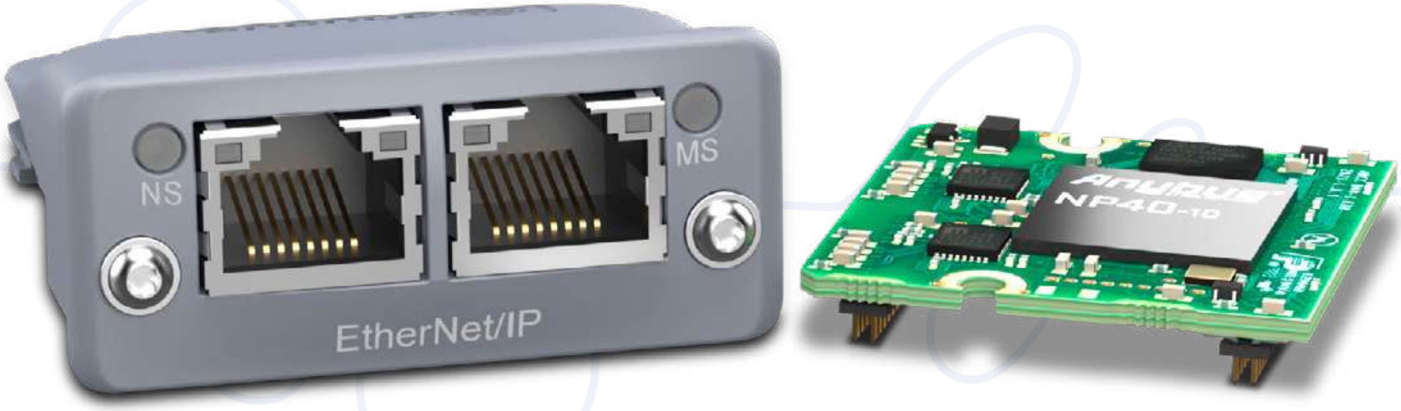


Figure 2: Module M40, complete with connectors and the Brick B40 — a plug-in solution, ideal for devices with limited space.

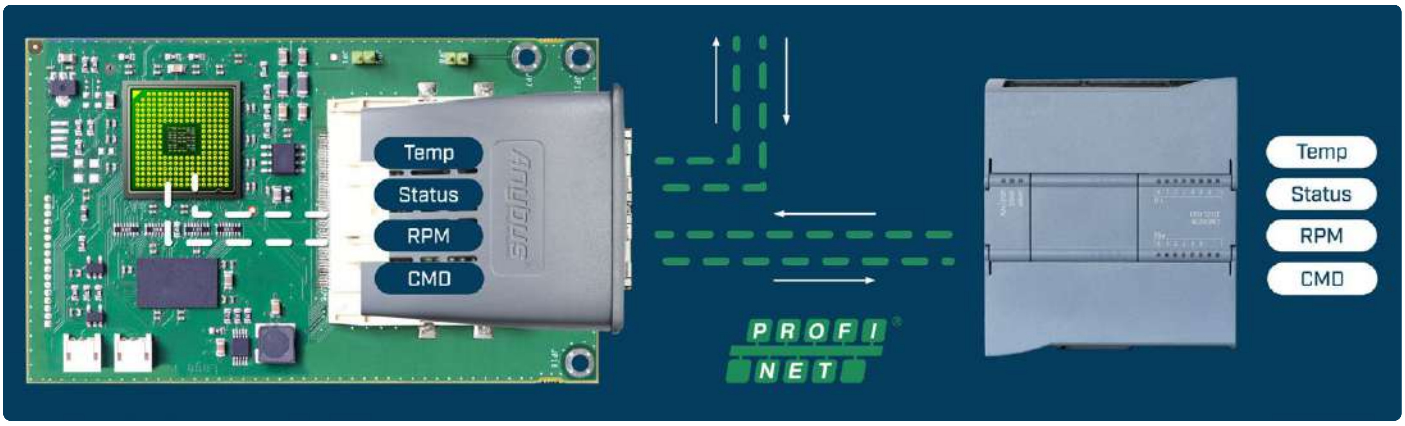


Figure 3: Anybus CompactCom exchanges information with the PLC and other nodes.

variants and SUB–D9 for Profibus. The benefit of this is that the module can also be installed later, possibly by end users at commissioning time.

There is also the B40 Brick (a plug-in board), which can easily be inserted onto two headers on a PCB. In contrast to the M40 Module, the B40 comes without a network connector, which allows any required connector to be used. Both versions use the same NP40 chip and are identical technologically.

The CompactCom can be integrated with your application via parallel (8/16 bits) or SPI data exchange with the microcontroller. In addition, a simple reset is required, connected to an I/O of the microcontroller. A small software interface written in the C programming language is available for integration at no extra charge.

Shorter Implementation Time Has Many Benefits

This integrated total solution by HMS Networks considerably shortens the time needed for implementation. Once the hardware is ready, the software will be operational in 2 to 3 days, and the first information exchanged with the PLC or master.

The Anybus CompactCom is pre-certified and is guaranteed to work according to the protocol standard (**Figure 3**). The pre-certification of CompactCom speeds up the entire certification process considerably because the tests can often be performed in a shorter time.

Simple Because of Its Many Features

The Anybus CompactCom offers many extras. All Ethernet variants are equipped with a web server, allowing the final application to be interfaced with via a standard browser. In addition, it has a complete file structure accessible via FTP which can be used, for example, to download host firmware. It also supports email clients, WebSockets, etc.

For more information on the Anybus CompactCom solution, visit the website [1]. ◀

230603-01

WEB LINK

[1] Anybus by HMS Networks: https://anybus.com

WÜRTH ELEKTRONIK
MORE THAN
YOU EXPECT

WE are here for you!
Join our free webinars on:
www.we-online.com/webinars

Adrastea-I is a Cellular Module with High Performance, Ultra-Low Power Consumption, Multi-Band LTE-M and NB-IoT Module.

Despite its compact size, the module has integrated GNSS, integrated ARM Cortex M4 and 1MB Flash reserved for user application development. The module is based on the high-performance Sony Altair ALT1250 chipset. The Adrastea-I module, certified by Deutsche Telekom, enables rapid integration into end products without additional industry-specific certification (GCF) or operator approval. Provided that a Deutsche Telekom IoT connectivity (SIM card) is used. For all other operators the module offers the industry-specific certification (GCF) already.

www.we-online.com/gocellular

- Small form factor
- Security and encryption
- Long range/wordwide coverage
- Multi-band support

#GOCELLULAR

Your Essential DFM Checklist

How to Start Designing for Manufacture

By Mark Patrick, Mouser Electronics

The race to bring new electronics-based products to market is never-ending. Design and engineering teams find themselves caught up in a fast-paced environment that demands design prototypes within weeks. Unfortunately, the rate at which designs become products may mask challenges that come to light during production planning. As a result, product launch timelines slip while engineering design issues are solved.

Almost all electronics design today faces multiple design constraints; the most common are the product's feature set, available space envelope, and the power consumption profile.

Taking a finished design into production introduces additional challenges for the procurement team, such as:

- Component availability and lead times
- The risk of counterfeit parts
- Obsolescence and other lifecycle issues

Together, the design and purchasing teams face many challenges, but introducing a design-for-manufacture (DFM) approach provides a framework that highlights potential issues before production commences. Using a DFM checklist enables manufacturers to remove production challenges and streamline the whole design, supply chain, and production process.

Identifying Your Challenges

For the design team, commencing a new development can be intriguing. The starting point is often a marketing specification that defines a new product's features, functions, and capabilities based on market expectations. It forms a starting point for the engineering teams to scope out an initial design concept and identify a myriad of other design queries. Typically, functional disciplines dictate the arrangement of engineering teams, including mechanical, hardware, software, and, increasingly, user experience. Each subteam reviews the specification, highlighting potential concerns or the need for further information.

Many of the queries may impact more than one engineering function or highlight potential design constraints. For example, the mechanical team may query:

- If the product worn on a user's wrist — what is the maximum optimum size for a range of different users?
- Will ingress protection be required for outdoor use?

The electronics hardware team may query:

- How long should the battery life be?
- What type of display suits the specification requirement? Readable in bright daylight? Touch controls?
- What are the maximum dimensions available for the PCB? Space is like a dynamic jigsaw puzzle accommodating PCB, display, battery!

Commencing a new design is a complex puzzle that needs close cooperation and communication between supply chain, production, marketing, and engineering functions. The mechanical engineering team's component or design decision may impact other business functions — not just their engineering colleagues. For example, mounting a display a particular way may result in an awkward user experience and create touchscreen sensitivity challenges.

Establishing Your Design-for-Manufacture Strategy

Your design-for-manufacture (DFM) strategy should establish a clear framework that encompasses every design decision and the potential impact on:

- component sourcing and procurement
- material handling
- bill of material costs
- production processes

- final assembly
- quality assurance and testing
- through lifecycle support

The goal of any DFM process is to facilitate ease of product development and production, contain costs, and smooth market introduction. A successful DFM framework should become a single, holistic process that involves communication and collaboration with all relevant functional groups. Design issues, sourcing challenges, and production constraints can be solved before encountering significant holdups.

Clear, ongoing, and timely communication is paramount.

Essential DFM Checklist Topics

Many potential topics should be on your DFM Checklist. Below, you'll find some of the most common encountered by electronic product manufacturers, grouped by business function.

Engineering Considerations

Discrete design or module decisions:
This is a frequent challenge facing many engineering teams. Today, many popular circuit functions have become incorporated inside commodity ICs. Wireless transceivers, for example, can be designed from scratch by specialist RF engineers or purchased as a single, compact, and regulatory pre-approved module. Wireless transceivers are also available integrated with a microcontroller as a wireless system-on-chip (SoC). DC/DC converters are another frequent example. Considerations include:

- A module benefits from being a single bill of material (BOM) item and can save an immense amount of non-recoverable engineering expenses (NRE). However, will it deliver the precise parameters the application requires?
- A discrete approach can match the exact design criteria, but at the expense of engineering time and increasing the BOM. Does the design warrant a discrete approach?
- What is the BOM cost of the module compared to handfuls of discrete components?
- Also, don't forget that BOM cost isn't the only consideration. Sourcing, managing, and stocking inventory can be more costly than the components. What additional cost and inconvenience would that add to your operation?
- Modules often benefit from being space-optimized, occupying a minimal PCB footprint. Do you need to save space in your design?
- How close are the discrete circuit parameters to a suitable module? Could the design be adapted to utilize a commodity module?

Safety and type-approval compliance:

- What are the standards to which your design needs to conform? Common examples include user isolation, safety, EMI/EMC, RF type approval, functional safety.
- Have time and cost estimates been calculated to achieve compliance? Is it a specialist skill that requires an external consultancy?
- Might a module approach remove the need for compliance testing and significantly simplify the complexity of production testing tasks and improve throughput?

Criteria for Selecting Components:
On first inspection, this topic might appear relatively straightforward. However, it can have far-reaching consequences when sourcing components, managing the BOM and searching for second-source alternatives. Engineers may specify a particular part based on circuit need without considering alternatives, different suppliers, or procurement options.

- Passive components — typically specified with a given tolerance value within a specific series range. Are high-tolerance components essential? The construction method, ceramic, polymer, etc., also affect the price of capacitors. Selecting all parts with the same construction method and lower tolerance value reduces BOM cost and complexity and eases inventory management challenges.
- Is the procurement team already sourcing a particular component? What has availability been like, and has a suitable second source been identified, or potentially already ordered?
- Many ICs are available in different package formats. Has the most space-efficient part been specified?
- Is the IC package format a special-order item? Would a more popular package format still fulfil the design requirements?
- Is the component popular and readily available? Has the manufacturer highlighted an end-of-life date?

Supply Chain

Component Availability:

- Has Engineering considered alternative components that might simplify sourcing, BOM cost, and ease production testing?
- How much material stock is typically ordered, and what does that represent in terms of end-product quantity?
- Are your component suppliers in regular communication with the procurement team? Do they keep you informed of potential delays, shortages, or allocations?
- Has a list of critical components that might encounter supply difficulties been identified? Is that list regularly reviewed and checked? Are their second source alternatives also monitored?

Other procurement risks:

- Do your suppliers provide component traceability as standard? Counterfeit parts appear across the electronics supply chain, so constant monitoring is required. Work only with reputable component suppliers and insist on traceability.
- Do your suppliers also monitor for repackaged parts? Like counterfeit parts, once in a product, they result in expensive downtime, lost production, and replacement material costs.
- Are all components received checked for all relevant documentation and certificates? Certification can become complex, yet mandatory for some product applications.

Production:

Reduce component handling:

- How many times are you handling components received from suppliers? Low-cost items such as passive

components occupy little space, and the time and labour cost of moving them from one location to another are costly. Are they delivered to where production takes place?
- Are component traceability and conformance tests conducted close to production?
- Can your component suppliers assist you in reducing the handling involved? Do they provide a "kiting" service?

Simplify the assembly process:

- How many steps and processes are involved in end-product assembly? Could they be streamlined, or some tasks combined?
- Would the use of modules ease the burden of production testing or inspection?
- Are subcontractors used for sub-assembly tasks? Is it practical and efficient? Could they do more, or is it specialized?
-

In addition to factors that impact specific business functions, the global nature of the electronics component supply chain introduces many other geopolitical, sustainability, and environmental considerations.

Global Considerations

- Do you or your supplier track the supply of raw materials to component manufacturers? Do any of your components rely on rare-earth minerals or compounds in short supply?
- How exposed is your material supply chain to regional or national trade disputes or geopolitical tensions? Have you subscribed to the news feeds of any relevant suppliers?
- What are your environmental, social, and governance (ESG) policies? Do you or an external organization audit your suppliers and monitor their sustainability performance?
- Do you publish a corporate social responsibility policy? How does it align with your product design and engineering methods?
- Are any of your suppliers located in regions that have experienced — or are prone to — environmental disasters? Is there a contingency should it reoccur and cause supply chain disruption?

This DFM checklist covers just some of the steps involved in formulating and deploying a DFM strategy. A regular and engaging communication approach with all internal functions, component suppliers, and sub-assembly manufacturers enables early problem identification.

Mouser — Supporting Your DFM Strategy with Online Tools and Services

Mouser provides free tools such as Forte, a BOM management and inventory application, to assist your component-sourcing needs. It can provide the basis for managing your product throughout its lifecycle. Forte's time-saving features include:

- A part match confidence indicator to validate the correct item is selected from the BOM
- Multi-quantity price breaks are highlighted with changing BOM quantities
- BOM import capability using any popular spreadsheet file, or create natively within Forte
- Notification of potential stock and obsolescence warnings
- A BOM export capability.

A complete list of Forte's key features, together with a quick-start guide, is available at [1].

During the hardware and mechanical design phases, comprehensive product, information such as datasheets, 3D CAD models, and PCB footprints, are freely available for download from [2].

In addition, a collection of useful tools such as Conversion Calculators, Project Share, APIs, Order Automation, Services, and many others can be found at [3]. Go ahead and explore.

230576-01

About the Author

As Mouser Electronics' Technical Marketing Manager for EMEA, Mark Patrick is responsible for the creation and circulation of technical content within the region — content that is key to Mouser's strategy to support, inform and inspire its engineering audience. Prior to leading the Technical Marketing team, Patrick was part of the EMEA Supplier Marketing team and played a vital role in establishing and developing relationships with key manufacturing partners. In addition to a variety of technical and marketing positions, Patrick's previous roles include eight years at Texas Instruments in Applications Support and Technical Sales. A "hands-on" engineer at heart, with a passion for vintage synthesizers and motorcycles, he thinks nothing of carrying out repairs on either. Patrick holds a first-class Honours Degree in Electronics Engineering from Coventry University.

WEB LINKS

[1] Forte's key features and quick-start guide: https://eu.mouser.com/bomtool
[2] PCB footprints and 3D models: https://eu.mouser.com/electronic-cad-symbols-models
[3] Mouser online services and tools: https://eu.mouser.com/servicesandtools

Join the Elektor Community

Take out a **GOLD** membership!

- ✓ The Elektor web archive from 1974!
- ✓ 8x Elektor Magazine (print)
- ✓ 8x Elektor Magazine (digital)
- ✓ 10% discount in our web shop and exclusive offers
- ✓ Access to more than 5000 Gerber files

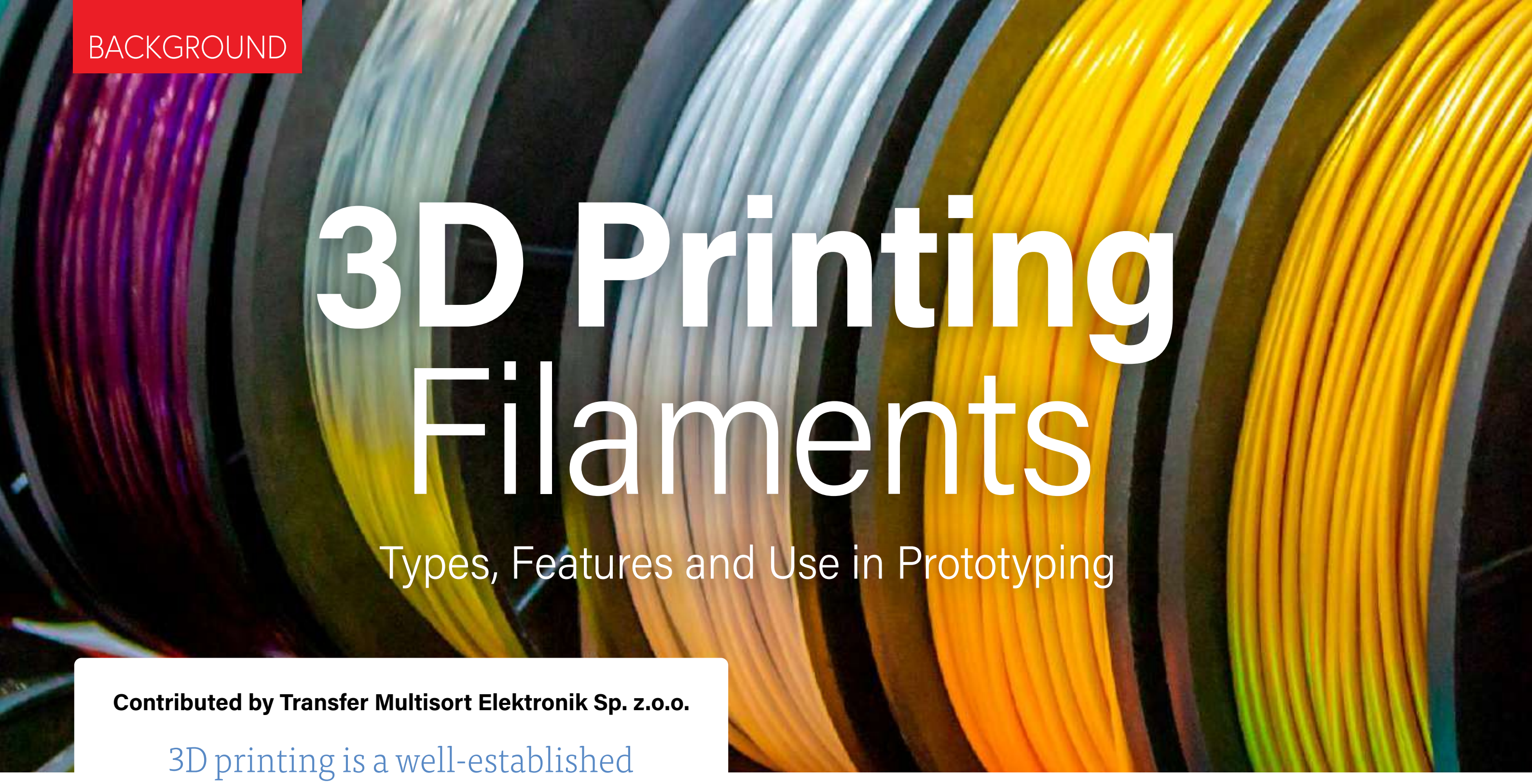
Also available

The Digital **GREEN** membership!

- ✓ The Elektor web archive from 1974!
- ✓ 8x Elektor Magazine (digital)
- ✓ 10% discount in our web shop and exclusive offers
- ✓ Access to more than 5000 Gerber files

www.elektormagazine.com/Member

elektor design > share > earn



Source: © mari1408/Adobe Stock

3D Printing Filaments

Types, Features and Use in Prototyping

Contributed by Transfer Multisort Elektronik Sp. z.o.o.

3D printing is a well-established solution, although many people still perceive it as something innovative and fresh. The ability to print entire objects from scratch has vastly changed not only the way we think of prototyping, but also the whole process of executing such projects. Just like a traditional printer needs a cartridge or toner to operate properly, a 3D printer cannot operate without a filament, which can have different properties that are worth considering when selecting the material.

What Are 3D Printing Filaments?

3D printing filaments [1] are thermoplastics used for printing three-dimensional objects through the process of building up the melted material by the 3D printer. They are available in different types and colors, made of e.g., ABS (acrylonitrile butadiene styrene), PLA (polylactide), PET (polyethylene terephthalate), nylon, and many others. They are supplied in the form of coils that are wound on rolls and then inserted into a 3D printer. The thermoplastic is unreeled and fed through the nozzle, which is heated to a temperature at which it melts. The nozzle, controlling the flow of the thermoplastic, moves around as per the programmed instructions (for example, using CAM). This resembles stereolithography (SLA), which also involves applying subsequent laser-cured layers. 3D printing is used in more and more fields, generating a growing list of filaments available on the market. Some are made from natural raw materials, other ones are synthesized entirely from plastics. All the types have one thing in common: They take on the form of thin, dozens or even hundreds of meters long, fibers that are wound onto spools of different sizes. The spools are convenient to mount in and unmount from the printer and, thanks to their side covers, protect the fibers from slipping off.

Filaments from recognized suppliers are available in two basic fiber diameters: 1.75 mm and 2.85 mm. They are usually made neatly and are free of any random coarsening and irregularities. This is crucial for printing quality, as every instance of coarsening means a surplus of material in the nozzle and slightly different material behavior when applied. It's worth remembering here that the thickness of a typical single layer is about 80% of the nozzle diameter, usually slightly less than 0.5 mm (e.g., for a Ø 0.4 mm nozzle, the obtained layer is 0.32 mm).

Features of the Most Important Filament Types

There are many types of filaments and thus many types of materials that they are made of. However, there is one group that could certainly be called the most popular, which is worth knowing when undertaking work with 3D printers.

