

06

771757 087002

Join this great initiative from
e-lektor & eeNews
in collaboration with
Studium Travel!

**Studium
Travel**

Unique Study Tours

eeNews
EUROPE

e-lektor
MEDIA & LEARNING

FOR READERS AND RELATIONS OF ELEKTOR EXCLUSIVE STUDY TOUR TO TAIWAN/TAIPEI

- **DESTINATION:** TAIPEI, TAIWAN
- **DATE:** FULL PROGRAM FROM 7 TO 13 SEPTEMBER 2025
- **FLIGHT:** FLEXIBLE BOOKING TO SUIT YOUR NEEDS
- **ACCOMMODATION:** EXCELLENT 4-STAR HOTEL

Join us on a journey to the cradle of electronics. Since the 1980s Taiwan has been a leader in our sector.

Many world-important innovations come from Taiwan. And even now, the country is still a crucial factor, partly due to top companies such as ASML. Developments in Embedded Systems, Embedded Systems Applications, Components, Production Technologies and innovations... you will see, experience and learn from that and more. A complete innovation tour full of knowledge and inspiration. Gain valuable insights. Masterclasses will bring you up to date on 'the next step is now'. This study trip is, in one word, fantastic. Are you joining in to experience this? Will you be there?

Register now! For more information and registration, please contact **info@studiumtravel.nl** or call Ronald Jans at **+31 651 158107**.

DISCOVER THE FUTURE OF ELECTRONICS AND EMBEDDED SOFTWARE, IN ASIA!

During this inspiring study tour we will take you to experts, you will visit leading companies and you will immerse yourself in the latest innovations.

In the program:

- Masterclasses at two renowned universities
- 8 company visits to local and (inter)national top companies in electronics, embedded software and digital programs
- Exclusive networking event at the Taipei Chamber of Commerce
- Time for relaxation, culture and leisure

A unique opportunity to expand your knowledge, network and gain inspiration!

Volume 51, No. 537
May & June 2025
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
www.elektor.com | www.elektormagazine.com

Content Director: C. J. Abate
Editor-in-Chief: Jens Nickel

For all your questions
service@elektor.com

Become a Member
www.elektormagazine.com/membership

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

Copyright Notice
© Elektor International Media b.v. 2025

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



Jens Nickel

International Editor-in-Chief, Elektor Magazine



Just Ask Elektor GPT

Projects that focus on measurement and testing are particularly popular with creative electronics engineers. Why? Because what you develop, “hack,” or simply rebuild can be used profitably in your own laboratory. I once asked our artificial intelligence program how many transistor testers we have published over the years. Elektor GPT came up with five designs published between 1988 and 2015. But, if you include multifunction testers and characteristic curve recorders, there were many more articles on the subject. As I’m just celebrating my 20th anniversary at Elektor and naturally can’t keep everything in my head, I’ve created a shortcut to www.elektormagazine.com/gpt. I’m using the AI tool myself more and more often as an intelligent article search solution.

You won’t find a transistor tester or characteristic curve recorder in this issue, but many other useful things are measured, tested, and optimized. Our author Roland Stiglmayr has shouldered the ambitious task of creating a kind of Elektor reference design for the control of a solar off-grid system. As you can imagine, this is a mammoth project; in this issue, you will find the first installment of a short series describing the MPPT tracker and charge controller (page 12). I also find Johannes Sturz’s PID controller tool particularly interesting. With it, you can optimize the PID parameters of a controller in Arduino code (page 40). Well done and quite useful!

Of course, we offer plenty of classic measuring solutions — for example, with the battery monitor and the milliohm meter adapter from my Elektor colleagues Saad Imtiaz and Clemens Valens (pages 6 and 54, respectively). And anyone who has always wanted to measure magnetic fields should take a look from page 18 onwards. By the way, there have already been other projects of that kind in Elektor. Try asking Elektor GPT yourself!



Submit to Elektor!

Your electronics expertise is welcome! Want to submit an article proposal, an electronics tutorial on video, or an idea for a book? Check out Elektor’s Author’s Guide and Submissions page:

www.elektormagazine.com/submissions



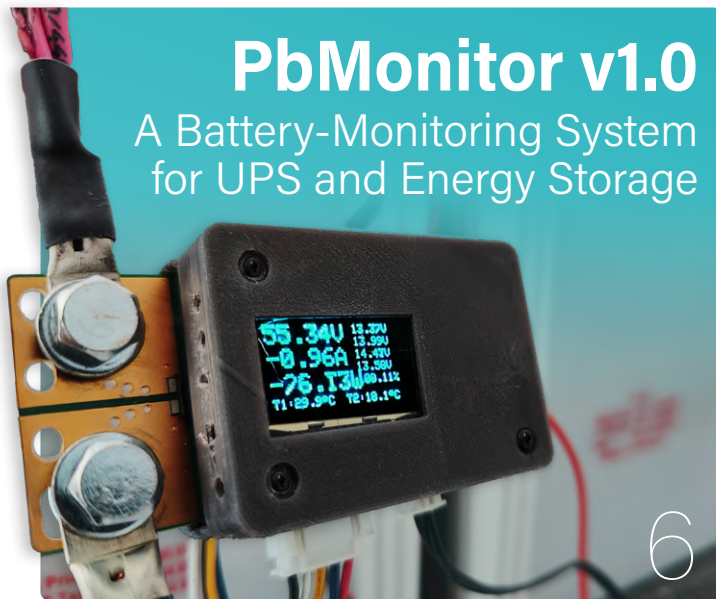
Elektor Labs Ideas & Projects

The Elektor Labs platform is open to everyone. Post electronics ideas and projects, discuss technical challenges and collaborate with others.