ABS (Acrylonitrile Butadiene Styrene)

This is one of the most popular types of 3D printing filaments. ABS is durable and resistant to compression and abrasion and performs well at high temperatures. It's the perfect choice for printing elements that are exposed to long-lasting use in challenging conditions. It may, however, be difficult to work with, and it may emit hazardous and unpleasant odors during printing. It also requires the use of a heated print bed and glue, as it is prone to distortion otherwise. Its printing temperature ranges from 210 to 250 °C. ABS filaments [2] are made during the thermoplastic synthesis process. It involves mixing three components: acrylonitrile, butadiene, and styrene, in a special device called a continuous mixer, in order to obtain a homogenous mass. The mass is heated to a high temperature to melt it and then fed through the nozzle, which forms thin fibers. The next step involves winding the fibers onto rolls. Proper mixing of components and correct melting and forming of the fibers is crucial for obtaining high-quality fibers that will perform well in 3D printers.

Typical applications of ABS include, among others, household appliances, bike helmets, and building blocks for children. ABS is also used to make all types of enclosures, covers, handles, levers, or minor finishing elements. Importantly, though, the material should not have any contact with food.

PLA (Polylactide)

This is a non-toxic and biodegradable type of filament, easy to process and safe for the environment, which is why it's the most popular variant of 3D printing filaments nowadays. PLA is more flexible and less resistant to high temperatures than ABS. At the same time, it's easier to print, as it's not prone to distortion, does not require a heated print bed, and needs lower printing temperatures (180–230 °C). It also does not emit unpleasant odors. PLA has a wide scope of applications, which include, for example, display products (figurines) or construction components, and that is why it's available in many colors. PLA filaments [3] are made during the thermoplastic synthesis process. This process involves mixing natural raw materials (e.g. corn starch) with chemical additives, such as lactic (polylactic) acid, and polyethylene glycol. This mass, just like in the case of almost all filaments, is heated to a high temperature and then fed through the appropriate nozzle, which forms it into thin fibers. The next step involves cooling down the fibers and winding them onto rolls.

An important issue is that PLA has low resistance to higher temperatures, so the elements printed with it should not be used in temperatures exceeding 55–60 °C. Its major advantage, however, is that the components made of this material can be used as a base while working with special-purpose (e.g. conductive) filaments and that it is biodegradable.

PETG (Polyethylene Terephthalate Glycol)

PETG is a durable and flexible type of filament, fully resistant to water and chemicals. That is why it is often used to print objects such as bottles and other types of packaging. PETG filaments [4] are made in the process of thermoplastic synthesis of such raw materials as terephthalic acid, glycerol, and some additional chemical compounds, including antioxidants and stabilizers. A major advantage of this filament is its resistance to chemicals (oils, greases, or petrol), hence it is applied very commonly. Its characteristic feature is also its transparency, which matters in a lot of applications.

PA (Nylon)

PA is a durable and elastic type of filament, insoluble and resistant to abrasion and mechanical damage. Nylon is often used for printing components that must withstand heavy loads, such as sprocket wheels or extension springs. However, it is worth keeping in mind its hygroscopic properties and protecting oneself against the hazardous fumes it emits during printing. Nylon filaments [5] are made in the thermoplastic synthesis process involving such raw materials as azelaic acid and glycerol, with the addition of antioxidants and stabilizers.

PC (Polycarbonate)

This is a type of filament that is resistant to high temperatures and characterized by a high level of durability. PC, also called polycarbonate, is often used to print parts that must withstand high temperatures and loads. PC filaments are made during the thermoplastic synthesis process involving the mixing of such raw materials as bisphenol A and carboxylic acid with chemical additives in the form of antioxidants and stabilizers.

TPE (Thermoplastic Elastomer) and Its Variants

TPE is a synthetic material with rubber-like properties that make it remarkably flexible and durable. It should be noted, however, that this is not essentially one type of material, but a wide class of copolymers and polymer blends whose soft and elastic fibres withstand strains that are unattainable with ABS or PLA. TPE filaments [6] are commonly used to manufacture car parts, household appliances, and medical products, however, they are not easy to use — TPE is difficult to mould into the desired shape.

Thermoplastic polyurethane (TPU) [7] is a specific variant of TPE whose filaments are very popular. Compared to regular TPE, this material is slightly stiffer, which makes printing easier. It is also slightly more durable and better preserves its flexibility at low temperatures. However, just like TPE, it is also a demanding filament to print — it requires a narrow filament path and a low speed of printing at a temperature ranging from 210 to 230 °C.

Another TPE variant is a thermoplastic copolyester (TPC). It is not as commonly used, but in some applications it has the advantage of higher resistance to chemicals, UV radiation, and temperature, which can even reach up to 150 °C without damaging the material.

3D Print in Prototyping

The importance of 3D printing for prototyping cannot be overestimated nowadays. This technology makes it possible to create fast and inexpensive physical models, which shortens the designing and testing phases and makes it possible to implement the final version of the item much quicker. By relying on 3D printing, the designers and engineers can create many physical variants of a given item and use them for testing the shape, size, and functionality in order to select the best option. It works perfectly, e.g. in the automotive industry, where 3D printing is used frequently when designing and creating prototypes for car body components or parts that are installed later inside the cars.

Another example of 3D printing in prototyping is creating medical models, such as prostheses, endoprostheses, or surgical tools. Here, too, reducing the time of searching for the final form is very important, and thus the popularity of 3D printers in the medical industry is rapidly increasing. Finally, it's worth mentioning that in prototyping the injection moulds themselves, here, once again, the use of 3D printing makes it possible to manufacture new plastic products quickly and easily.

There is one more aspect of using 3D printing in prototyping processes — the financial one. Making the prototypes yourself instead of outsourcing them costs much less and at the same time protects better the manufacturer's intellectual property.

3D Printing in Electronics

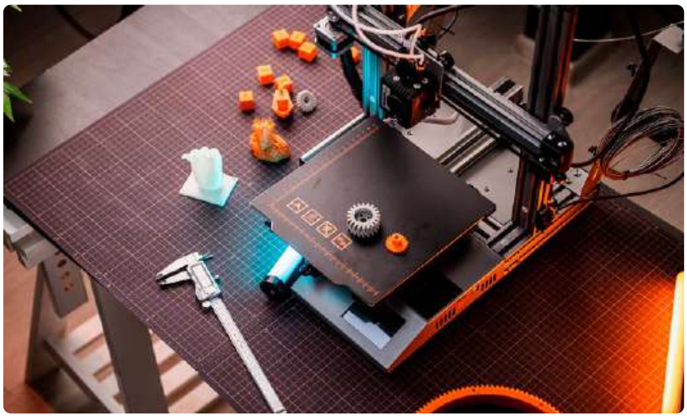
It is obvious that 3D printing has applications in electronic design, too. During CAD/CAM designing, separate programs design PCBs, electrical engineering components, and enclosures. Thanks to the fact that the enclosures can be printed immediately, they can be quickly and efficiently designed and modelled until the optimum design is achieved.

The newest solutions give users even more: printing enclosures [8] with electronic circuits embedded in them. This is the result of combining two processes that used to be independent — 3D printing and automatic placement of components on PCBs [9] — with a very interesting solution in the form of a conductive filament. This results in a machine for manufacturing fully functional electronic devices. First, the entire conductive circuit is designed in a suitable environment, and then it is sent to the software, which combines it all into a file from which the printer reads information about printing the enclosure, creating conductive connections and the component layout.

3D printing also enables short-run production, custom order production, and the development of components for beta testing. Moreover, it also makes it possible to create spare parts for non-standard devices, which are usually quite expensive.

WEB LINKS

[1] 3D printing filaments: https://tinyurl.com/tmefilament
[2] ABS filaments: https://tinyurl.com/tmeabsfilament
[3] PLA filaments: https://tinyurl.com/tmeplafilament
[4] PETG filaments: https://tinyurl.com/tmepetgfilament
[5] Nylon filaments: https://tinyurl.com/tmenylonfilament
[6] TPE filaments: https://tinyurl.com/tmetpefilament
[7] TPU filaments: https://tinyurl.com/tmetpufilament
[8] Enclosures for electronics: https://tinyurl.com/tmeenclosures
[9] Universal PCBs: https://tinyurl.com/tmeunipcb
[10] ASA filaments: https://tinyurl.com/tmeasafilament
[11] HIPS filaments: https://tinyurl.com/tmehipsfilament
[12] PCABS filaments: https://tinyurl.com/tmepcabsfilament
[13] PMMA filaments: https://tinyurl.com/tmepmmafilament
[14] PVA filaments: https://tinyurl.com/tmepvafilament
[15] SILK filaments: https://tinyurl.com/tmesilkfilament
[16] Filaments from the iglidur series:
https://tinyurl.com/tmeiglidurfilament



Source: © ProstoSvet / Adobe Stock

Filaments Available from TME

TME offers filaments manufactured by many renowned suppliers and presents a wide spectrum of materials made of synthetic and natural components.

These include such materials as ABS in many variants, i.e. ASA (acrylonitrile styrene acrylate) [10], HIPS (high-impact polystyrene) [11], PA (nylon, i.e. polyamide), PCABS (blend of polycarbonate and ABS) [12], PET (polyethylene terephthalate), PLA (polylactide), PMMA (polymethyl methacrylate) [13], PVA (polyvinyl alcohol) [14], SILK (glossy variant of PLA) [15] and thermoplastic elastomers — TPE and TPU. Moreover, there are also filaments from the iglidur series [16] adapted to specialist industrial applications, based on highly efficient base polymers with the addition of different types of fibers and solid greases. These latter components provide the preferred properties that are required in specific niche applications, for example the necessary durability, resistance, or limited friction and wearing of the elements printed with them. Due to that, they will be suitable for e.g. manufacturing bearings. ◄

230577-01

Transfer Multisort Elektronik Sp. z o.o.
Headquarters: ul. Ustronna 41, 93-350 Łódź, Poland
export@tme.eu - www.tme.eu/en

TME
Electronic Components

TRANSFER
MULTISORT
ELEKTRONIK
GLOBAL DISTRIBUTOR OF ELECTRONIC COMPONENTS

Tired by waiting for standard PI ?

bpi

Transfer Multisort Elektronik Sp. z o.o.
Łódź, Poland, export@tme.eu

Join us:

tme.eu
YOU NEED IT, WE HAVE IT!
tme.com

Specialists for Effective Signal Analysis from ELF to EHF Band

Aaronia's Latest Real-Time SPECTRAN® V6 Series Spectrum Analyzers

Contributed by Aaronia AG

The millimeter wave or extremely high frequency (EHF) band offers enormous bandwidth and high resolution due to its high frequencies and associated short wavelengths. However, the high frequency also poses a major challenge for signal analysis, as the propagation characteristics of EHF signals are susceptible to absorption and scattering. With the USB real-time spectrum analyzers of the SPECTRAN® V6 series, Aaronia AG provides the appropriate tools for all required analyses.

The real-time spectrum analyzers of the SPECTRAN® V6 X USB series are specially designed for near- and far-field measurements, for measuring and localizing sources of interference radiation or for monitoring EMC problems. The real-time bandwidth of up to 450 MHz and the sweep speed of >3,000 GHz/s accelerate all necessary measurements enormously and thus save time and money.

Within the frequency range of 9 kHz to 55 GHz, the new SPECTRAN® V6 X devices of the PLUS series can detect and localize even extremely short interfering signals and determine or eliminate their cause. The sweep speed of the ECO, 5G, and XPLORER series is up to 3 THz/s. It enables a faster update of the spectrum and the detection of transient signals, which is particularly important when analyzing frequency-hopping signals or when searching for intermittent interference.

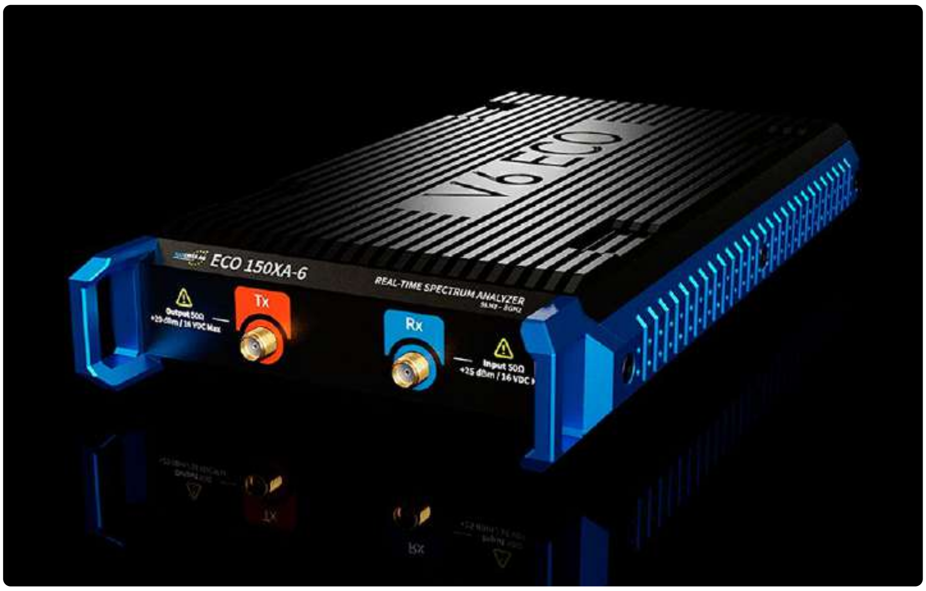


Figure 1: SPECTRAN® V6 ECO 150XA-6 with one input (Rx) with 44 MHz RTBW and an additional 44 MHz Tx output

For example, if SPECTRAN® V6 analyzers with two inputs are used, the two inputs can be used to measure simultaneously in front of and behind a shield. Both results can then be compared with each other in real time without changing the measurement setup or having to reconnect cables, for example. This simplifies measurement tasks and reduces the effort considerably.

By combining several SPECTRAN® V6s, the real-time bandwidth can be increased as desired. Cascading just four V6s allows gapless real-time measurement from 20 MHz — 1 GHz, for example, which means an unbeatable time advantage for a large number of measurements. "Measurement technology, as well as the demands placed on it, are changing at breakneck speed," says Thorsten Chmielus, Managing Director of Aaronia AG. "With our USB real-time spectrum analyzers, we are able to react quickly to new circumstances. With the continuous further development of the SPECTRAN® series, we are always setting new benchmarks and future-proofing our products."

Powerful for Small Budgets — SPECTRAN® V6 ECO

With the SPECTRAN® V6 ECO (**Figure 1**) Aaronia opens a new market segment in real-time spectrum analysis. In its basic version, the USB device offers a real-time bandwidth (RTBW) of 44 MHz and an extended frequency range from 9 kHz to

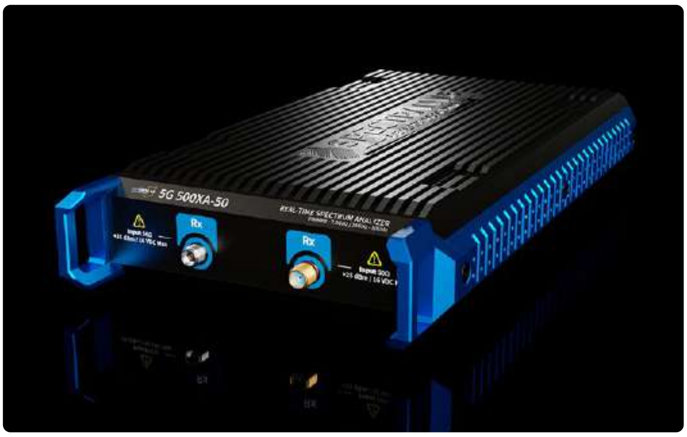


Figure 2: SPECTRAN® V6 5G-500XA-50 with Rx "low frequency" and Rx "high frequency."

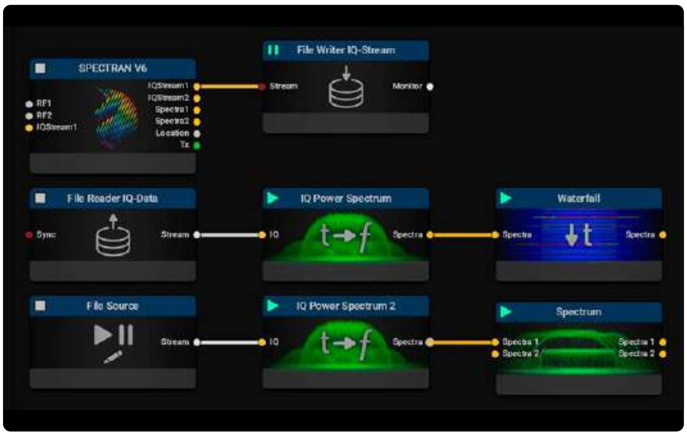


Figure 3: RTSA-Suite PRO recording the full IQ bandwidth.

6 GHz. The entry-level ECO 100XA-6, including the dedicated RTSA-Suite PRO Software costs less than 1,500 euros, a revolution in the market of real-time spectrum analyzers. Alternatively, the device can be ordered with an integrated signal generator or as a dual RX variant. The latter closes the gap between affordable, yet low-power SDRs and expensive and fast, high-end analyzers. With its two independent inputs, the SPECTRAN® V6 ECO 200XA-6 offers 2 × 44 MHz RTBW on the one hand. On the other hand, it achieves up to 3 THz/s sweep speed with the unique, patented high-speed tictoc LO function.

Next year, the devices will also be available as an 18 GHz version for less than 10,000 euros, including a signal generator. Furthermore, almost all models will be expandable with an optional PowerMeter input, which can reliably detect signal peaks up to 70 GHz.

For measurements in the mobile radio environment, the devices of the SPECTRAN® V6 5G series (**Figure 2**) are convincing. In addition to the previous FR1 band, they also support the new 5G bands between 24 GHz and 53 GHz (FR2) as well as WiGig 45 GHz (802.11aj). The associated frequency profiles are now included in the latest version of RTSA Suite PRO.

Key Factor Software

However, it is not only the real-time bandwidth as well as POI and sweep speed of the hardware that are astounding. Rather, the software used plays a decisive role. With the modular RTSA-Suite PRO real-time spectrum monitoring and analysis software, Aaronia provides a powerful software package for signal recording and data analysis. Thus, the Record & Replay function of the SPECTRAN® V6, in combination with RTSA-Suite PRO, allows for recording and replay of the full IQ bandwidth (**Figure 3**). In this way, all the information necessary to recover a signal can be stored on the locally connected computer for detailed examination. Another invaluable advantage is the practically unlimited recording duration, which depends only on the capacity of the storage media used.

Ease of use is a top priority for Aaronia. Thanks to the modular system, even complex measuring tasks can be configured within a very short time by drag-and-drop. By combining different blocks, a visual image of the measurement setup is created. Frequently-used so-called "missions" can be saved as finished projects and recalled as needed. Numerous predefined configurations are available for download on the Aaronia website [1]. These can be imported into RTSA-Suite PRO, so that measurements can be started immediately. If the additional functions are subject to charge, Aaronia offers all existing customers the option of testing the function in 30-day full-feature trial versions.

Many modules, with a total value of 7,782.00 euros, are already included free of charge in the basic version. These include various 2D and 3D views, IQ processing, triggers, AM/FM decoders, file reader and file writer, remote HTTP or scripts, among others.

Not Only for the SPECTRAN® V6

The RTSA-Suite PRO can not only be used with Aaronia measuring instruments, but is also compatible with other branded devices and extends their range of use. For example, if you extend your Tektronix spectrum analyzer with RTSA-Suite PRO via a software license for about 2,500 euros, you will receive the file writer/reader for saving and playing back data, worth 4,550 euros, as a bonus.

Upgrade Service

Aaronia offers an exclusive trade-in service for its products. If an old device or the previous version of a product is to be exchanged for a successor, up to 50% of the original price will be credited to the new device. All you have to do is send in the old unit and the original invoice. This offer applies to all active components, such as measuring instruments and amplifiers.

And the Future?

Aaronia is already working on the next SPECTRAN® generations. With a frequency range of up to 110 GHz (depending on the model), the SPECTRAN® V6 XPLORER is the new player in the high-frequency range. As a cost-effective development platform, it opens up completely new areas of application, due to its performance.

230580-01

WEB LINK

[1] Website Aaronia: https://aaronia.com

Challenges of DFM Analysis for Flex and Rigid-Flex Design

Contributed by DownStream Technologies

Flex and rigid-flex PCB construction is not a new concept. It has become commonplace as engineers look for alternative circuit packaging for ever shrinking electronic products. A flat one-sheet schematic for a straight ribbon cable is analogous to its physical flat substrate. A flat multi-sheet schematic that details circuitry for a rigid-flex design bears little visual resemblance to its three-dimensional, variable material rigid-flex assembly. However, in both schematic examples, schematic-based analysis tools are applied equally. This same truth also applies to common FR4-based two-layer or multilayer PCBs. Today's PCB analysis tools are applicable across all combinations of rigid PCBs, regardless of layer count or size. However, due to unique properties of flexible substrates and combined flexible and rigid substrates, flexible designs require a specific collection of analysis — both functional and manufacture-oriented. Integrity analysis such as impedance, coupling, crosstalk and noise is complicated by variable stackups across flexible designs. A single transmission line can be stripline in a rigid-flex area and microstrip in a flex area. Material types and dielectric constants above or below a trace as it traverses a design also vary. While the challenges for signal integrity analysis for flexible designs are worthy of conversation, this article will focus on the current challenges to *Design for Manufacture* analysis of flex and rigid-flex designs.

Contrast Rigid PCBs With Flex and Rigid-Flex

Some designers design flexible PCBs as simple bendable circuit boards, but there are vast differences between rigid and flexible. Both technologies produce an electrical interconnect function, but are manufactured using different types of materials and processes. They also have varying applications. No need to design a rigid-flexible PCB for the motherboard of a desktop PC, but rigid-flex is required for most medically implanted devices.

A typical rigid PCB is composed of electro-deposited copper-clad fiberglass substrates bonded together. While there are variances in materials used to bond substrates, it is commonly sheets of cloth pre-impregnated with uncured epoxy. This bonding material composition is not engineered to be flexible. The copper is chemically etched to create a circuit pattern. The hardness of the bonded substrates requires mechanical routing to trim the raw PCBs. All layers of the PCB are commonly identical in size and shape unless cavities, embedded components, or other such exotic construction are present. The rigid PCB layer stackup is identical across the entire PCB area. Soldermask and silkscreen are almost always applied.

Flexible PCBs are comprised of rolled annealed copper over flexible polyimide substrates. Flexible layers or cores are produced with or without adhesives. Adhesiveless flex is prevalent in applications requiring higher performance, while those with adhesives are often found in low-layer-count applications. The most common usage is copper foil laminated to a substrate with epoxy or acrylic adhesive. Both substrate material and adhesive are engineered for bending to minimize trace fracture. Like rigid PCBs, a chemical etching process is used to create a circuit pattern. The flexible nature of the materials requires die cutting or "blanking," rather than mechanical routing. Each layer of

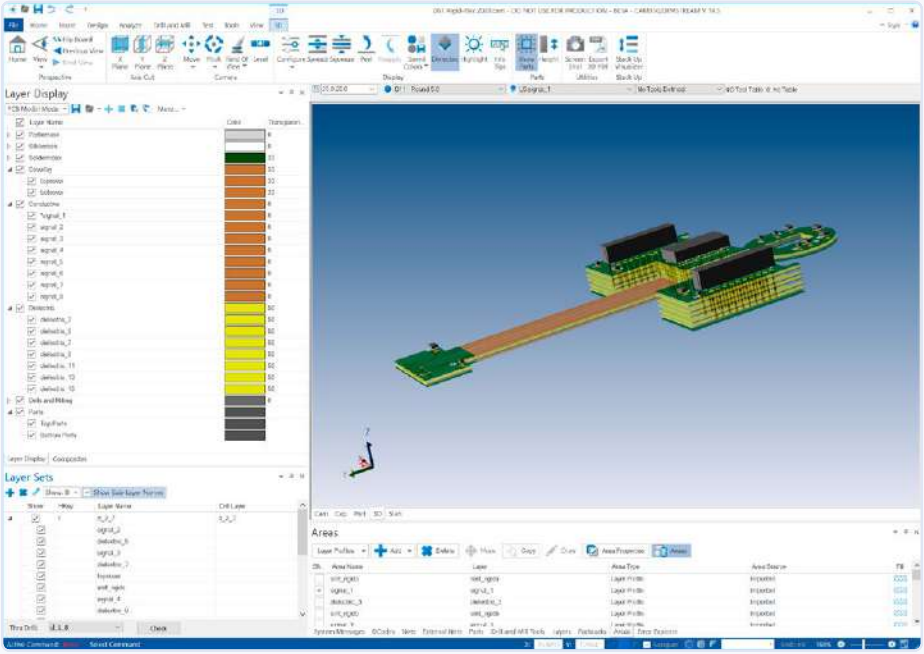


Figure 1: A rigid-flex design in 3D with layers separated for improved visualization.

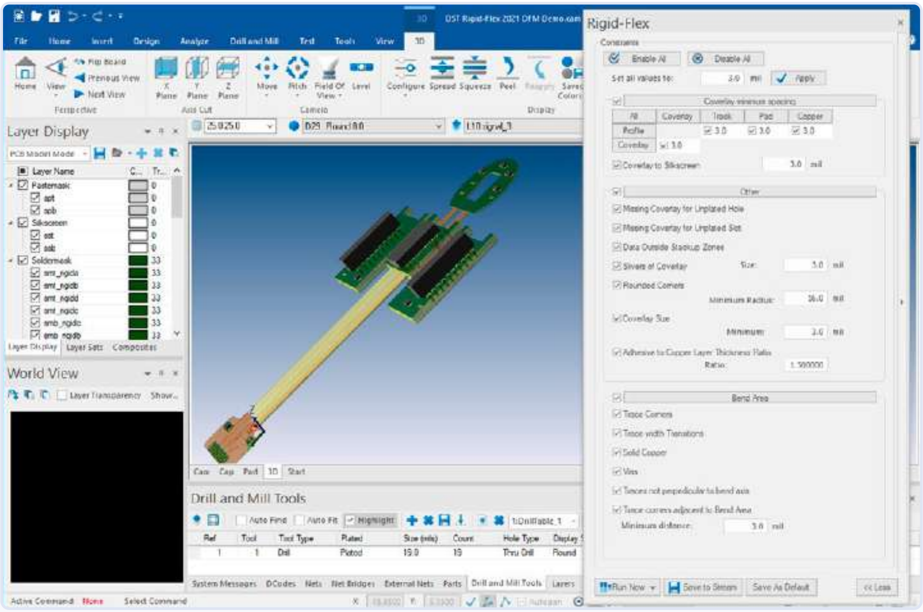


Figure 2: Rigid-Flex DFM analysis currently available in DownStream's DFMStream product.

a flexible double-sided core has an identical shape. However, multiple-layer flex is likely to have variances in shape for each layer or core. Flexible PCBs require a thin film insulator over the conductors, known as a coverlay. Unlike rigid PCB soldermasks, coverlays are die cut, much like the flexible layers they insulate. The stackup of a multi-layer flexible PCB can vary across the PCB area. This is especially true with multi-layer flex, where layer shape varies among the collection of layers or cores. A flexible ground or power plane area is typically crosshatched, versus solid for rigid PCBs. The crosshatch reduces potential for conductors to fracture. Alternatively, flexible layers can be shielded with a layer of copper or silver foil. Masking and screening over flexible layers is not rare, but uncommon.

Rigid-flexible PCBs are obviously a combination of rigid and flexible materials. Rigid-flex is in essence a hybrid PCB combination of materials and processes from both rigid and flexible PCBs. The two material types are generally processed separately and bonded together later in the fabrication process. The layer stackup commonly varies greatly across the entire PCB (**Figure 1**). There may be areas of rigid-flex, flex only, various combinations of rigid and flex layer count and so on.

There is also rigidized flex, where blank FR4 or other rigid materials are selectively bonded to flexible substrates to provide stiffness. The rigid stiffener material rarely has conductors present.

Application of Rigid vs Flex and Rigid-Flex

Rigid PCBs are a foundational technology in today's electronic products. Rigid PCBs offer mechanical integrity and electrical conductivity and reliability, but are limited by their two-dimensional profile. Their flatness limits designers to two dimensions, which severely limits design flexibility, especially as electronic devices decrease in size. Flexible PCBs are bent to take advantage of a three-dimensional space, while also accommodating components. Flexible PCBs enable maximal utilization of space to package electronics, but at a premium cost compared to conventional PCBs.

Rigid and flexible PCBs are present in many electronic products. However, some applications benefit more from one type of circuit board. Rigid PCBs make sense for products such as televisions, desktop PCs, Blu-ray players and other larger electronic products. Flexible PCBs are present in smartphones, smartwatches, tablets, cameras, printers and laptops. They are a fundamental requirement for implanted miniature medical devices such as pacemakers, cochlear implants, and implanted defibrillators. Complex multi-PCB assemblies interconnected with wires or cabling are often redesigned with rigid-flex PCBs to improve reliability and reduce weight and space. This is the catalyst for many military and aerospace products being designed with rigid-flex. One example is a single use smart bullet that can change its trajectory if its intended target moves.

The introduction of small outline, or all surface-mounted semiconductors ushered in a revolution of miniature re-packaging. Think Sony Walkman versus a typical boom box. For years, flexible PCBs were relegated exclusively to replacement of multi-wire cables. Who would not recall the presence of a flexible, flat cable connected to the head of a dot matrix or impact printer. The head would bob back and forth across the paper while the cable dynamically flexed and provided a more reliable interconnect between the printer head and motherboard. The introduction of rigid-flex is not the same game changer as surface mounted packages because of its somewhat limited application and cost differential. We shouldn't expect a new collection of desktop PCs to be designed with rigid-flex motherboards as a means to reduce costs. However, miniaturized and reliable technology such as pill cameras, foldable cell phones or implanted medicals devices would not be without rigid-flex technology.

Conclusion

Designers and fabricators alike have managed fairly well to date with limited access to flex-specific DFM analysis tools. Today, Flex and rigid-flex have become more main stream and the underlying technology is continuously evolving. As is common with all newer technologies, PCB design and analysis tools are playing catch-up. PCB CAD tools have now been updated to support design for flex and rigid-flex, but many still lack the support needed for intelligent data relaying to fabricators. Likewise, most DFM tools to date have been inadequate for properly analyzing flex and rigid-flex designs for manufacturing problems. At DownStream, we are fortunate to have a long list of users fully enmeshed in flex designs who partnered with us to develop a flex-specific DFM solution, and we continue to work with these customers to enhance our capabilities for flex DFM analysis. In addition to the DFM analysis support described in this document, our plans include the ability to analyze additional trace fracture potential, such as I-beaming, as well as improved 3D visualization and DFM for flex and rigid-flex in their bent state (**Figure 2**). These are just a few examples. Like the underlying technology, the PCB design and analysis tools must also continuously evolve to ensure customer success, and this is a cornerstone of Downstream's commitment to our industry.

230590-01

Setting Up an SMT Line

Contributed by Opulo

Manufacturing your electronic designs is a challenging task. At very small scales, using tweezers and manually placing components onto a pasted PCB will do, but anything more than a couple of dozen boards makes this plan infeasible. If you're making hundreds of thousands of units a year, hiring a factory to do all assembly, programming, and quality control can be a good solution. But what about in between? We've been stuck with two stark options: inefficient manual assembly or expensive outsourcing.

But now, you can run an SMT line for production quantities of boards while saving money on expensive contract manufacturer overhead and ensuring quality by keeping production in-house. In this article, we'll show you how to set up and operate your own SMT line for manufacturing your product.

Solder Paste and Stencils

The first step in running SMT is applying solder paste. This is the process of using a squeegee to move paste across a stainless-steel stencil placed on your PCB, which results in paste only where the components require it (**Figure 1**). There are three critical components to having a successful paste printing setup.

Paste

Choosing the correct paste for your board can mean a world of difference for the pass rate of your production line. A good paste will stencil easily and reflow cleanly, with no errant solder balls. Chipquick SMDLTLFP250T3 is a great low-temperature paste that is very easy to rework. It can also be great for reflow ovens that struggle to reach the standard 250 °C peak reflow temperature. However, if your reflow oven can reach 250 °C easily, Loctite GC10 is an excellent paste choice. It has an incredibly long shelf life, and reflows cleanly with very low likelihood of bridging or cold joints.

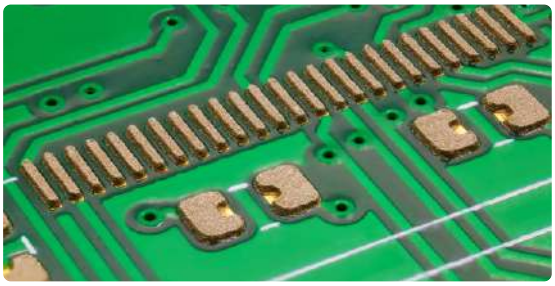


Figure 1: LumenPnP Motherboard Solder Stencil.

Stencil

In order to apply solder paste only where the components on your board require it, you'll also need a stencil. These are thin sheets of laser-cut stainless steel with voids that line up with the pads on your PCB, so a squeegee only deposits paste in the correct spot. You can typically buy these custom-made from your board shop very inexpensively.

Getting your stencil electropolished is a common finishing option at most board shops, and we've found that it results in a longer stencil lifetime and a cleaner paste release. You can also choose to get your stencil "framed" which means the sheet of steel is mounted tightly in an aluminum frame for use with a paste printing jig.

Printing Jig

A paste printer is a jig that perfectly aligns your stencil to your board in a repeatable manner. Achieving precision alignment of the stencil to your PCB is essential. Any deviation can result in paste being applied across pads, greatly increasing the likelihood of a solder bridge. The 3040 solder paste printer is a reliable and cost-conscious choice, holding its alignment well after setup.

Component Placement

Arguably the most difficult part of PCBA production is placing components onto the bare PCB with precision. And, with difficulty come other challenges; even the lower end of these machines can cost tens of thousands of dollars, and require an industrial environment with high-pressure air to operate.

However, the LumenPnP (**Figure 2**) [1] is a desktop pick-and-place machine that allows you to populate your boards for a fraction of the cost, in any setting. The LumenPnP supports components down to 0402, and it even has powered feeders that serve up a steady stream of parts for the machine to place, meaning the assembly process is completely automated. After setup, it's just a matter of mounting your pasted PCB and running the assembly job.

It's also designed to be highly configurable, supporting a wide range of board requirements. The machine supports up to 50 powered feeders, with space for a couple of dozen strip feeders, meaning that even a board with many unique parts can be assembled in just one job. And even if you have a very large board to populate, the LumenPnP can assemble panels as large as 360 mm × 270 mm.

The LumenPnP is fully open source, meaning that you can modify, understand, and maintain your machine indefinitely. It also has a community thousands strong, always working on upgrades, mods, and improvements to the machine.

Reflow

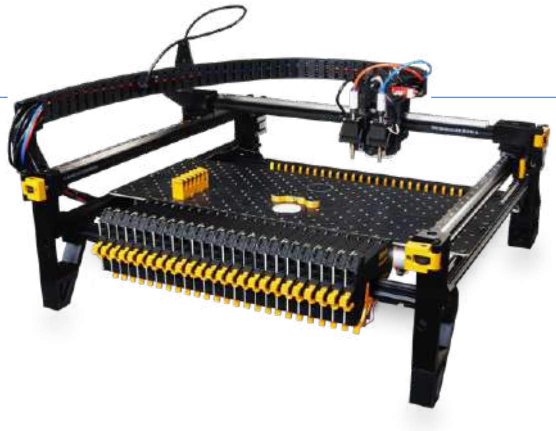
With a pasted, populated board, you now need a reflow oven to heat the solder paste and cause it to reflow into molten solder. This heating process should follow a specific temperature profile optimized for your chosen solder type.

Reflow Master Pro

This hardware module converts a standard toaster oven into a reflow oven. With a fantastic UI and great performance, the Reflow Master Pro is what Opulo uses for all PCBA production. You must provide your own toaster oven and handle the installation yourself, but the results are worth the effort.

Whizoo Controleo3

Whizoo offers a completely off-the-shelf solution for those that are just looking to buy something that's ready to reflow out of the box. It's quite a bit more expensive than other options, but all of the assembly is done for you. The Controleo3 also comes with a mechanism to vent the hot air from the reflow chamber after the process is done, helping to follow the cooldown period of the temperature profile more closely.



◄ *Figure 2: LumenPnP 3.1 with Powered Feeders.*

Testing

After reflow, it's critical that you test your PCB. There is no one-size-fits-all approach to this as it depends heavily on your design, but there are a few tools that can help with performing speedy and accurate tests on your completed PCBAs.

Although it requires modification to suit your product, an off-the-shelf testing jig can provide a great place to start when building up a testing solution. These jigs provide a tray where the device under test (DUT) can mount, and a plate that articulates downward to push the DUT into an array of spring-loaded test pins that make contact with the PCB to check connections.

These pins will need to make contact with dedicated test points on your PCB, so you'll need to take this test jig into consideration even while you're designing your product. Of course, this is just a consistent interface with your product; you'll still need to develop any hardware and software that's actually doing the testing.

However, a functional test might be quicker and easier to deploy than using a pogo pin jig. This means actually using the product as it's intended to be used as a test, such as connecting motors to your PCBA and checking that they move correctly. The tests you deploy depend on your product and scale.

Conclusion

With solutions in place for all the discussed topics, you're well on your way to establishing a finely tuned SMT assembly line. The advantages of in-house assembly are significant, liberating you from the constraints of larger companies' demands for massive orders at better prices.

For further insights on hardware manufacturing, check out *Open Hardware Manufacturing Podcast* [2], where we discuss best practices for making your hardware, and interview folks in open hardware to see how they solve difficult problems in manufacturing. ◀

230593-01

WEB LINKS

[1] LumenPnP: https://opulo.io/products/lumenpnp
[2] "Open Hardware Manufacturing Podcast": http://shows.acast.com/ohm-podcast

The Right Combination for a Reliable Assembly

By Paolo Corviseri and Fabian Volmer (Balver Zinn)

Because of ongoing miniaturization, increasing operating temperatures and device packing density of electronic assemblies, solder material manufacturers have to respond with more reliable, more cost-effective and sustainable solutions. Previous standard alloy compositions, such as SAC305 or other alloys, are gradually being replaced by new alloy compositions. The BALVER ZINN alloy SN100CV® is one of these new, highly reliable, cost-effective alloy compositions.

Reliability and Thermal Stability of Lead-Free Solders

Extreme operating temperatures of electronic assemblies call for new solutions. SN100CV® alloy, developed by Nihon Superior and licensed by BALVER ZINN, offers a micro-doped alloy that offers a high reliable solution for these applications.

The technical advantage of SN100CV® originates in the solidification mechanism. Silver-containing alloys, such as SAC305 and other alloys, are subjected to phase growth after thermal stress (aging). This effect (called Ostwald ripening) results in grain coarsening of the alloy. The alloy becomes more grainy and results in a more brittle and degraded alloy.

SN100CV® shows no coarsening effect due to a different strengthening mechanism called Solid Solution Strengthening (**Figures 1** and **2**).

Incorporation of atoms in the alloy with a larger diameter (bismuth atoms) into the tin matrix results in a distortion of the microstructure lattice, leading to a more reliable alloy.

SN100CV® is a lead-free and silver-free alloy where the mechanical properties are dramatically improved by adding only 1.5% bismuth. It offers higher thermal stability and is more cost-effective as the silver is eliminated from the alloy composition. Besides these advantages, the dissolution rate of copper (Cu) is dramatically reduced, making SN100CV® ideal for utilization in liquid soldering processes such as wave / selective and tinning applications.

SN100CV® is less susceptible to stresses caused by thermal and mechanical effects.

The grain coarsening effect due to thermocycling tests are shown in comparison for SAC305 and SN100CV® in **Figures 3** and **4**.

SN100CV® is also available in solder paste and Cored Solder Wire (CSW).

It is observed that the mechanism of Solid Solution Strengthening in the SN100CV® alloy results in significantly lower failure probabilities as other alloys.

Additional microdoping in the alloy, like the addition of nickel, additionally increases the thermal load of the alloy. The nickel barrier in the Inter Metallic formed during the soldering process inhibits phase growth, which is beneficial in aging behaviour. These benefits of SN100CV® alloy are also introduced in other product groups like solder paste and Cored Solder Wire.

JEAN-151 solder paste platform is now also available with the robust properties of the SN100CV® alloy and can be implemented as a cost-effective, high-reliability solution in all reflow processes. The solder paste is classified ROL0 according to IPC J-STD-004 and available in 9 different alloys and 4 different solder powder sizes with all of them exactly the same flux technology.

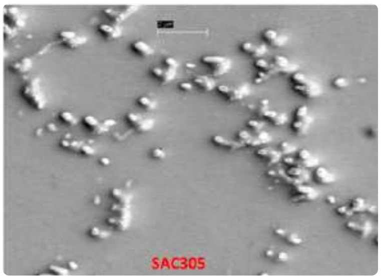


Figure 1: SAC305 after aging at 125 °C for 2520 h.

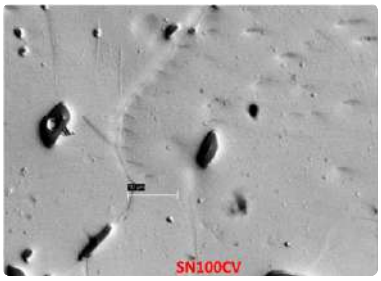


Figure 2: SN100CV® after aging at 125 °C for 4200 h.

The robust process capabilities of the JEAN-151 solder paste resulted in a shelf life of 1 year (< 20 °C), without any degradation of the printing or soldering properties.

The most popular alloy in reflow soldering processes is still the SAC305 alloy, with a melting range of 217–219 °C (**Figures 5-10**).

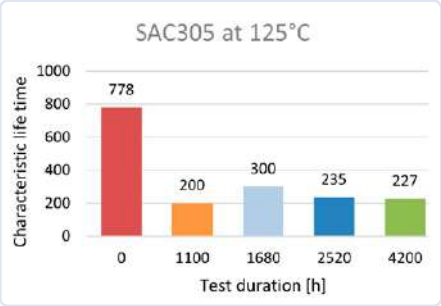


Figure 3: Characteristic lifetime of SAC305 after aging for 4200 h.

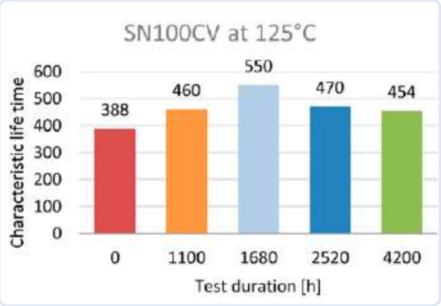
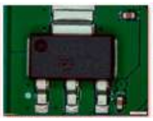
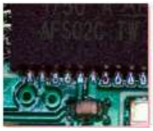


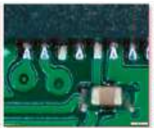
Figure 4: Characteristic lifetime of SN100CV® after aging for 4200 h.

5


6

7


8


9

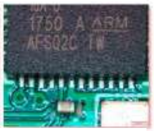

10

Figure 5: Reflow at oxygen atmosphere. Figure 6: Reflow at nitrogen atmosphere. Figure 7: Vapour phase process. Figure 8: Reflow at oxygen atmosphere. Figure 9: Reflow at nitrogen atmosphere. Figure 10: Vapour phase process.

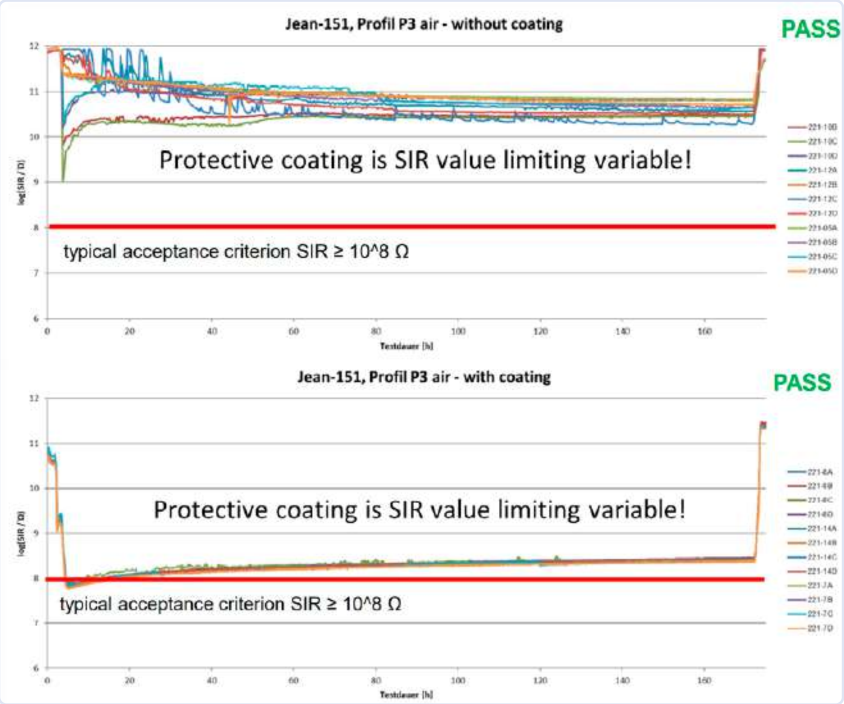


Figure 11: SIR tests of the solder paste JEAN-151 SAC305 T4 with nitrogen as well as with/without protective coating from Lackwerke Peters.

The SABI6 alloys (SnIn5Ag3.5Bi0.5) and SN100CV® are considered as very high reliable alloys and are successfully implemented in automotive and e-mobility applications.

The JEAN-151 SN100CV® solder paste is besides a technically premium product, considerably less expensive due to the absence of silver in the alloy. To confirm/verify the process capabilities of JEAN-151 SAC305 T4 solder paste, the product was subjected to the independent Siemens CT test. This test procedure includes SIR test with and without conformal coating. All tested process and material combinations fully met all requirements (SIR ≥ 10^8 Ω) without showing signs of corrosion, dendrites, discoloration or any other anomalies (**Figure 11**).

The coated comb structures (coating: Twincure DSL 1600 E-FLZ / Lackwerke Peters) showed significantly lower SIR values than the uncoated test structures, showing the high reliability of the JEAN-151 solder paste.

The high reliable SN100CV® is, of course, also available as Cored Solder Wire SN100CV® LF2220NC, a REL0 classified, halide-free no-clean solder wire.

Increased soldering temperatures and increased Cored Solder Wire feeding speeds in manual and robotic soldering applications are prone to flux splattering.

Due to big temperature differences between the melting point of the solder alloy and the boiling point of the solvents / resins in the flux core of the wire, high vapour pressures will be formed inside the Cored Solder Wire, resulting in splattering of the flux.

With the development of StarCore® Cored Solder Wire, BALVER ZINN group has introduced a special flux core shape. This shape gives faster alloy melting at some areas in the wire and results in lower vapour pressure in the core of the wire. As a result, the number of flux splashes is significantly reduced compared to a standard single- or multi-Cored Solder Wire. The optimized StareCore® Cored Solder Wire additionally shows faster wetting, better soldering behaviour and the flux residues appear considerably clearer.

The latest StareCore®234E low-activated flux system features high SIR values ≥ $1 \cdot 10^9$ Ω (IPC-J-STD-004A) and significantly clearer residues. It also gives improved wetting properties and increased flow characteristics, resulting in efficient flux distribution. The ROL0 — activator system offers equal soldering properties compared to many M1 classified Cored Solder Wires.

SN100CV® alloy is considered a high-reliability alloy and offers perfect liquid soldering processes (wave/selective) in combination with versatile fluxes. In sustainable VOC-free processes, Balver Zinn Group offers reliable VOC-free fluxes like REGI-BLUE. It incorporates a special ingredient eliminating corrosion risks, which makes the REGI-BLUE the safest VOC-free soldering flux in the world (**Figure 12**).

REGI-BLUE also meets the SIR limit criteria of $1 \cdot 10^8$ Ω (IPC-J-STD-004A) in uncleaned state as a no-clean product. It shows no signs of electrochemical migration or dendrites, with hardly any visible residues and excellent soldering properties. Besides REGI-BLUE (water-based), REGI-RED (alcohol-based) and other derivatives of the derivatives of the REGI series, we are currently working on new, better and stronger products.

Summary

After the success story of SN100C®, BALVER ZINN is once again breaking new ground with the alloy SN100CV®. SN100CV® is an adequate alternative to high-priced, silver-containing alloys such as SAC305 and is compatible with all common processes. SN100CV® is available as solder paste with JEAN-151 as well as solder wire with e.g. StarCore® and is compatible with the latest flux series REGI.

BALVER ZINN and SN100CV®, with the right combination to a reliable assembly! ◀

230600-01

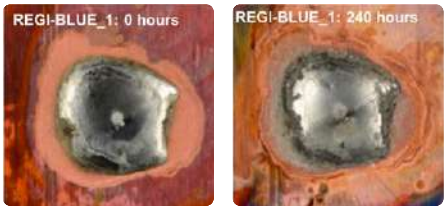


Figure 12: Copper corrosion test of REGI-BLUE; test coupon for (O h) and after the test method (240 h) without any indications of corrosion.

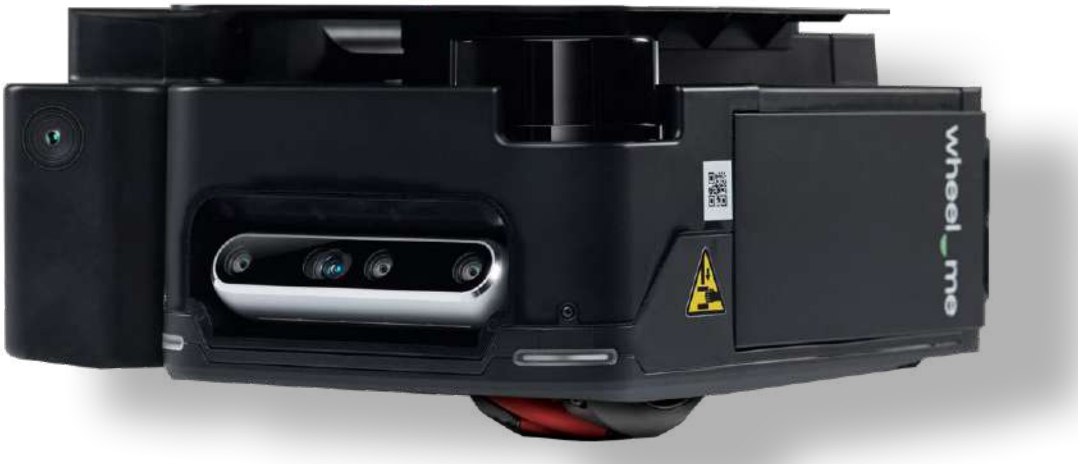


Figure 1: wheel.me Genius 2.

Revolutionizing Industries

The Rise of Autonomous Mobile Robots (AMRs)

Contributed by wheel.me

In the ever-evolving landscape of technology, one innovation stands out for its transformative potential in many industries — autonomous mobile robots (AMRs). These intelligent machines, equipped with advanced sensors, navigation systems, and artificial intelligence capabilities, reshape how we approach tasks and processes across various sectors. We will explore the fascinating world of AMRs, their technological landscape, and practical applications across manufacturing, logistics, healthcare, and more.

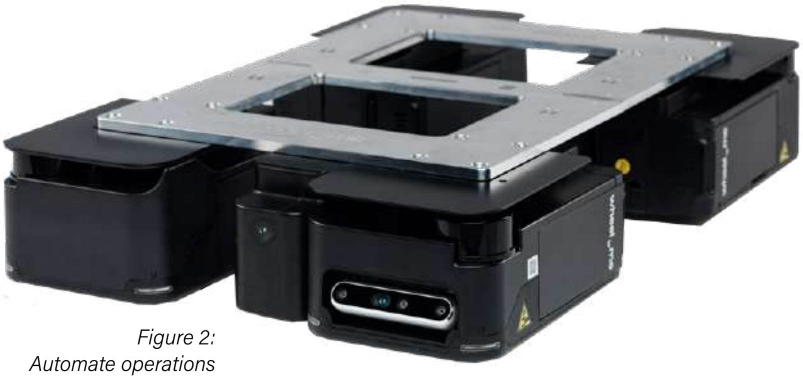


Figure 2: Automate operations with flexible AMRs.

Autonomous mobile robots (AMRs) epitomize a convergence of cutting-edge technologies, navigating autonomously within predefined environments, thanks to a synergy of critical components (**Figure 1**). Sensors, akin to their sensory organs, provide real-time data about surroundings, ensuring obstacle detection and safe navigation. Advanced navigation systems utilize this sensory data to craft and continually update maps of the environment, enabling precise navigation and adaptation to dynamic conditions. Furthermore, artificial intelligence (AI) serves as their cognitive engine, enhancing AMRs' efficiency by allowing them to learn from experience, optimize routes, and adapt to ever-changing circumstances. In essence, AMRs seamlessly amalgamate sensors, navigation systems, and AI, rendering them invaluable assets renowned for their autonomy and versatility across diverse industries.

AMRs in Manufacturing

AMRs are revolutionizing production lines in the manufacturing sector by seamlessly integrating them into existing processes. They can transport materials, deliver components, and execute tasks precisely and efficiently. Benefits include increased efficiency, flexibility, and cost savings (**Figure 2**).

AMRs in Logistics and Warehousing

The logistics and warehousing industry has also been quick to embrace AMRs. In large-scale warehouses, these robots streamline operations by efficiently moving goods, picking orders, and managing inventory. AMRs can work alongside human employees, enhancing productivity and reducing the risk of workplace injuries. Key benefits here include optimized order fulfillment, inventory management, and scalability.

AMRs in Healthcare and Medicine

In healthcare settings, AMRs are proving invaluable. These robots are used for tasks such as medication delivery, assisting patient care, and disinfecting hospital rooms. Their precision, reliability, and ability to operate autonomously make them essential tools in modern healthcare facilities.

Challenges and Future Trends

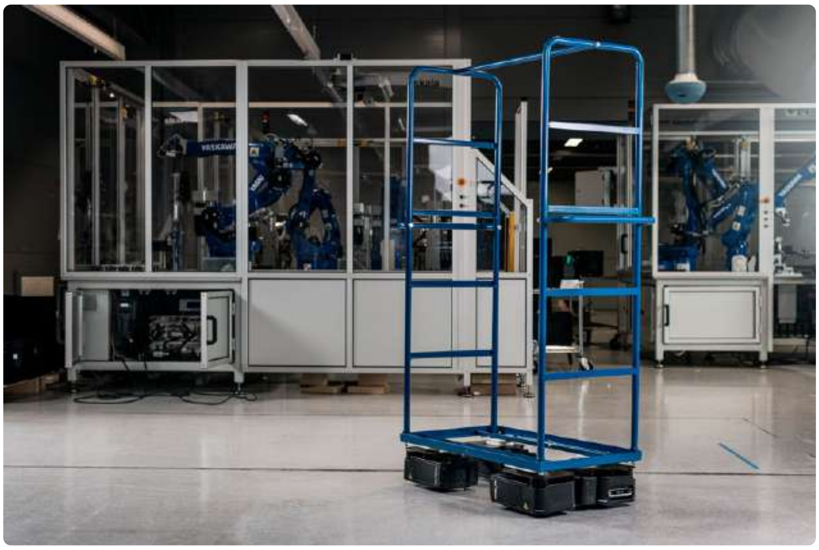
While the potential of AMRs is undeniable, several challenges must be addressed for their widespread adoption. These include seamless integration into existing workflows, ensuring safety, and managing initial costs. Looking ahead, we can anticipate AI advancements, increased interconnectivity among AMRs, and broader adoption as technology matures and costs decrease.

Autonomous mobile robots are at the forefront of automation and are poised to reshape industries profoundly. From manufacturing to healthcare and logistics, the adaptability and efficiency of AMRs make them indispensable tools for modern businesses. Wheel.me [1] offers a game-changing solution by transforming virtually anything into an AMR (**Figure 3**), thereby revolutionizing automation across industries. Their innovative approach provides unmatched flexibility, enabling businesses to adapt existing equipment, such as carts or trolleys, into autonomous systems. This flexibility not only preserves investments, but also streamlines integration. Additionally, wheel.me's disruptive pricing democratizes access to cutting-edge automation technology, making it accessible to a broader range of businesses, from small businesses to established enterprises. What's more, their rapid deployment capabilities ensure that businesses can swiftly implement AMR solutions, realizing efficiency gains and cost savings in a matter of weeks rather than months. In essence, wheel.me's technology offers a trifecta of benefits: adaptability, affordability, and agility — making automation a feasible and transformative option for industries of all sizes.

As the technology continues to evolve and barriers to adoption are overcome, we can expect these robots to play an even more significant role in driving efficiency, reducing costs, and improving safety across diverse sectors. The era of AMRs has arrived, and it promises a future where automation is not just simple, but also incredibly powerful. ◀

230606-01

Figure 3: Make anything into an AMR with wheel.me.



WEB LINK

[1] wheel.me: https://wheel.me

Evolved for More Challenges

Rohde & Schwarz Adds Eight-Channel R&S MXO 5 to Next-Generation Oscilloscopes

Contributed by Rohde & Schwarz

With the all-new R&S MXO 5, Rohde & Schwarz continues to evolve its series of next-generation oscilloscopes, which started with the successful R&S MXO 4 in 2022. The R&S MXO 5 is the company's first eight-channel oscilloscope. It expands upon the industry-firsts pioneered by the R&S MXO 4 and will empower engineers to tackle even more demanding design challenges.

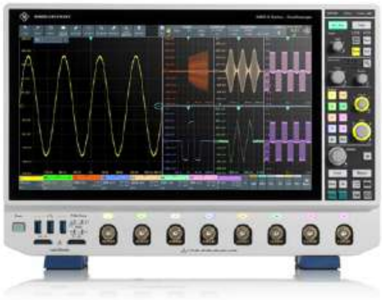


Figure 1: The R&S MXO 5 is the next-generation oscilloscope from Rohde & Schwarz, evolved for more challenges.

Rohde & Schwarz presents its new R&S MXO 5 oscilloscopes (**Figure 1**), available with four or eight channels. Building on next-generation MXO-EP processing ASIC technology (**Figure 2**) developed by Rohde & Schwarz and introduced with the R&S MXO 4, the new eight-channel R&S MXO 5 oscilloscopes take measurement performance to the next level.

Industry Firsts in an Eight-Channel Oscilloscope

The new R&S MXO 5 shows more of a signal's activity in both the time and the frequency domains than any other oscilloscope. The R&S MXO 5 is the world's first eight-channel oscilloscope with 4.5 million acquisitions per second and 18 million waveforms per second across multiple channels. Engineers can capture intricate signal details and infrequent events with exceptional precision. The R&S MXO 5 has digital triggering on all eight channels to surpass its competitors in the accurate isolation of small signal anomalies. The groundbreaking capability of 45,000 FFTs per second provides engineers with unparalleled spectrum signal viewing, particularly for EMI and harmonic testing.

By capturing up to 99% of real-time signal activity with the fastest acquisition capture rate in the world, the R&S MXO 5 speeds up signal analysis while detecting rare and random events missed by most other oscilloscopes. The features let engineers debug designs more efficiently across multiple applications — from power conversion to automotive analysis — making power and signal integrity measurements as well as logic and bus protocol debugging a breeze.

Figure 2: A look inside reveals a custom-made 200 Gbit/s ASIC: the MXO-EP with strong signal processing power.

Philip Diegmann, Vice President of Oscilloscopes at Rohde & Schwarz, says: "Launching the R&S MXO 4 in 2022 introduced the next generation of oscilloscopes with unmatched performance and value. At Rohde & Schwarz, we continue to evolve the usability of our oscilloscopes to new levels. Today, we are excited to introduce the R&S MXO 5, continuing our legacy of oscilloscope excellence. Building on the same engineering breakthroughs, this oscilloscope series was meticulously designed by our dedicated development team to capture both the big picture and smallest details of electrical signals with unparalleled accuracy and speed. Evolved for more challenges, the scopes will speed up our customers' understanding and testing of their electronic systems."

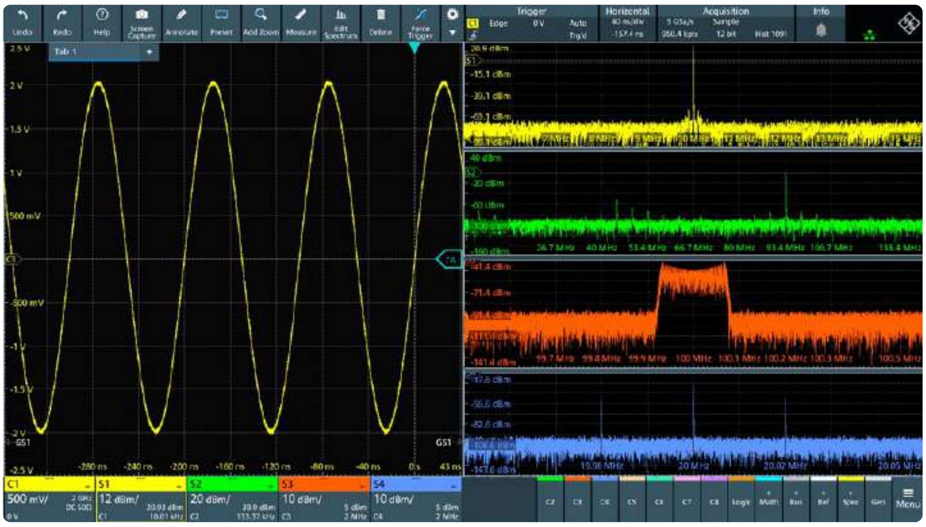


Figure 3: A normal sine tone can conceal many spectral elements. The four simultaneous spectra can be set at different spans and ranges to examine various frequency details.

Deepest-in-Class Standard Memory

The R&S MXO 5 series offers standard simultaneous acquisition memory of 500 Mpoints across all eight channels — or double the standard memory of the competition. The ample storage can be used for extensive data capture. Furthermore, a memory extension option can double the recording length to 1 Gpoints for the most demanding applications. The exceptional memory depth in the R&S MXO 5 oscilloscope is critical for a wide range of troubleshooting tasks, enabling the capture of extended periods of time and retaining accurate bandwidth information even with slower time base settings.

First Digital Trigger for Eight Channels

As the first eight-channel oscilloscope with digital triggering, the R&S MXO 5 sets a new standard in signal analysis. The digital trigger outperforms all other oscilloscope triggers with a remarkable sensitivity of 0.0001 div. The sensitivity helps precisely isolate small physical layer anomalies, even when large signals are present. No other instrument on the market can match this trigger sensitivity. Other instruments often require signal processing corrections to align analog path trigger events, resulting in a slower and noisier trigger performance. The digital trigger seamlessly complements the 18-bit vertical architecture in the oscilloscopes, so that engineers can fully leverage the precision of the R&S MXO 5.

Superior RF Measurement

The R&S MXO 5 excels in RF measurements, both in the time and frequency domains. It is the first oscilloscope with 45,000 FFTs (Fast Fourier Transforms) per second. The exceptional speed, combined with the ability to simultaneously display four different time-independent spectra, provides engineers with outstanding RF signal visibility unmatched in its class (**Figure 3**). The advanced capabilities are standard R&S MXO 5 features.

Evolved User Experience

An unparalleled large 15.6" full-HD capacitive touchscreen and an intuitive user interface optimize the learning curve and give R&S MXO 5 oscilloscopes a seamless and engaging visual experience. The small footprint and first-in-class VESA mounting make the oscilloscopes ideal for any engineering workspace (**Figure 4**). The R&S MXO 5 has the industry's lowest audible noise level, which is quieter than a whisper for a focused lab environment where engineers can perform their tasks precisely.

Low Starting Price

The R&S MXO 5 oscilloscopes are available in four and eight-channel models, in bandwidth ranges with 100 MHz, 200 MHz, 350 MHz, 500 MHz, 1 GHz, and 2 GHz models. The eight-channel models start as low as € 19,500, the industry's lowest entry price in this instrument class. Various upgrade options are available to users with demanding application needs, such as 16 digital channels with a mixed-signal oscilloscope (MSO) option, an integrated dual-channel 100 MHz arbitrary generator, protocol decode and triggering options for industry-standard buses, and a frequency response analyzer to enhance the capabilities of the instrument.

The new R&S MXO 5 series oscilloscopes [1] are now available from Rohde & Schwarz and selected distribution channel partners.

230607-01

Rohde & Schwarz will demonstrate the new R&S MXO 5 series oscilloscopes at productronica 2023 in Munich in hall A1.375, from November 14 to 17, 2023.

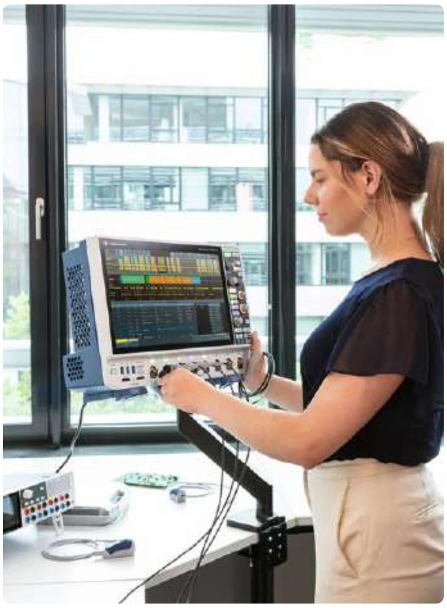


Figure 4: VESA mounting makes the R&S MXO 5 oscilloscopes ideal for any engineering workspace.

WEB LINK

[1] R&S MXO 5: https://rohde-schwarz.com/product/MXO5

Starting Out in Electronics...

...Amplifying Differences

By Eric Bogers (Elektor)

At the end of the previous edition, we briefly described the constant current source and addressed the question of why you actually need a circuit that can deliver a constant current to a variable load. There we also answered this question: without a constant current source, it would be impossible to build a differential amplifier.

The Differential Amplifier

A differential amplifier is at the heart of every operational amplifier (opamp). A differential amplifier has two inputs: an inverting input (IN–) and a non-inverting input (IN+). It amplifies the *difference* between these two inputs. **Figure 1** shows the schematic diagram of a differential amplifier.

The voltage divider consisting of R8 and R9 provides a voltage that is approximately 2.7 V above the negative supply voltage, so the voltage over R7 is approximately 2 V. This results in a current of 2 mA — transistor T3 acts as a constant-current source.

If the voltages on both inputs of the differential amplifier are the same, this 2 mA current is divided equally between the two transistors above the constant-current source (1 mA each). This produces a voltage of 7.5 V over each collector resistor (R1 and R2).

Now, suppose we apply a voltage of a few millivolts to the non-inverting input (IN+), causing the current through T1 to rise to 1.5 mA. The constant current source maintains the total current at 2 mA because that is what it is designed to do, so the current through T2 must drop to 0.5 mA. This means that the voltage over R2 drops to just 3.75 V, so the output voltage of the differential amplifier rises by 3.75 V.

If we had applied the voltage to the inverting input instead of the non-inverting input, the current through T2 would have risen instead of dropping, and the output voltage would have dropped.

But, what happens when we apply the same voltage to *both* inputs? The answer is that the output voltage does not change, because the same current flows through both transistors when the same voltage is applied to both inputs. In other words, a differential amplifier amplifies the *difference* between the voltages on the two inputs, rather than the sum of the voltages.

One of the major advantages of a differential amplifier is its high temperature stability, provided that the temperature rise in both transistors is the same. This is because the increase in the base-emitter voltage due to the rising temperature is the same in both transistors, so they compensate for each other.

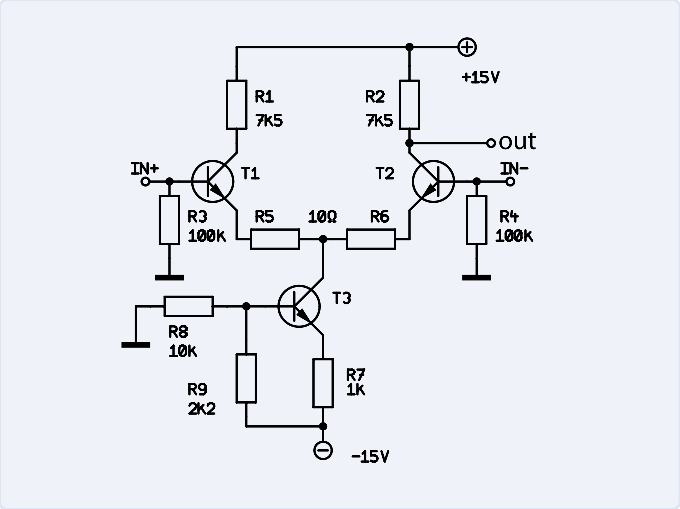


Figure 1: The differential amplifier.

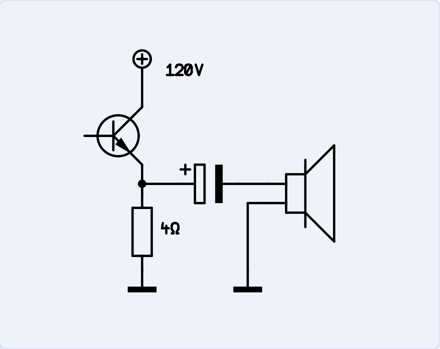


Figure 2: A Class-A output stage.

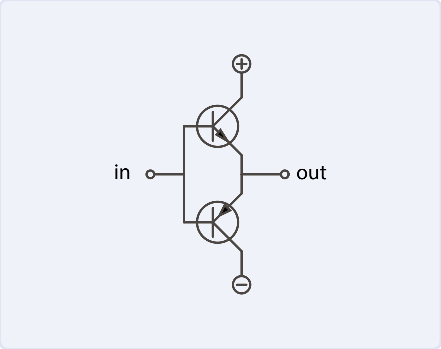


Figure 3: A push-pull output stage.

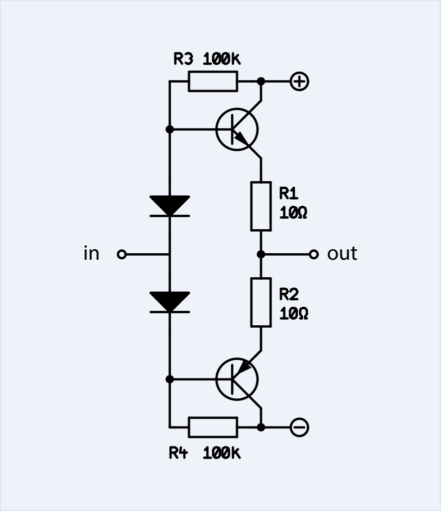


Figure 4: The Class-B amplifier.

The Class-A Amplifier

Suppose you want to build a power amplifier that can deliver 400 W to a 4 Ω load. As you probably already guessed, you can use an emitter follower for this (see **Figure 2**).

A drawback of an emitter follower is that it can *charge* the output capacitor but is not able to *discharge* the output capacitor – which is essential in order to process the negative half-waves of the AC signal, which of course is what is needed. In practice, the capacitor is discharged through the emitter resistor. This means that its resistance should not be higher than the output impedance, which, in this case, is also 4 Ω.

(For the nit-pickers among our readers, here we completely ignore the fact that a loudspeaker is not a resistive load, but instead a fairly complex device with a correspondingly complex impedance. To ensure that a loudspeaker does its job properly, the amplifier should have an output impedance that is as low as possible, so that the speaker cone is suitably damped. The ratio of the load impedance to the output impedance is called the damping factor, and it should be a high as possible. However, it is not possible to achieve a good damping factor with a 4 Ω emitter resistor. For that, the resistance needs to be much lower. But we don't need to go into that level of detail here.)

You can use the following formula to calculate the output voltage:

$$U_{\mathrm{eff}} = \sqrt{P \cdot R} = \sqrt{400\ \mathrm{W} \cdot 4\ \Omega} = 40\ \mathrm{V}$$

This is the RMS value. The peak voltage is correspondingly higher, and, because you have to power both half-waves of the signal from a single supply voltage (in this example), you also have to double the value:

$$U_{\mathrm{tt}} = 2 \cdot \sqrt{2} \cdot U_{\mathrm{eff}} = 113.14\ \mathrm{V}$$

This value is certainly on the tight side. You need a bit of margin, because there's a small voltage drop over the transistor, and the voltage on the filter capacitor will also drop slightly during the half-waves of the output signal. As a rule of thumb, you should add a few volts for this, resulting in a supply voltage of 120 V. (To be clear, here we are only dealing with a paper design, and we don't have to worry about whether it would actually be possible to find a transistor suitable for this example.)

You should set the operating point so that the voltage over the emitter resistor is half the supply voltage. For the power dissipation in both the emitter resistor and the transistor, this means:

$$P = \frac{U^2}{R} = \frac{(60\ \mathrm{V})^2}{4\ \Omega} = 900\ \mathrm{W}$$

In other words, even when the amplifier is not delivering any power to the loudspeaker, you are looking at a total power dissipation of 1800 W. That's quite a lot, and of course it's totally unacceptable.

Push-Pull Output Stage: The Class-C Amplifier

The power dissipation can be reduced considerably by using two transistors, each, in turn, handling the alternating half-waves of the input signal.

In the circuit shown in **Figure 3**, the NPN transistor amplifies the positive half-waves of the signal and the PNP transistor amplifies the negative half-waves. This circuit is called a Class-C amplifier, and, unfortunately, it has a serious shortcoming: Both transistors are cut off when the input signal level is less than the 0.7 V forward voltage of the base-emitter junctions. This causes severe distortion of the amplifier output signal.

The Class-B Amplifier

It's easy to get around this shortcoming of the Class-C amplifier by applying a bias voltage to the base of each transistor. In the circuit shown in **Figure 4**, two diodes are used for this purpose.

The voltage across each of these diodes is equal to the base-emitter forward voltage of the corresponding transistor. A true Class-B amplifier has zero quiescent current, so each transistor only conducts during its respective half-wave of the sinusoidal signal.

However, transistor characteristics are distinctly nonlinear at voltages just above the base-emitter forward voltage, which leads to significant distortion in practice. This can be countered by applying a slightly higher bias voltage to the base of each transistor, resulting in a corresponding quiescent current. A typical value is around 1 percent of the maximum output current. In the present example, we assumed an output current of 10 A, so the quiescent current would be approximately 100 mA.

The quiescent current is usually achieved by connecting a trimpot in series with the two diodes and adjusting it to set the desired quiescent current. The emitter resistors are essential for the temperature stability of the circuit. When the transistors heat up, the base currents (and therefore the collector currents) rise sharply, assuming a constant base-emitter voltage, causing the transistors to get even hotter and further increasing the current. This vicious cycle continues until the transistors fail. For this reason, it is necessary to mount the output transistors and the diodes on the same heat sink, so that the forward voltage of the diodes will drop as the temperature rises. The emitter resistors serve to reduce the base-emitter voltage.

Emitter resistors are also indispensable when several output transistors are connected in parallel, to ensure that the power dissipation is distributed evenly over the transistors so that a greater maximum power can be achieved.

And, speaking of power dissipation, the power dissipation in the quiescent state (without any signal) is determined by the value of the quiescent current. The formula for a single transistor is:

$$P = U \cdot I = 60\text{ V} \cdot 0.1\text{ A} = 6\text{ W}$$

This is truly negligible compared to the power dissipation of a Class-A amplifier, even considering that you have to double the dissipation because you have two transistors in the circuit.

It's easy to calculate the total power dissipation by subtracting the output power from the power fed into the amplifier:

$$P_{\text{in}} = U \cdot I = 60\text{ V} \cdot 10\text{ A} = 600\text{ W}$$

$$P_{\text{loss}} = P_{\text{in}} - P_{\text{out}} + P_{\text{quiescent}} = 600\text{ W} - 400\text{ W} + 12\text{ W} = 212\text{ W}$$

(Note that the value of 60 V used to calculate the input power [instead of 120 V] is actually correct because each of the two transistors only conducts half of the time.)

Unfortunately, the above example calculation does not accurately reflect reality, because here, we chose a very low value for the supply voltage. In this case, the difference between the supply voltage and the maximum output voltage is less than around 3.6 V. Twice this value would be more realistic, at least if we want to avoid making the filter capacitors in the power supply ridiculously large. If we take this into account, the maximum power dissipation increases to 248 W.

Note that when we discuss maximum power dissipation here, we're referring to sinusoidal signals. The absolute maximum power dissipation occurs with a square-wave output signal with an amplitude equal to half the supply voltage. Under this condition, the power dissipation can be as high as 264 W, which is 6 percent more.

Translated by Kenneth Cox — 230533-03

Editor's Note: This series of articles, *Starting Out in Electronics*, is based on the book, *Basiskurs Elektronik*, by Michael Ebner, which was published in German and Dutch by Elektor.

Questions or Comments?

If you have any technical questions or comments about this article, feel free to contact the Elektor editorial team at editor@elektor.com.

Related Products

- **B. Kainka, Basic *Electronics for Beginners* (Elektor, 2020)**
  Book: www.elektor.com/19212
  E-Book: www.elektor.com/19213

Analogue Electronics and Microcontrollers Projects
Basic Electronics for Beginners
Burkhard Kainka
elektor

wheel.me Genius 2

wheel.me invented the world's first autonomous wheel that turns anything into an autonomous robot effortlessly.

Maximal Flexibility

Endless use cases, no change in infrastructures required

Robotics-as-a-Service

All-in-one package, no large upfront cost, flexible terms

Disruptive pricing

Low-risk, low price and immediate return on investment

Easy navigation

Dynamic obstacle detection, omnidirectional movement and focus on safety

www.wheel.me

wheel_me

From Our Partners At
Elettronica In
ELETTRONICA APPLICATA
elettronicain.it

Mini Reflow Plate

For Assembling or Repairing Small SMD Circuits

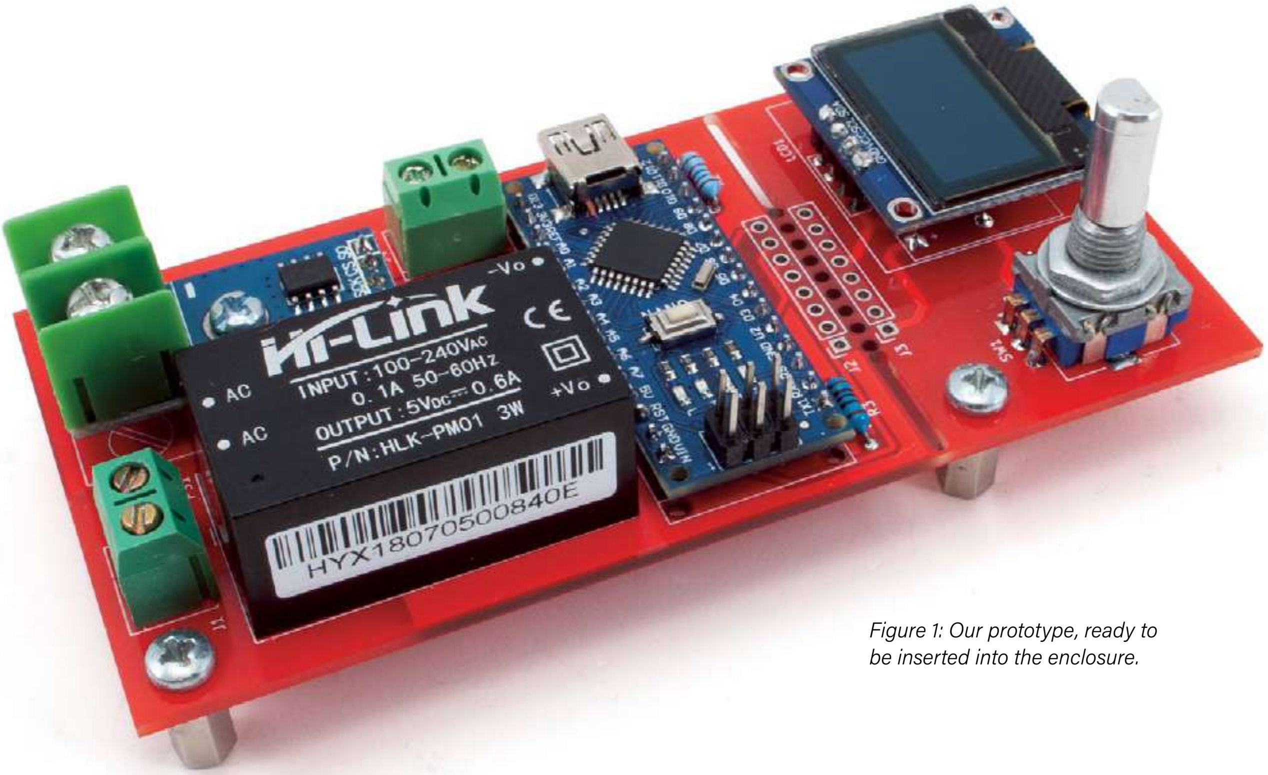


Figure 1: Our prototype, ready to be inserted into the enclosure.

By Massimo Divito (Italy)

The transition to SMD miniaturized components marked, for many, the beginning of a series of assembly difficulties to be overcome. In addition to the not-so-easy placement of components, specific temperatures and timing must be adhered to for the soldering cycle. Professional reflow soldering stations are expensive. In this article, we present an effective yet reasonably priced solution for assembling/repairing our boards.

When compared to through-hole technology (THT) components, working with surface-mount technology (SMT) components, also known as SMDs, can be quite demanding. It requires keen eyesight, a very steady hand, a soldering iron with an appropriate tip, and perhaps most importantly, a great deal of patience.

When the presence of surface-mount components in a circuit is prevalent and, especially, when chips with many pins are involved, the situation becomes more complicated, all the more so if there are many elements close together; using a soldering iron can become impractical in this case. For this reason, we thought of a design that could meet your needs in such situations: Something simple and inexpensive that could be easily implemented.

The soldering technique that is used in the professional field for SMDs is called reflow soldering, and it normally uses ovens. Instead, we thought of a heated plate, suitable for smaller circuits, that takes up little space in our laboratory and that can be used as a preheater for soldering and desoldering operations with the hot air station, if necessary.

Reflow soldering is a process in which a solder paste, consisting of powdered tin and liquid flux, is used instead of the normal tin/lead alloy (or lead-free type for ROHS-compliant applications) to solder components onto the printed circuit board.

Solder paste is deposited on the pads of the printed circuit board using either a syringe or a special stencil, after which all the electronic components are first placed, and finally the entire assembly undergoes controlled heat treatment, which melts the alloy and permanently solders the components.

How It Works

Our design consists of an aluminum plate, containing electrical resistors operating at 230 V that heat it; the printed circuit board to be soldered is placed on this plate. The resistors are operated by a solid-state relay that is controlled, in turn, by an Arduino Nano-based circuit, as visible in the fully-assembled board shown in **Figure 1.** Connected to the circuit is a K-type thermocouple with a clamping ring, screwed to the plate, which constantly senses the temperature.

This is controlled by a rotary encoder that allows the mode of operation to be selected from two different profiles: one for tin alloys with lead and one for lead-free alloys. The third mode allows the plate to be set to a fixed temperature; this feature is useful, for example, to preheat a circuit on which we are going to work with a hot-air stylus. We have then added a small 0.96-inch OLED display on which we will select the profile we want to start and, during the reflow phases, it will show us the set temperature and the temperature reached by the plate, as well as the time left until the process is completed. As we have already mentioned, to carry out the soldering process correctly, it is necessary to follow a precise temperature curve to allow the solder paste to melt properly and at the same time prevent the components from being damaged by too much heat; as a rule, each manufacturer specifies, for its own integrals, the optimal curve.

The entire process is divided into four stages:

- Preheat: The solder paste aggregants begin to evaporate.
- Soak: The aggregants complete evaporation, the flux melts and deoxidizes the pads.
- Reflow: The metal part of the solder melts and creates a permanent mechanical and electrical connection between the components and the circuit board. The flux evaporates.
- Cooling: The tin cools by solidifying.

To achieve this, it is necessary to know, at each instant of the remelting process, the temperature that the plate should have. Since the plate (**Figure 2**) temperature is read approximately every 250 ms, for a single profile lasting 380 s we need to know 1520 temperature values (380,000 / 250). Storing and accessing so many values becomes quite complicated at the firmware level, and any future changes would be cumbersome. Furthermore, considering that, in our case, there are two profiles (but we could add more if we wanted), the number of values doubles. That's why we arrived at an alternative solution, thanks to which we can define only some of the temperatures and, through them, calculate the others.

If we look at the graphs of the profiles, visible in **Figure 3**, we notice that the temperature changes, both up and down, are linear. We can then divide each profile into several segments, of which we need to know only the extremes. Each segment is defined by two pairs of values that indicate its beginning and end, and each pair is composed of the values of seconds and temperature, respectively; these values are easily stored in a two-dimensional array consisting of six elements: the six peaks that describe the profile precisely.

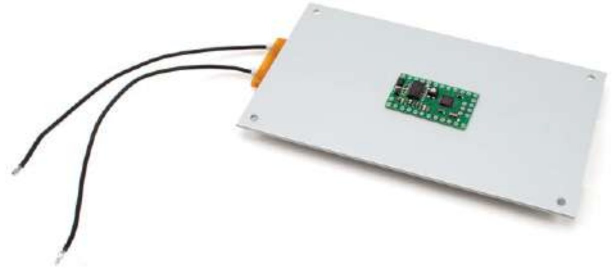
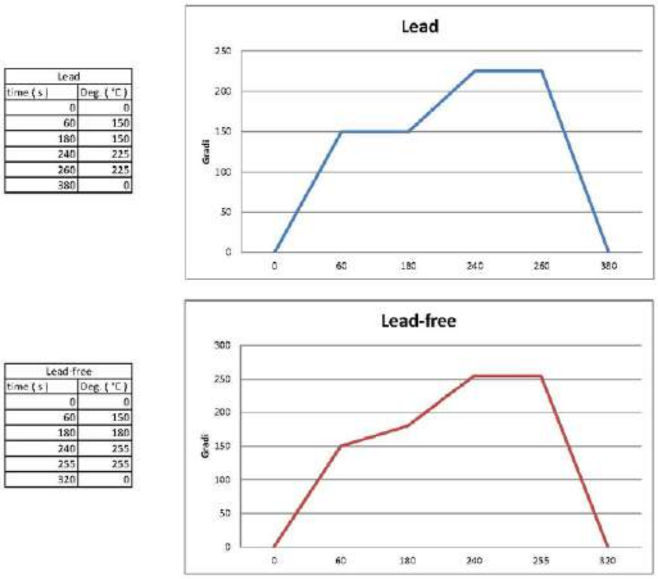


Figure 2: For soldering, a board can be placed on the heated plate, after sprinkling the pads with solder paste.



Lead	
time (s)	Deg. (°C)
0	0
60	150
180	150
240	225
260	225
380	0

Lead-free	
time (s)	Deg. (°C)
0	0
60	150
180	180
240	255
255	255
320	0

Figure 3: Temperature curves for soldering using leaded and lead-free alloys.

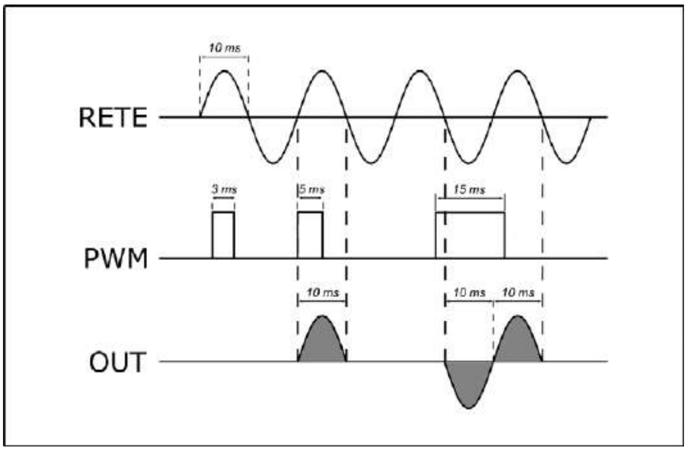


Figure 4: Principle of operation of the PWM-driven zero-cross trigger method (RETE = mains voltage).

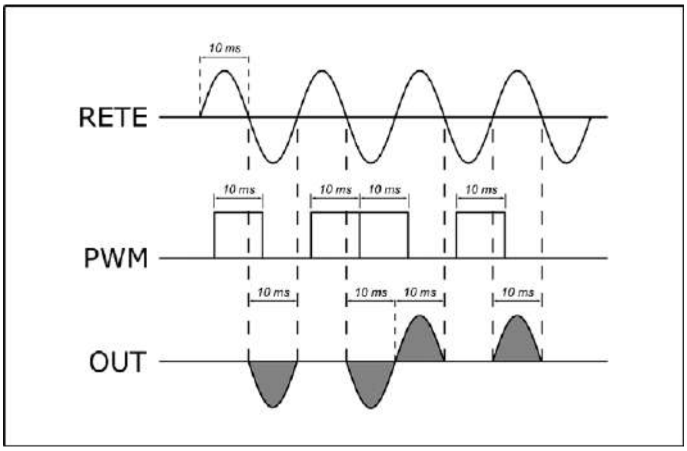


Figure 5: Behavior of the zero-cross trigger method driven with PWM at 1 Hz (RETE = mains voltage).

Thus, if we want to know the temperature value to be set on the plate at a given time, all we have to do is apply the linear interpolation method, which allows us to extrapolate an unknown value between two known values. Knowing the temperature value the plate should have and its actual temperature, we can adjust its "on" or "off" accordingly. To make the temperature follow more precisely what is indicated by the profile, we will use PID control technique. This type of control simply adjusts a quantity by constantly monitoring it and acting on certain variables, trying to keep it close to the set value. Clearly, operation is not so trivial, in fact, it follows precise mathematical formulae. Fortunately, there are special libraries for Arduino that this much easier.

As we have already mentioned, to turn on the plate we use a solid-state relay that has a feature called the "zero-cross trigger" method, which causes the change of state of its output — that is, the transition between "on" and "off" and vice versa — to occur only when the mains voltage wave, which we remember is 50 Hz, passes through 0 V. This means that from the time we energize or de-energize the input of the relay to the time it turns the load "on" or "off," a few milliseconds may elapse.

Clearly, this peculiarity does not allow us to have an exact control of the output. In fact, driving the relay with a pulse of the duration of less than that of a half-wave, which we've perhaps sent a millisecond after the zero crossing, will result in failing to turn on the load. If, on the other hand, we send the pulse at the exact zero-transition point, we'll turn the load on, but its switching off will only occur at the next zero crossing. Taking **Figure 4** as an example, following a PWM that drove the relay for a total of 23 ms, we end up with the output active for a total of 30 ms. One solution would be to use a zero-cross detector circuit which, as the name implies, can detect all the times the grid voltage passes through zero, to properly synchronize the PWM output. In our case, in order not to complicate the circuit further, we solved this problem by using a very low PWM frequency, more precisely 1 Hz. In this way, we will have a total period of 1 second, or better, 1,000 milliseconds, divided into 100 steps of 10 milliseconds each. In practice, for every one percent change in duty-cycle, we will have a change in pulse length of 10 ms, which is exactly the duration of a half-wave of the network frequency. In this way, the duty-cycle value that we set will be correctly passed on to the output of the relay, since we are certain that the same delay applied to turning the load on will also be applied to its deactivation, the only difference being a possible delay ranging from 1 to 9 milliseconds, which will shift the entire wave forward (**Figure 5**). This design provides only one mode to regulate heating, whereas cooling occurs naturally by heat dissipation. Thus, the actual curve will not be the same as the theoretical one. Much depends on the environment in which we operate, but we can assure you that, after several trials, we have never had any problems in completing the reflow process with excellent results.

Wiring Diagram

Thanks to the use of Arduino (U1) the circuit (**Figure 6**) is simple; in fact, besides it, there is only a module with a MAX6675 on board. This chip is used to amplify the signal present on the terminals of the K-type thermocouple, convert it from analog to digital and return it via SPI interface. In fact, reading this signal via the analog inputs would be more complicated, since the amplitudes involved are in the microvolt range and because of this, would have to be properly amplified, and electrical noise could affect the measurement. For this reason, we decided to use this chip and, more precisely, a pre-assembled module (BRD1) that includes it, since the MAX6675 is produced only in SMD format!

The signals from the thermocouple are properly converted and sent through the SCK, CS, and SO pins connected to digital inputs of U1, which processes them and decides, accordingly, whether to increase or decrease the temperature of the plate, whose heating element is powered through a solid-state relay. In our case, we used a Fotek SSR-25-DA — housed in the same enclosure and not visible in the schematics — that supports loads up to 25 A. The driving of this relay is via a digital output through resistor R1, as is usually done to turn on an LED. In our case, we're going to drive a built-in opto-isolator, located in the input section.

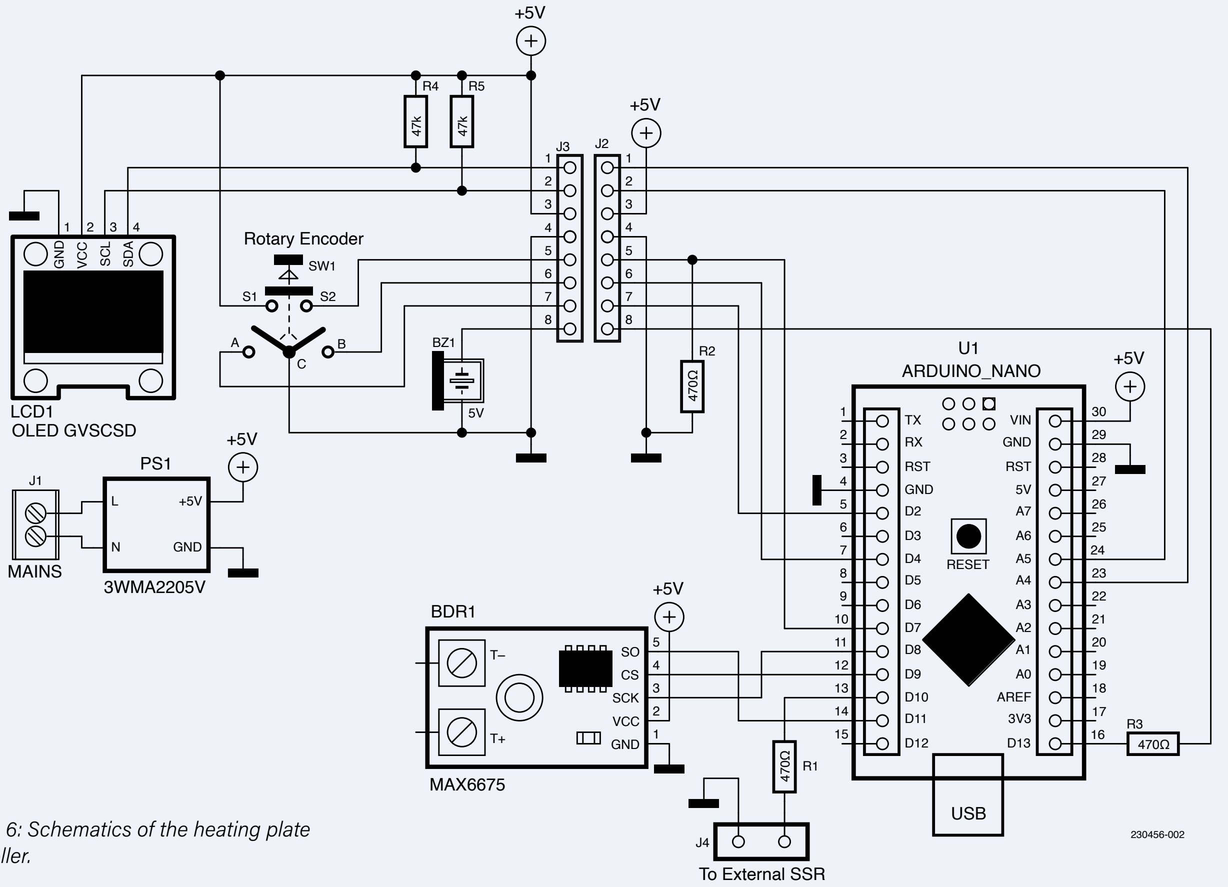
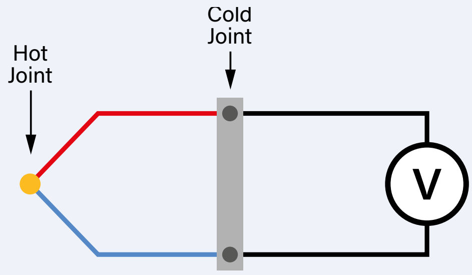


Figure 6: Schematics of the heating plate controller.

The Thermocouple



Thermocouples are temperature transducers, commonly used in multiple fields: industrial, scientific, engineering, etc. Because of their small size and fast response time, they can be used even in particularly critical environments, being able to accurately measure an extended range of temperatures. Their big advantages are that they are very simple, durable, and inexpensive.

Thermocouples are designed for measuring the high temperatures of physical bodies, components, and substances.

Each thermocouple consists of two different metal parts joined at a point called the "hot junction." This point is brought into contact with the object or substance whose temperature is to be measured. The other end, the one formed by the two free wires, is called the "cold junction," and, if it is connected to a voltmeter, the temperature difference between the two junctions creates a potential difference proportional to the temperature (Seebeck effect). Thermocouples can also look very different from each other on the outside, but basically all have the same principle of operation. The cold junction originates from the difference between the metal of the thermocouple wires and that of the connecting wires.

There are different types of thermocouples, but the type-K is definitely the most widely used in industry. It is made of a combination of nickel-based wires and the measuring ranges are -200 to +1,260 degrees Celcius. Furthermore, the corrosion and oxidation resistance of nickel allows it to be used in a wide range of applications.

The choice to use such a component is dictated by the fact that the load applied to it is driven in PWM, that is, through a sequence of pulses that turns the plate on and off multiple times per second. An ordinary electromechanical relay, besides not having the required switching speed, would not last long and its contacts would be irreparably damaged by electric arcing.

On the SDA and SCL pins of the Arduino, we connected LCD1, a small one-inch display based on the SSD1306 chip, adding pull-up resistors R4 and R5. Furthermore, we have the rotary encoder in quadrature SW1, with its pins A and B connected to digital inputs of U1: As you can see from the schematic, we did not provide pull-up resistors, but enabled the ones located inside the microcontroller via software; pin C is connected to GND instead. The built-in pushbutton of the encoder shaft, which allows the various selections, is connected on one side to Vcc and on the other to pin 10 of U1, together with pull-down resistor R2.

Finally, there is the buzzer, BZ1, which is connected to pin 9 of U1 via resistor R3. The entire circuit is powered by the PS1 module, an AC/DC converter from Hi Link that mounts directly on the circuit board. This module has a 230 V AC input and a 5 V, 600 mA output. This solution saves us from using external power supplies, by facilitating the housing of the entire circuit in a single enclosure.

Firmware

The Arduino sketch, downloadable at [1], is quite meaty, so here we comment only on some parts of it, which are the ones that are directly relevant to understand and which allow you to customize the project to your liking. As a first step, the four libraries we use:

- Display (*U8g2* by Oliver)
- MAX6675 (*MAX6675_Thermocouple* by Yurii Salimo)
- PID (*PID* by Brett Beauregard)
- PWM (*https://github.com/maxint-rd/FastPwmPin*)

Next, the pins connected to the MAX6675 and the rotary encoder are defined, as are two custom graphic symbols that will compose the thermometer icon on the display. Notice that in the initial declarations the `profiles` array, which, as we mentioned, defines the two solder profiles:

```
int profiles [2][6][2] = {
{{0, 0}, {60, 150}, {180, 150}, {240, 225},
    {260, 225}, {380, 0}}, // Lead profile
{{0, 0}, {60, 150}, {180, 180}, {240, 255},
    {255, 255}, {320, 0}} // Lead-free profile
};
```

Here, you will need to act if you want to change or add a profile, modifying the array values appropriately and remembering that, for each pair of values, the first refers to time in milliseconds and the second to degrees; the tables in Figure 3 will clarify the concept. After this array is another block of variables that you may find useful to modify:

```
int tempMinPreheater = 60;
  //set the minimum temperature that
    // can be set as a preheater
int tempMaxPreheater = 250;
  //set the maximum temperature
  // that can be set as the preheater
int tempStepPreheater = 5;
  // set the value of the setting
  // steps for the preheater
int tempAmbient = 50;
  //set the maximum ambient temperature,
  // if the plate is warmer, the
  // remelting process cannot be started
int tempAdjust = 0;
  //correction of the
  // temperature read by the probe
```

All these code lines are integrated with comments, so it is easy to see that the first two indicate the minimum and maximum values that can be set; the third indicates the value that will be added to or subtracted from the set degrees for each rotation of the encoder; these values refer only to the preheater functionality. The variable `tempAmbiente` indicates the maximum value the plate can have before starting the remelting process and is referred to as room temperature, because the plate usually assumes the temperature of the environment in which it is located. This expedient avoids starting onesoldering process immediately after finishing another. In this case, in fact, it would distort the reflow curve with the risk of thwarting the whole procedure. The `tempAdjust` variable is used to correct the plate temperature reading.

The rest of the sketch needs little comment. The reading of the buttons and encoder is done without using interrupts: to read the long press of P1 and to periodically update the display, a few timers are used instead. In the code that handles the reflow functionality, we can see how we use the linear interpolation formula to calculate at what temperature to set the plate:

```
reflowNowTime = (unsigned long)(millis() - reflowStartTime);
  // Calculates how many milliseconds
  //the reflow cycle started;
for (int i = 0; i < 6; i++) {
 if ((reflowNowTime >=
    (profiles[profile][i][0] * second)) &&
     ( reflowNowTime <
       (profiles[profile][i + 1][0] * second)) {
  xa = profiles[profile][i][0] * second;
  xb = profiles[profile][i + 1][0] * second;
  ya = profiles[profile][i][1];
  yb = profiles[profile][i + 1][1];
  tempTarget = ((((reflowNowTime - xa) *
         (yb - ya)) / (xb - xa)) + ya;
  // y0=( ( (x0-xa)*(yb-ya) ) / (xb-xa) ) + ya
  // Linear interpolation formula

  phaseActual = i;
  ruleHeater();
 }
}
```

In the `for` loop we scroll through the array of profiles to get the known values, on the *x*-axis, milliseconds, and on the *y*-axis, degrees centigrade. In this way, we have the values `xa`, `ya`, `xb`, `yb`, while `x0` will be equal to the milliseconds that have passed since the start of the reflow phase: With these values, we can calculate `y0`, which represents the temperature value to be set for our heater plate.

Practical Realization

A special printed circuit board has been designed for the controller, which facilitates its assembly and makes it compact and neat. This printed circuit board gives us the ability to detach the encoder and display part to allow its installation in a wide variety of configurations; in case you decide to split the PCB, you will obviously have

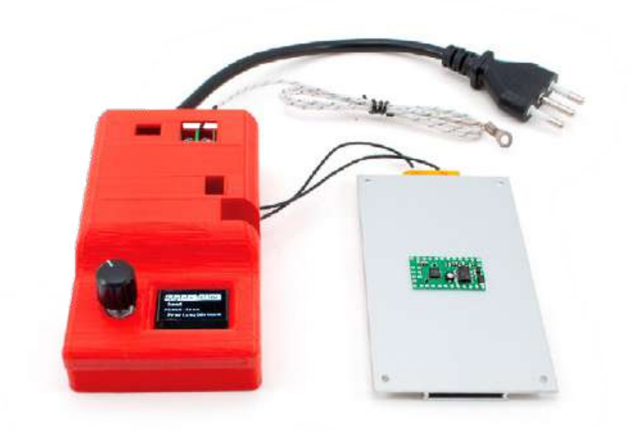


Figure 7: The complete system, including the thermocouple, which is screwed in place using the eyelet terminal lug.

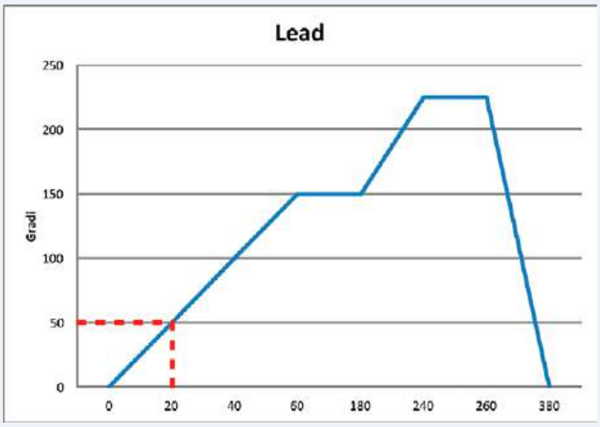
to connect the two parts via connectors J2 and J3. The optimal placement of the circuit is in a plastic enclosure that allows total isolation. One solution might be to use a junction box for electrical installations, with the lid properly drilled for the display and rotary encoder.

Given then the presence of the mains voltage, one might consider boxing along with the circuit the solid-state relay and the 230-volt outlet for the plate connection and the K-type thermocouple probe. To allow an accurate temperature reading, it is important to position the latter correctly, perhaps anchoring it solidly to the heating plate by some metal bracket: One solution might be to insert it into an aluminum eyelet terminal and then tighten it with pliers without exaggerating the pressure to avoid damages. In this way, the probe (**Figure 7**) can be anchored to the plate with a simple screw, perhaps adding some thermo-conductive paste to the mechanical connection, such as that used for PC processors, which aids thermal conduction and improves reading accuracy.

Another solution would be to leave it "loose" and stick it to the PCB you are soldering, each time you perform a reflow process, using a piece of kapton tape.

Warning: Place extreme care in the fabrication and wiring, given the presence of mains voltage, which is very dangerous: Remember to connect the ground wire to the plug and to all metal parts of the device, plate included.

Linear Interpolation



In mathematics, interpolation refers to a method of finding new points in the Cartesian plane from a finite set of data points. It thus makes it possible to derive, in an approximate way, the trend of a curve by knowing only a few points of it. In linear interpolation, which is also the simplest method of interpolation, one needs only two adjacent points, identified by (xa, ya) and (xb, yb), joining them to draw a segment.

Therefore, to calculate the point (x0, y0), intermediate to the known points, we use the formula:

$$y0=\frac{(x0-xa)(yb-ya)}{xb-xa}+ya$$

Let us give an example by referring to the Figure and substituting for the formula the points of the first segment namely (0,0) and (60,150), assuming that we want to know the value of y0 given that x0 = 20

$$y0=\frac{(20-0)(150-0)}{60-0}+0$$

$$y0=\frac{20*150}{60}$$

$$y0=\frac{3000}{60}$$

$$y0=50$$

Calibration and Use

Once the Arduino has been programmed and the project assembled, after carefully checking the wiring (especially the 230 V wiring), we can turn on the circuit and verify its proper operation. Before using it, however, it is convenient to perform a simple, but useful calibration. In practice, depending on where we place the K-type probe, we may have a slight variation in readings; this is due to the arrangement of the heating resistors, which are usually centered, and it is precisely at these that we will get a more accurate reading. As we move, the readings will have small differences that will increase as the distance from the heat source increases.

To correct the reading, we have provided the `tempAdjust` variable, which, by default, is 0. If you have a way to read the temperature of the plate through an infrared or other thermometer, you will be able to verify the temperature read by our circuit and that taken with the thermometer. You will have to assign the value resulting from the difference between the two to the above variable. In this way, you will be able to correct the reading to a value closer to the real one. Once this is done, you will be ready to solder your circuits. We do recommend, however, that you start with small board sizes, for example small SOIC-DIL adapters — you will have to become familiar with the amount of solder paste to deposit on the tracks and the correct positioning of the components.

Practical Use

In addition to being used for reflow soldering, this design can also function as a PCB preheater in soldering and desoldering with the hot air station. In the latter case, the PCB should be brought to about 150 °C, when the components to be soldered are heated with the hot air station, melting flux and solder. ⏮

230456-01

Questions or Comments?

Do you have questions or comments to submit about this article? Please write to the Elektor Editorial Team at editor@elektor.com.

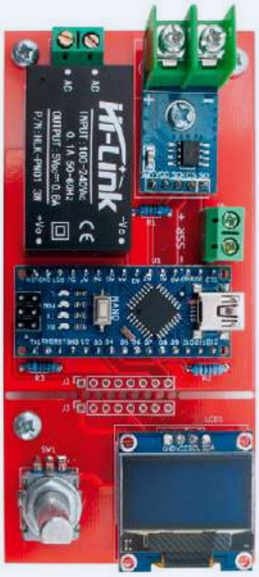
Related Products

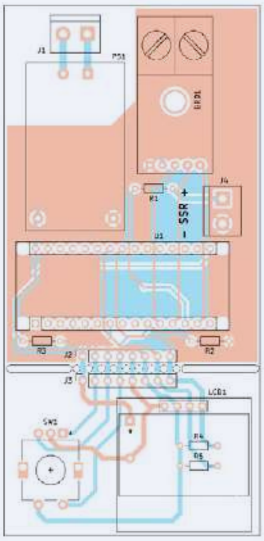
- **Upgraded T-962 Infrared Reflow Oven (Elektor Version) (SKU 20346)**
  www.elektor.com/20346
- **2-in-1 SMD Hot Air Rework Station ZD-8922 (SKU 20141)**
  www.elektor.com/20141

WEB LINKS

[1] Software Download: https://elektormagazine.com/230456-01

Component List





Resistors
R1...R3 = 470 Ω
R2 = 10 kΩ
R4...R5 = 47 kΩ

Modules
U1 = Arduino Nano
PS1 = 3WMA2205V (PSU Module)
BRD1 = MODMAX6675 (for K-type thermocouples)
LCD1 = OLEDGVSCSD (0.96" OLED)

Miscellaneous
BZ1 = 5 V buzzer (without oscillator)
Header strip, female, 4 pins
Header strip, female, 15 pins
Header strip, female, 5 pins
Screw terminal block, 2 poles
Rotary encoder with pushbutton

Join our Community

www.elektormagazine.com/community

elektor
design > share > earn

Q&A With Alberto Savoia

Founder & Innovation Agitator, Pretotype Labs LLC

Don't Start with a Prototype — Start with a Pretotype!

Check That a Market Exists for Your Product Before Warming Your Soldering Iron

By Stuart Cording (Elektor)

We technical types wield a powerful tool. Thanks to our understanding of electronics and software, it doesn't take much to dream up a high-tech solution to the problems around us. And, from there, it's just a short hop to founding a business and world domination. What could go wrong? Well, quite a lot.

Most sources quote that 20% of new businesses fail by the end of their first year, and 50% have folded by year five [1]. Reasons for failure range from legal challenges and burnout to losing focus and poor product. But none of these reasons make the top five. Incredibly, after a lot of work and money spent, 42% of startup post-mortems blame there simply being no market need [2].

So, why do startups get this fundamental element of their business plan so wrong? One person who has dedicated much time to this question is Alberto Savoia. As a former employee of both Sun Microsystems and Google, he has seen how startups achieve success, building a $100-million company with $3 million of funding. But, he also folded a startup that received $25 million of venture capital cash. Alberto spoke to Elektor about product success and failure, his passion for soldering, and why everyone should try pretotyping.

Stuart Cording: Looking at your profile, one would say you are eminently qualified to build a successful startup. However, your third startup failed — why?

Alberto Savoia: Well, I felt horrible. Over the five years, we hired almost 100 people who committed to us and had stock options. So, when it didn't work out, yeah, I felt bad for everybody. And that is what motivated me to look at and answer the question. We hired the best people, right. We had the best venture capitalists (VC) in Silicon Valley, raising $25 million, so we had plenty of money. People told us the product would work. And we'd built it.

About Alberto Savoia

As Google's first Director of Engineering, Alberto Savoia led the team that launched Google AdWords. Later, in his role as Google's *Innovation Agitator*, he developed pretotyping — a rigorous data-based system for validating new product ideas. Prior to Google, Alberto was a successful serial entrepreneur and the first Director of Software Technology Research at Sun Microsystems Laboratories, where he played a key role in the development of Java technology and tools.

Alberto's work on innovation has garnered significant industry recognition and awards, including *The Wall Street Journal Technical Innovation Award*, *InfoWorld's Top 25 CTOs Award*, and *InfoWorld's Technology of the Year Award*. Today, he is highly sought-after as a speaker, teacher, and coach, renowned for his ability to help organizations unlock their full potential for innovation and growth. From his inaugural *Entrepreneurial Innovation* workshop at Stanford in 2011, to his current coaching of Fortune 500 companies, Alberto has made a deep and lasting impact on the product management and innovation practices of many leading companies worldwide. His latest book, *The Right It—Why So Many Ideas Fail and How to Make Sure Yours Succeed*, was published by HarperCollins in 2019.

But it had failed. What did we do wrong? It turned out that we built what I call the wrong "it." It didn't matter how well you built it or marketed it; it wouldn't succeed in the market. So, it led me to think, how can I find out if a product is going to be the right "it?"

Stuart Cording: Most books recommend two things to create a successful startup — a solid business plan and a great team. Why is this not enough?

Alberto Savoia: I've written about a dozen business plans and viewed a few hundred while helping VCs with due diligence. Now, let me tell you that most business plans are a work of fiction. I teach people to redact business plans. Take three or four different color markers. Anything that is an assumption, mark with the red highlighter. Anything that is belief or hope, use a yellow marker, and so on.

So, once you eliminate everything that is irrelevant, like how some other company succeeded, you're left with at most half a page worth of material. The rest is fiction — especially the five-year financial projection!

Stuart Cording: So, how can you tell if your business idea is a winner?

Alberto Savoia: This is why I came up with the definition "The Right It," a product that, if competently executed, will succeed. This means the wrong it, no matter how good your execution, will fail in the market. So that's the first thing.

Now, the next question is, is my product the right it? The first thing I do is describe it in very precise language. Your readers are probably all engineers, so they won't say, "I'm going to use a big resistor." That's not enough. How many ohms will it be? How many watts will it need to handle?

So I start with an assumption, a clear hypothesis, and then test that hypothesis. We're dealing with human beings and markets, so there is no 0% or 100% chance of success. Instead, I work in terms of probability of success, mapping our chances into five buckets of 10%, 30%, 50%, 70%, and 90%. I run experiments, see if the hypothesis is confirmed, then make changes to see if we can improve our chances of success.

Stuart Cording: So we're talking about first building a prototype or minimum viable product (MVP) that, by definition, takes time and costs (lots of) money, correct? Or is there a better way?

Alberto Savoia: Don't get me wrong, I'm super happy in front of a soldering iron or a compiler. I like to build stuff, right? What I don't like is building stuff that fails when I could have been building stuff that succeeds.

To solve this, I invented a new term called "pretotyping," a simplified version of pretend prototyping. For example, I came up with an idea for a tone control, "The Delicious One," for audiophiles, because there is a trend to make pure amplifiers that don't modify the input signal. Eventually, I built this, but it took months to get it right. So, to ensure it was the right it before I started, I pretotyped it.

I knew what I wanted it to look like, so I made a big black box with inputs, outputs, and a nice knob. There was nothing inside it, but I used it to make photos and create a website that you can still visit called *deliciousdecibels.com*. I promoted it on audiophile websites in the US, and, if someone emailed me, I told them that the device was still in development and would cost $300. So, now they have some skin in the game. Later, I shared the tone control curves with them, created a video, and wrote again, saying they could pre-order it for $50 if they wanted to. More skin in the game came with people placing pre-orders.

So, that's how I gathered my own data about the market for my tone control. Not information about the success of past devices, known as other people's data or OPD. Gather your own data, or YODA, in that pretotyping stage.

Stuart Cording: In your book, *The Right It*, there are some clever pretotyping approaches, but some suggestions, such as faking "The Delicious One," possibly cross ethical boundaries for some. How have you seen practitioners of the approach tackle this?

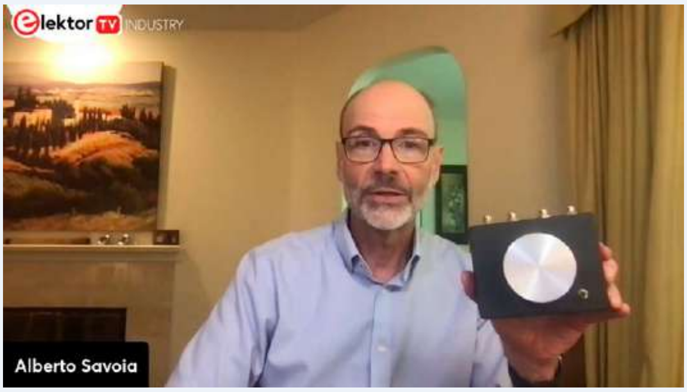


Figure 1: Pretotype of "The Delicious One," using the fake door technique.

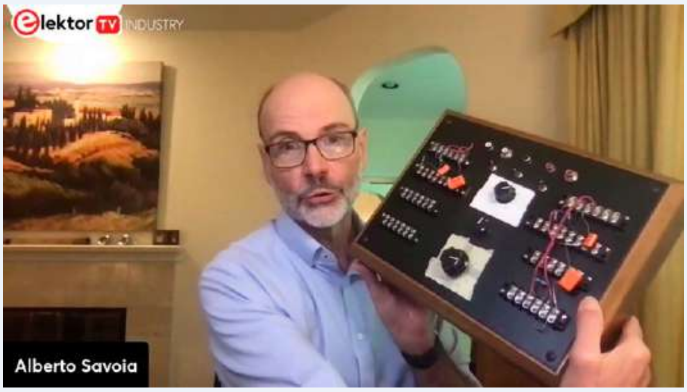


Figure 2: Prototype of the tone control, developed once enough "own data" as to its viability had been collected.

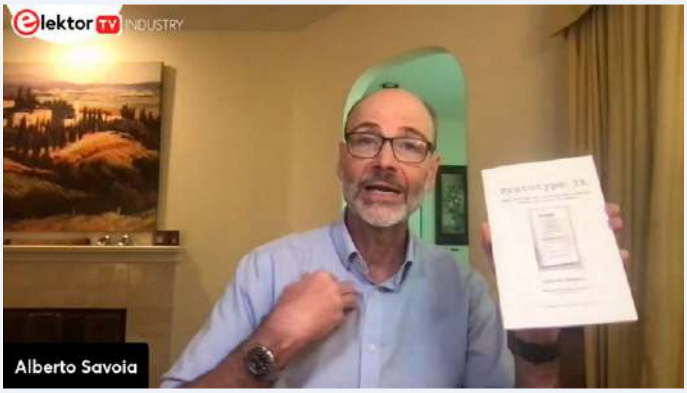


Figure 3: Before committing to writing a book on pretotyping techniques, Alberto pretotyped the book. Gathering interest from colleagues showed there was a market for it.

Figure 4: Based upon the success of the pretotype, Alberto received an advance to write "The Right It," which took nine months.

Alberto Savoia: The approach taken with the tone control is known as "the fake door." So you don't have the product; you just pretend that you have the product. Now, it's not unethical unless you collect money from investors or people buy it, and you don't deliver the product. In fact, I could not teach these approaches at Stanford [University] if they did not pass an ethical review.

Entrepreneurs and innovators are some of our most valuable resources, if not the most valuable. So, the biggest ethical problem, in my opinion, is people not teaching entrepreneurs to use their resources to build products the world actually needs. When you do pretotyping, you're running small tests on a small sample of people. With the fake door, the worst thing that can happen is you say, "I'm sorry, the product is not available, but thank you for expressing your interest." By them clicking a button on your website, you've got a vote that they want the product.

Stuart Cording: Hand on heart, Alberto — have you ever seen anyone find "The Right It" and, despite funding and a competent team, still have their startup fail?

Alberto Savoia: Oh, yeah, of course. The idea can be the right it. But as I write in the book, it is no guarantee that you are the one that is going to succeed, right? So what happens? You determine there is demand for your hardware. You build your box; you do everything right. And then some people develop a software app that does the same thing on a smartphone at a fraction of the price, right? In fact, many early apps failed as Apple and Android integrated their functionality directly into the operating system. So, yes, you can fail even if you have the right it. But that becomes a matter of execution, which is not my domain of expertise.

So, again, with your YODA proving your hypothesis, it's pretty much guaranteed that it will succeed if it's viable. I love electronic stuff, and I think Elektor is a great magazine. So I say to your readers, apply the same rigor and experimental attitude to testing the market as you do with testing and designing your product.

Unfortunately, there are no guarantees in life, but there are avoidable failures, right? What hurts the most is spending $5 million building it, then launching it, and it fails. I get emails like that every week, saying, "I wish I'd read your book before." Because that five-year failure experience could have been avoided in five days by learning that nobody wanted to buy your product.

elektor TV
Watch this on Elektor TV - Industry

The full interview with Alberto is available on the Elektor TV — Industry YouTube channel in Episode 27 of Elektor Engineering Insights (https://youtu.be/-wTUKiTfF5s). And, if you'd like to learn more about YODA, testing your market hypothesis, and pretotyping techniques, take a look at The Right It — Why So Many Ideas Fail and How to Make Sure Yours Succeed *by Alberto Savoia, published by Harper One (ISBN 978-0-06-288465-7) [3].*

230484-01

About Stuart Cording

Stuart Cording is an engineer and journalist with more than 25 years of experience in the electronics industry. You can find many of his recent Elektor articles at *www.elektormagazine.com/cording*. In addition to writing for Elektor, he hosts a regularly livestreamed interview show, *Elektor Engineering Insights* (*www.elektormagazine.com/eei*).

WEB LINKS

[1] Startup Failure Rate: https://failory.com/blog/startup-failure-rate
[2] No Market Need — Final Evaluation Report [PDF]: https://tinyurl.com/nomarketneed
[3] "The Right It" by Alberto Savoia: https://albertosavoia.com/therightit.html

2023: An AI Odyssey

Getting Help Designing a Physical Project

By Brian Tristam Williams (Elektor)

Having delved into the capabilities of ChatGPT previously, especially our chat about discrete mathematics, I thought more and more about the Tower of Hanoi problem, developing a nice, unhealthy obsession, and decided to embark on an ambitious physical project with the assistance of our AI friend.

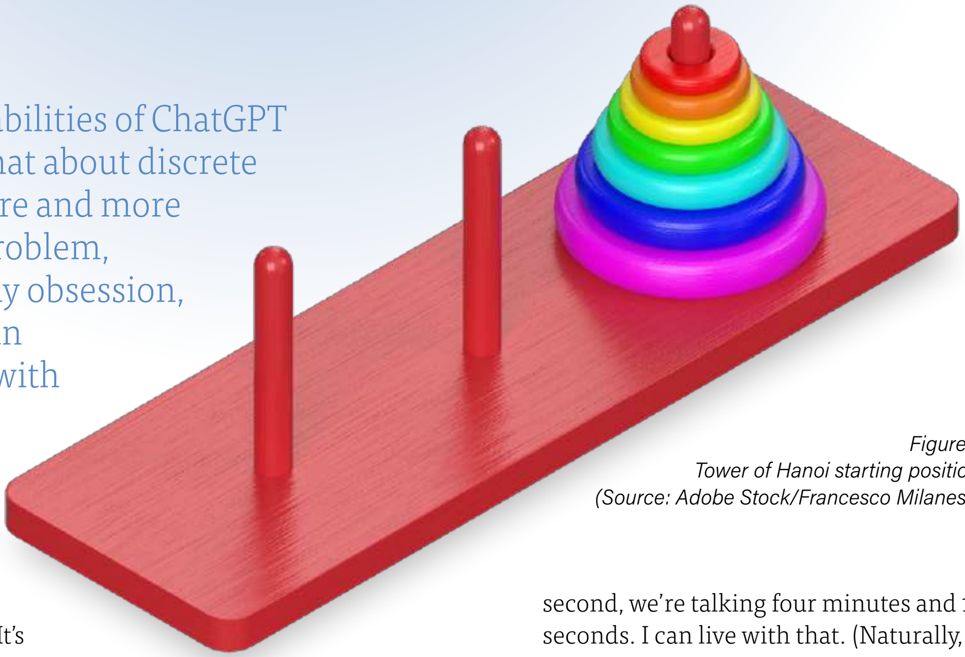


Figure 1:
Tower of Hanoi starting position.
(Source: Adobe Stock/Francesco Milanese)

The Tower of Hanoi

Let's first recap the Tower of Hanoi rules: It's a classic mathematical puzzle that involves three pegs, or shafts, and a number of disks with holes in them that can be stacked on any of the pegs (**Figure 1**). The disks are in different sizes, and the game begins with all of the disks stacked on top of each other, largest at the bottom and smallest at the top, on a single peg. The objective is to move the entire stack to another shaft, but you can only move one disk at a time. You can make as many moves as you want, but you cannot put any disk on top of a smaller one. You can read more at the game's Wikipedia entry [1], or even try it out yourself online [2].

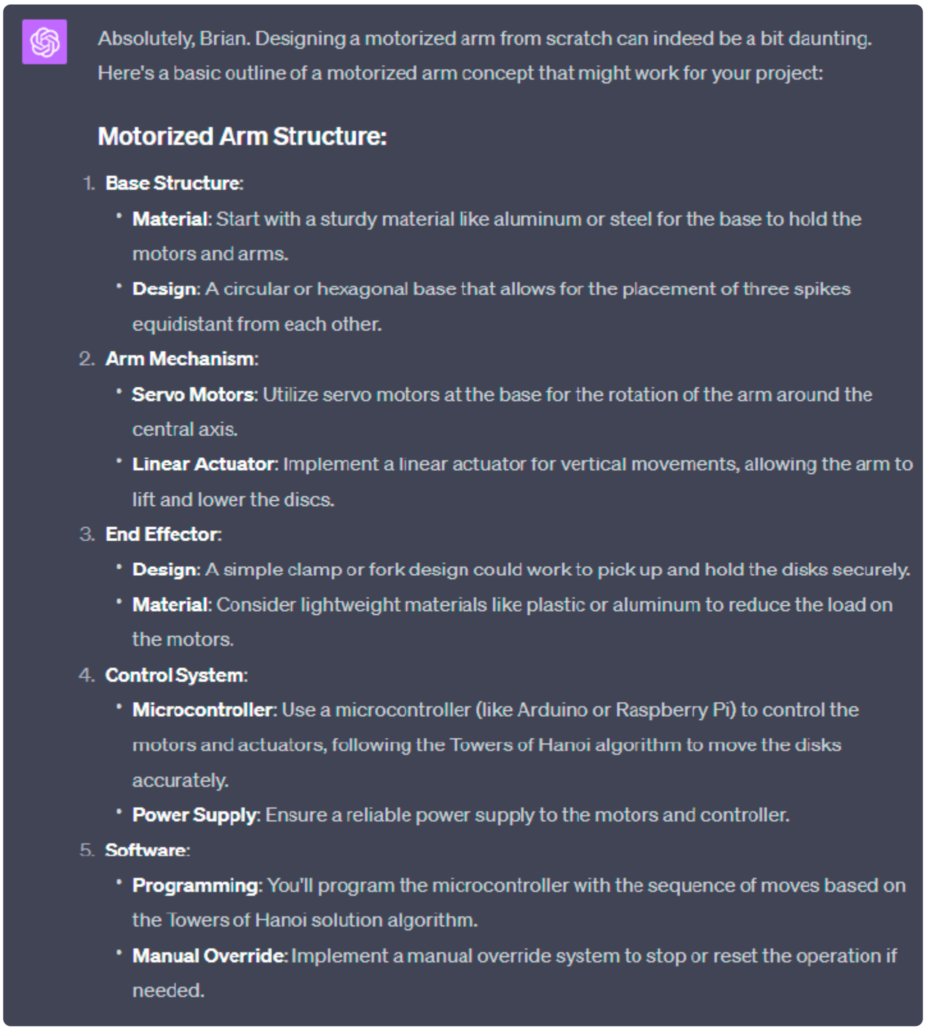
This problem is often used to teach recursive algorithms in computer science, because the minimum number of moves required to move a stack of *n* disks is always a power of two minus one. For example, in the most minimal example, let's say we have only one disk, and we have to move it to another shaft. How many moves would that take? Well, with $n = 1$, the minimum number of moves, $2^n - 1$, is 1. One move, obviously. Start with two disks, and you need $2^2 - 1$, or three moves. Three disks means at least seven moves, four disks 15 moves, and so on.

Let's Build It

With some nightly obsessive thinking, the project I decided on was a Tower of Hanoi egg timer that solves itself, one move per second — if I get my way. When the puzzle is solved, your eggs are ready. Of course, everyone likes their eggs different, but general recommendations are to boil them for between three and nine minutes, depending on your preference. For my needs, I reckon eight disks is ideal, which will take $2^8 - 1$, or 255 moves to solve it. At one move per second, we're talking four minutes and 15 seconds. I can live with that. (Naturally, it should be adjustable for others' preferences, but that's a detail for way later.)

Because programming a microcontroller in assembly, C, or Arduino, etc., is no problem for me, and I can wield a soldering iron without searing off any more of my fingerprints, the programming, circuit board design (if necessary) and construction poses little challenge. The reason that this is deemed an ambitious project for me is that I am useless at mechanical engineering, even though I've been fascinated by physical computing since I saw the first desktop robot arms of yore.

I need to build some sort of mechanism that will lift each disk up above its shaft, move it above another shaft, and drop it neatly into place. I realize that doing this one time per second may be a tall order, but that's the goal at this point.



Absolutely, Brian. Designing a motorized arm from scratch can indeed be a bit daunting. Here's a basic outline of a motorized arm concept that might work for your project:

Motorized Arm Structure:

1. **Base Structure:**
   - **Material:** Start with a sturdy material like aluminum or steel for the base to hold the motors and arms.
   - **Design:** A circular or hexagonal base that allows for the placement of three spikes equidistant from each other.
2. **Arm Mechanism:**
   - **Servo Motors:** Utilize servo motors at the base for the rotation of the arm around the central axis.
   - **Linear Actuator:** Implement a linear actuator for vertical movements, allowing the arm to lift and lower the discs.
3. **End Effector:**
   - **Design:** A simple clamp or fork design could work to pick up and hold the disks securely.
   - **Material:** Consider lightweight materials like plastic or aluminum to reduce the load on the motors.
4. **Control System:**
   - **Microcontroller:** Use a microcontroller (like Arduino or Raspberry Pi) to control the motors and actuators, following the Towers of Hanoi algorithm to move the disks accurately.
   - **Power Supply:** Ensure a reliable power supply to the motors and controller.
5. **Software:**
   - **Programming:** You'll program the microcontroller with the sequence of moves based on the Towers of Hanoi solution algorithm.
   - **Manual Override:** Implement a manual override system to stop or reset the operation if needed.

Figure 2: ChatGPT hit the ground running with some good ideas.

Ask ChatGPT

So, with my concept in place, I headed off down the Green Brick Road to consult the Wizard of Chat. We had a very lengthy discussion, which stopped when I'd used up my GPT-4 quota and was told not to look behind the curtain anymore, at least for a while. Probably a good thing, as this article can only be so long. Of course, I won't regurgitate the 9,784-word discussion here, but you'll find the original chat at [3].

I began by explaining that I am lacking in mechanical skills but that I can do all the software, electrical, and wiring stuff. ChatGPT was more than eager to assist. It came up with some good conceptual ideas, such as rotating the base to move the shafts themselves, rather than having to move an arm or the like to aim at the discs. How to lift them was another challenge to address: Do we use a robot gripper claw, an electromagnet, suction, or something else? How would the arm go up and down? A linear actuator seems like a good bet, but those are generally quite expensive, so it might be overkill. Of course, the bot pointed out that we'd need sensors to detect certain things, such as, say, when the disc is fully grasped, and perhaps when you've reached the top of your travel (if software alone cannot keep track of this), and so on. It wanted to help me detect the size of each disc, but I explained that this is unnecessary, as the game state will be in memory the entire time.

I asked it to describe the build, broadly. I was impressed at the things that it took into consideration (**Figure 2**). It tried to help with software, but I reiterated that I can take care of that; just help me build something physical that works and won't hurl disks through the microwave oven's door.

We've Got It Covered

As we explore the topic of artificial intelligence in helping us to realize our creative vision, be it designing an electronic product, writing code for our latest flight of fancy, or, indeed, helping to create actual art, it's worth noting that we've already had two striking magazine covers this year that drew on generative AI imagery. This somewhat refreshing throwback to when our covers were hand-illustrated works of art in themselves cannot be managed by the robot alone, so it was up to our very talented Art Director, Harmen Heida, to refine the prompts, and tweak and carefully finesse the final image using his traditional digital graphics tools.

I'm no artist, so I'm always envious of the creativity that goes into creating a striking cover design, not to mention the good-looking magazine you hold in your hands now. While Midjourney [4] can go some way toward helping us see what we could only imagine before, there are many other more technical visions we can realize using AI.

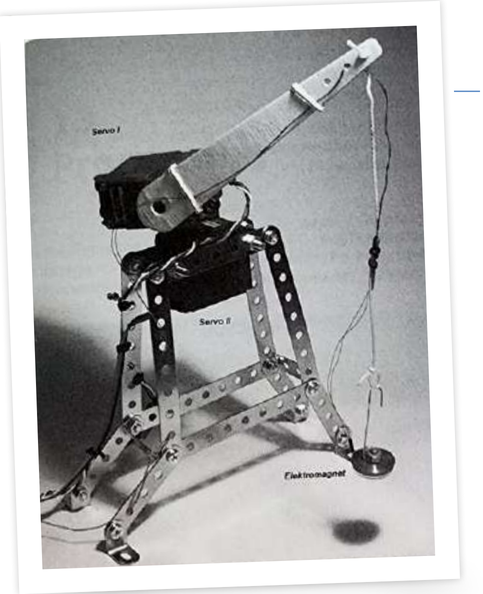


Figure 3: How Dr. Spanner prototypes things. (Source: Arduino: Circuits and Projects Guide, (Elektor, 2013))

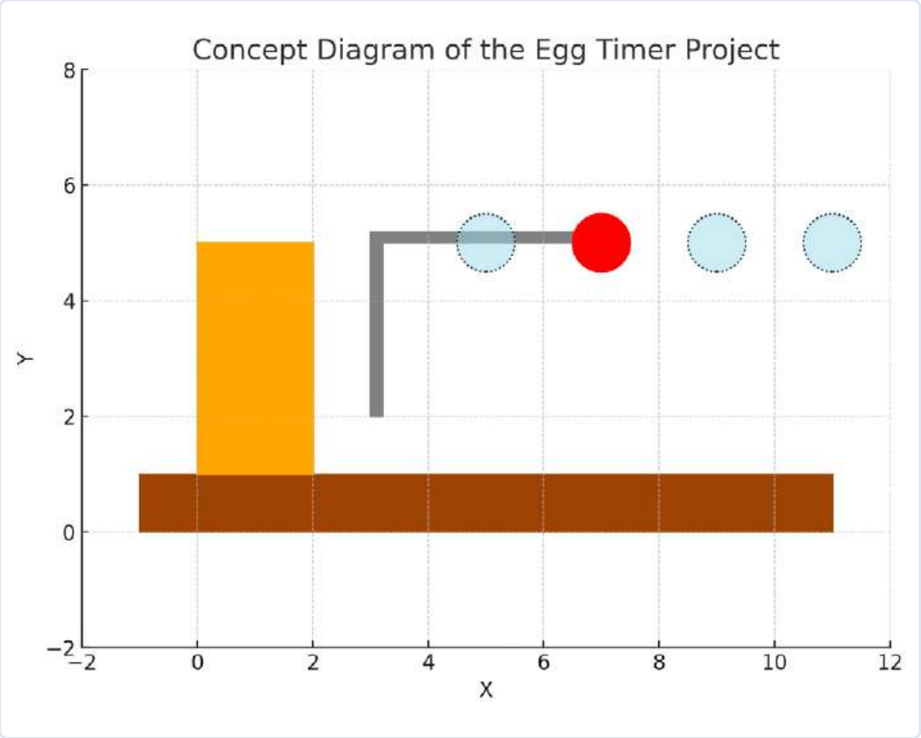


Figure 4: Not an encouraging Concept Diagram.

Choosing the Nuts and Bolts

When it came to rotating the mechanism, as is necessary to move between pegs, the option between a servo motor (my initial thought) and a stepper motor was presented. When I asked which to choose, it gave me a thorough rundown of the pros and cons of each. Based on the response (again, you can follow along at [3]), I'm still leaning toward a servo for rotation around the Y-axis.

Now, how to control the movement up and down the Y-axis? A linear actuator, or perhaps a cheaper micro metal gear motor with a shaft? I'm sure there are many better ways, and the solution is probably obvious to many readers out there — I could even have consulted one of my brilliant Elektor colleagues, but the challenge here was to do it myself with the help of ChatGPT (and a lot of googling for appropriate parts).

After the discussion on vertical movement, I ran into the snag that you will have to start the timer only when the water has reached boiling point. That alone takes several minutes. What am I supposed to do, as we all know that a watched pot never boils? Unless we could find some way to get the timer to watch the pot. Temperature sensor? Listen for boiling? Fixed timer? OK, a problem for later; let's just get the Tower working.

I opted to prototype on something quick and simple, such as the Meccano set I bought when I was working with Dr. Günter Spanner's book, *Arduino: Circuits and Projects Guide* (Elektor, 2013). The book even gave me a taste of putting together some simple mechanical structures (**Figure 3**), so I'm happy to *Meccanotype* my model. Later, we can talk about laser cutting some custom parts.

Visualizing It All

The ability to present visualizations is what differentiates GPT-4 from earlier versions, so the next step was to ask it to show me what it had in mind. Choosing between a 2D Sketch, a Flow Diagram, or a Concept Diagram, I opted for the last. The output led me to wonder about throwing the word "intelligence" out there so easily (**Figure 4**).

Well, that clearly was not going to do, so I hoped its 2D Sketch would result in something actually resembling the ideas we talked about. It was hardly any better (**Figure 5**) — the stack of disks (all the same size, apparently) is not even on any of the shafts! With a bit of correction, we reached

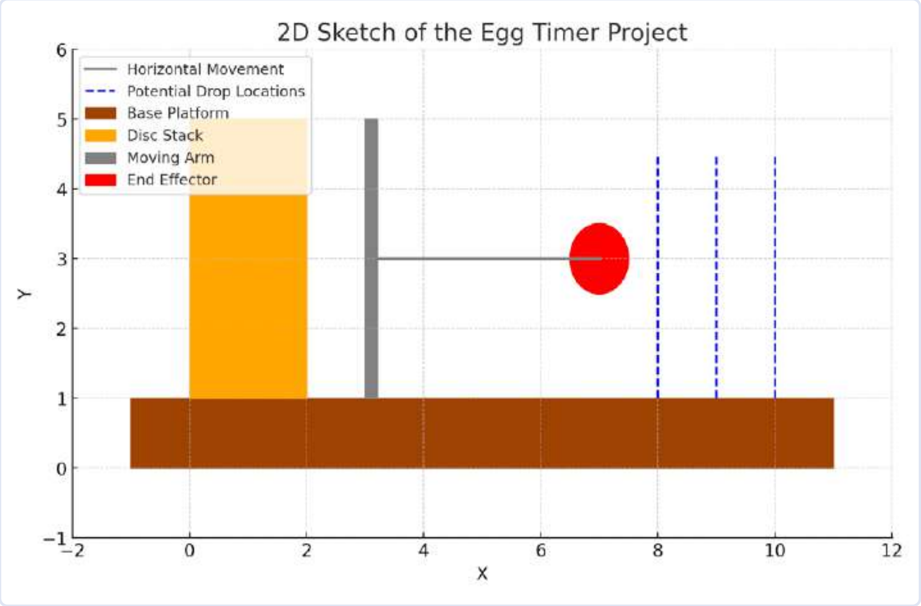


Figure 5: The 2D Sketch is not much better.

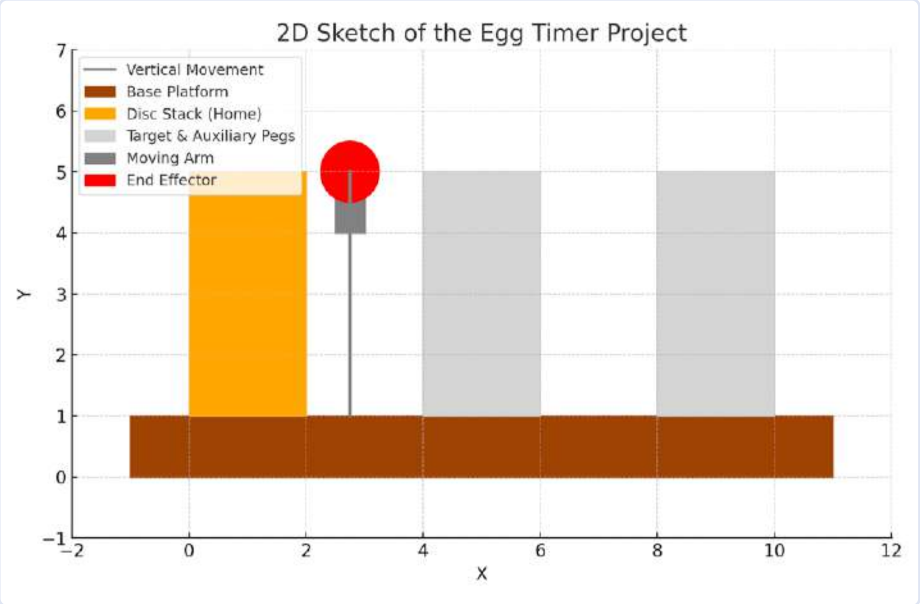


Figure 6: The improved 2D Sketch had me dropping this visualization idea.

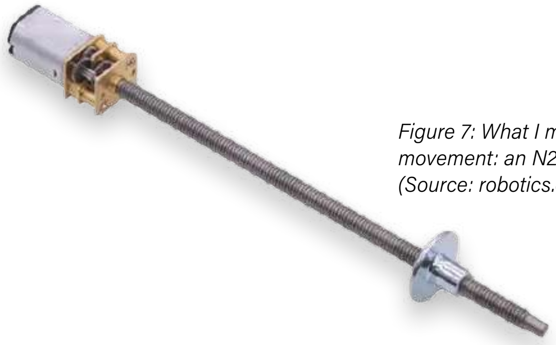


Figure 7: What I might use for the vertical movement: an N20 micro metal motor shaft, 100 mm. (Source: robotics.org.za/N20-500-L100-6V)

Figure 6, but, at that point, I decided that visualizations were not going to be the strong point of this endeavor, at least not with the prompts I was using.

I did digress when it offered me a "visualization" in the form of a file to download that had no extension. The mystery file was too intriguing to leave alone.

So, at this point, I decided on a servo for the base rotation and a robot gripper claw to grab the discs. What we concluded on, before my tokens ran out, was a discussion on what kind of linear mechanism to use, in particularly me asking about an N20 motor with a 100 mm M4 shaft, 500 RPM, 6 V (**Figure 7**).

To Be Continued...

At this point, not only was I out of compute time, but we're out of space for this installment. The challenge, however, will continue. I have set myself a project that is somewhat out of my electronics-only comfort zone, so wish me luck, and be sure to join me next time for a progress update! ◀

230181-C-01

About the Author

Brian Tristam Williams has been fascinated with computers and electronics since he got his first "microcomputer" at age 10. His journey with Elektor Magazine began when he bought his first issue at 16, and since then, he's been following the world of electronics and computers, constantly exploring and learning. He started working at Elektor in 2010, and nowadays, he's keen on keeping up with the newest trends in tech, particularly focusing on artificial intelligence and single-board computers such as Raspberry Pi.

Questions or Comments?

Do you have technical questions or comments about this article? Email the author at brian.williams@elektor.com.

Related Products

- **Dr. Günter Spanner, *Machine Learning with Python for PC, Raspberry Pi, and Maixduino* (E-book), Elektor 2022**
www.elektor.com/20150

PDF
Machine Learning with Python
For PC, Raspberry Pi, and Maixduino
python
Günter Spanner
elektor

WEB LINKS

[1] Tower of Hanoi — Wikipedia entry: https://en.wikipedia.org/wiki/Tower_of_Hanoi
[2] Tower of Hanoi — Maths Is Fun: https://mathsisfun.com/games/towerofhanoi.html
[3] Consulting ChatGPT on the project: https://tinyurl.com/chatgpthanoi
[4] Midjourney - Generative AI image tool: https://midjourney.com

Brussels Is Innovating

Support for Deep Tech

Illustration: Midjourney

By Priscilla Haring-Kuipers (The Netherlands)

Europe is known to have excellent scientists, but when it comes to bringing ideas to market, other world regions are leading. To overcome this, the European Innovation Council (EIC) was founded in 2021. I travelled to Brussels to speak with Isabel Obieta (EIC Programme Manager for Responsible Electronics) and Francesco Matteucci (EIC Programme Manager for Advanced Materials for Energy & Environmental Sustainability) about how the EIC is supporting ethical innovation.

Priscilla: What is the European Innovation Council?
Francesco: The EIC was born in 2021 from the desire to overcome the European paradox. In Europe, we have the best scientists in the world, but not the best scientific entrepreneurs. We are a one-stop-shop agency, which means we fund the innovation journey from ideation to marketing of our deep-tech innovators via three main funding schemes: *Pathfinder*, *Transition*, and *Accelerator*, including the EIC Fund, which makes equity investments in beneficiary companies. There are two types of calls in the EIC: One is open for all kinds of deep-tech proposals and the other is based on the challenges identified and supported by the Programme Managers.

Priscilla: What do you do as a Programme Manager?
Isabel: We each have our own background and bring our own vision to the EIC. It is not we who decide what gets funded, but our challenge is to find companies around the world that we think are relevant and connect them. In Pathfinder, I try to follow twenty projects, in Transition, I follow five or six, and in Accelerator, around ten. I really want to get to the basics of electronics and change how devices can be produced in a more sustainable way, so I feel that I need to go deeper into the beginning of the story.

Francesco: We are part of the community of our topic, and we help the community grow and strongly innovate and bring out the socioeconomic impact. We stay for a limited amount of time (maximum four years) and we support setting challenges of around 50-100 million euros on specific topics. I read a lot of papers and I meet a lot of scientists, engineers, and financial experts in order to understand where the future of my field is. I help the projects in my portfolio on the scientific side and minimize the reinvention of the wheel. I meet with the project partners, helping them with a network of investors, corporate managers, and brilliant people. I feel it is my responsibility to stimulate young scientists and entrepreneurs into understanding that you need to have an impact with what you're doing.

Isabel: We are very open; we go to all sorts of networking events where people can contact us directly, and many people do. We aim to understand who's who in our field and help projects through this journey of innovation. Sometimes,this is very fundamental research, so we try to understand what's behind the science. I think this is important, because we're all experts in our

Figure 1: Isabel Obieta.

Figure 2: Francesco Matteucci.

fields, so we can speak about the content and really contribute.

Francesco: Our role depends not only on our experience and the topic, but also on the beneficiary's reaction. It strongly depends on the one-to-one relationship, and I think this is what makes our job amazing. We also organize events together with the European Research Council, where they bring the Nobel Prizes and we bring the investors, corporates, and entrepreneurs.

Priscilla: What are the ethical developments in your field?
Isabel: One of my challenges is radical innovation in chips using new materials. Already, some people are working with organic electronics that could provide a way forward. Replacing silicon is going to be impossible, but, maybe for some specific devices, we can use an additive process instead of a subtractive one, or use less material.

Francesco: We are speaking a lot about sustainability and, in particular, the socio-economic impact of electronics. I think we need to map the entire electronics value chain and create good KPIs for ethics.

Isabel: I see a big change this year. A big push is coming from policy but there is also a lot of awareness and change coming from semiconductor companies. I am optimistic.

Priscilla: What do you think is the most important ethical question?
Isabel: One example is the human-centric Industry 5.0, where we can help a worker do their job better but also feel part of their workplace. I think that the social part of electronics is important, so we don't forget that electronics are here to help us with our way of living.

Francesco: Did we consider the holistic approach from an environmental point of view? If yes, did we identify a KPI with which we can compare solution A to solution B? For example, to extract hydrogen, you need a huge amount of fresh water, so we need to discuss distribution problems: Do you want to give someone the water to survive or to make hydrogen to power an airplane?

Isabel: Metrics for everything. Now we want to monitor crops with all kind of sensors, but is it worth the money and the environmental impact of manufacturing and implementation? We are trying to build metrics that are simplified, but realistic. This is also something we can contribute to the community of our beneficiaries, helping them to really measure what they are doing and the change they are provoking.

230489-01

European Innovation Council — Total Budget, 10.1 Billion Euros

"Europe's flagship innovation programme to identify, develop, and scale up breakthrough technologies and game-changing innovations" with three funding schemes:

- Pathfinder: research and deep tech projects with a high degree of scientific ambition and risk — up to 4 million euros per project.
- Transition: maturation and validation of novel ideas from lab to business — up to 2.5 million euros per project.
- Accelerator: supporting SMEs to develop and scale up to new markets or disrupt existing ones — up to 2.5 million euros per project.

WEB LINKS

[1] Isabel Obieta, Programme Manager for Responsible Electronics: https://eic.ec.europa.eu/isabel-obieta_en

[2] Francesco Matteucci, Programme Manager for Advanced Materials for Energy & Environmental Sustainability: https://eic.ec.europa.eu/francesco-matteucci_en

[3] European Innovation Council: https://eic.ec.europa.eu/index_en

The Elektor Store

Never expensive, always surprising

The Elektor Store developed from the community store for Elektor's own products, such as books, magazines, kits and modules, into a mature web store that offers great value for surprising electronics. We offer the products that we ourselves are enthusiastic about or that we simply want to try out. If you have a nice suggestion, we are here: sale@elektor.com.

Raspberry Pi 5 (8 GB RAM)

The new Raspberry Pi 5 delivers more performance than ever before. Thanks to the faster CPU, GPU and RAM, Raspberry Pi 5 is up to 3× faster than its already fast predecessor. In addition to the speed boost, the Raspberry Pi 5 (which features the new Raspberry Pi RP1 silicon for advanced I/O capabilities) also offers the following features for the first time ever: RTC, an on/off button, and a PCIe interface.

Price: €94.95

www.elektor.com/20599

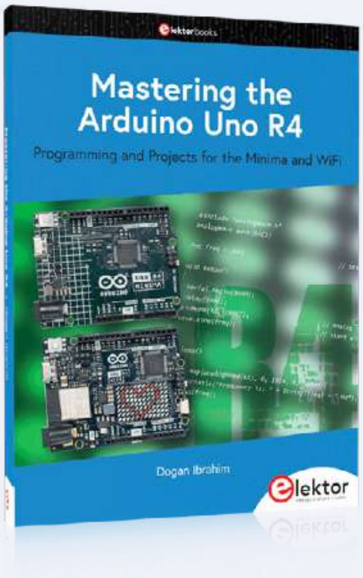
Mastering the Arduino Uno R4

Based on the low-cost 8-bit ATmega328P processor, the Arduino Uno R3 board is likely to score as the most popular Arduino family member so far, and this workhorse has been with us for many years. Recently, the new Arduino Uno R4 was released, based on a 48 MHz, 32-bit Arm Cortex-M4 processor with a huge amount of SRAM and flash memory. Additionally, a higher-precision ADC and a new DAC are added to the design. The new board also supports the CAN Bus with an interface.

Price: €34.95

Member Price: €31.46

www.elektor.com/20619


elektor books
Mastering the Arduino Uno R4
Programming and Projects for the Minima and WiFi
Dogan Ibrahim
elektor

Joy-Pi Advanced – Development Platform for Raspberry Pi, Arduino and ESP32

Price: €369.00

Member Price: €332.10

www.elektor.com/20589

PiKVM V4 Plus – KVM Over IP for Raspberry Pi CM4

Price: €449.00

Member Price: €404.10

www.elektor.com/20580

Raspberry Pi Pico Advanced Kit With 32 Modules and 32 Projects

Price: €59.95

Member Price: €53.96

www.elektor.com/20581

CrowVi 13.3" IPS HD Touch Display (1920×1080)

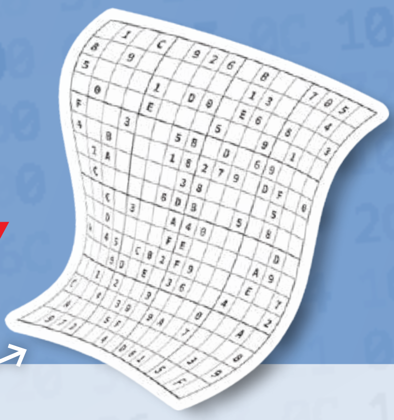
Price: €199.95

Member Price: €179.96

www.elektor.com/20583

Hexadoku

Puzzles with an Electronic Touch



Traditionally, the last page of *Elektor Magazine* is reserved for our puzzle with an electronics slant: Welcome to Hexadoku! Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor Store vouchers.

The Hexadoku puzzle employs digits in the hexadecimal range 0 through F. In the diagram composed of 16×16 boxes, enter digits such that **all** hexadecimal digits (that's 0–9 and A–F) occur once only in each row, once in each column, and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle, and these determine the starting situation.

Correct entries received enter a prize draw. All you need to do is send us **the digits in the gray boxes**.

SOLVE HEXADOKU AND WIN!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for five Elektor store vouchers worth **€50 each**, which should encourage all Elektor readers to participate.

PARTICIPATE!

By December 15th, 2023, supply your name, street address and the solution (the digits in the gray boxes) by email to:
hexadoku@elektor.com

PRIZE WINNERS

The solution to the Hexadoku in our September/October edition is: **013BE**.
Solutions submitted to us before October 15th were entered in a prize draw for 5 Elektor Store vouchers.
The winners are posted at elektormagazine.com/hexadoku.

Congratulations, everyone!

	4	2		D	9					6	0		B	E	
				1	4	6			A	7	E				
	7	E	1	0		2			C		F	9	3	D	
B					A	7	F	9	D	1					6
2	9	4	5	8							C	D	E	7	B
7		A	B	4							6	8	F		C
D			8	7	3	9			B	5	A	1			0
	0			B	C					E	4			9	
	A			F	7					B	1			C	
9			4	5	1	8			3	F	7	B			E
3		6	7	9	▨	▨	▨	▨	▨		5	F	1		4
C	F	1	2	3							9	5	8	0	7
E					B	3	8	1	4	A					9
	1	C	3	A		5			9		8	6	7	B	
				6	F	E			7	C	3				
	8	9		C	0					2	B		A	4	

D	A	E	3	F	B	1	5	8	C	2	6	4	0	9	7
B	F	6	0	C	E	A	7	D	5	9	4	8	1	2	3
4	1	5	8	3	2	0	9	E	7	B	F	6	A	C	D
2	7	9	C	6	D	8	4	A	0	1	3	B	E	5	F
9	0	B	E	7	4	5	6	F	D	8	1	3	2	A	C
3	2	4	6	9	1	C	A	0	B	5	7	D	F	E	8
7	C	A	D	8	3	B	F	9	4	E	2	5	6	0	1
5	8	1	F	D	0	2	E	C	3	6	A	7	9	B	4
6	3	F	9	B	5	7	D	1	E	A	C	2	4	8	0
8	4	0	1	E	F	3	C	2	6	7	5	9	B	D	A
A	5	D	B	4	8	6	2	3	F	0	9	C	7	1	E
C	E	2	7	0	A	9	1	B	8	4	D	F	3	6	5
F	9	C	2	A	6	D	0	4	1	3	8	E	5	7	B
E	D	7	A	1	9	4	3	5	2	C	B	0	8	F	6
0	B	3	5	2	7	F	8	6	A	D	E	1	C	4	9
1	6	8	4	5	C	E	B	7	9	F	0	A	D	3	2

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.

PROTEUS DESIGN SUITE

Driving forward with Manual Routing

Push and Shove Routing for dense layouts

Dedicated Differential Pairs Routing mode

Length Matching and Net Tuning Support

Visual DRC shows legal paths for route placement

HEAVY TRAFFIC

FASTER ROUTING AVAILABLE

labcenter
Electronics

www.labcenter.com

info@labcenter.com

Development tools in one location

Thousands of tools from hundreds of trusted manufacturers

Choose from our extensive selection at **mouser.com/dev-tools**

MOUSER ELECTRONICS