www.elektormagazine.com/labs

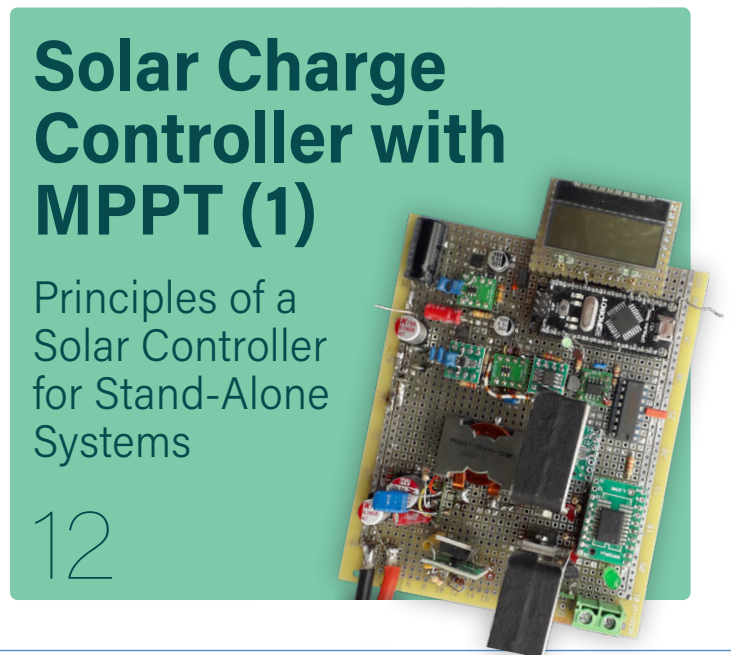
The Team

International Editor-in-Chief: Jens Nickel | **Content Director:** C. J. Abate | **International Editorial Staff:** Hans Adams, Asma Adhimi, Roberto Armani, Jan Buiting, Rolf Gerstendorf (RG), Ton Giesberts, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-Francois Simon, Clemens Valens, Brian Tristram Williams | **Regular Contributors:** David Ashton, Stuart Cording, 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



PbMonitor v1.0

A Battery-Monitoring System for UPS and Energy Storage



Solar Charge Controller with MPPT (1)

Principles of a Solar Controller for Stand-Alone Systems

Regulars

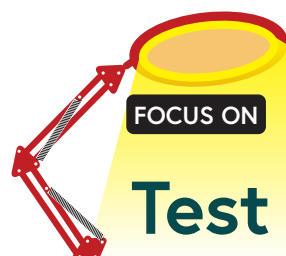
- 3 **Colophon**
- 50 **Starting Out in Electronics...**
...Continues with Tone Control
- 70 **Peculiar Parts**
Analog Meters
- 84 **From Life's Experience**
Who Doesn't Honor the Small Things?
- 86 **2025: An AI Odyssey**
The Transformative Impact on Software Development
- 89 **Err-lectronics**
Corrections, Updates, and Readers' Letters

Features

- 28 **Precise or Accurate?**
Your Instruments Need to Be Both!
- 34 **AD7124: A Precision ADC in Practice**
Features for Sensor Signal Conditioning
- 40 **PID Control Tool**
Optimize Your Parameters Easily
- 98 **Nortonized Wien Bridge Oscillator**
Small Changes Yield Significant Improvements
- 102 **Putting a \$0.10 Controller to the Test**
The CH32V003 RISC-V Microcontroller and MounRiver Studio in Practice

Projects

- 6 **PbMonitor v1.0**
A Battery-Monitoring System for UPS and Energy Storage Applications
- 12 **Solar Charge Controller with MPPT (1)**
Basic Principles of a Solar Controller for Stand-Alone Systems
- 18 **B-Field Integration Magnetometer**
With Home-Made Sensors
- 54 **Milliohmmeter Adapter**
Uses the Precision of Your Multimeter
- 64 **Frequency Counter**
Portable and Auto-Calibrating Via GPS
- 72 **Stand-Alone Crystal Tester**
How Accurate Is Your Clock Source?
- 78 **Low-Cost I²C Tester**
Connect I²C Devices Directly to Your PC
- 94 **Raspberry Pi Standalone MIDI Synthesizer (2)**
Enhancing Our Setup with Intelligence
- 108 **An FPGA-Based Audio Player with Equalizer (2)**
Adding Volume Control, Advanced Mixing, and a Web Interface



Test & Measurement



Precise or Accurate?

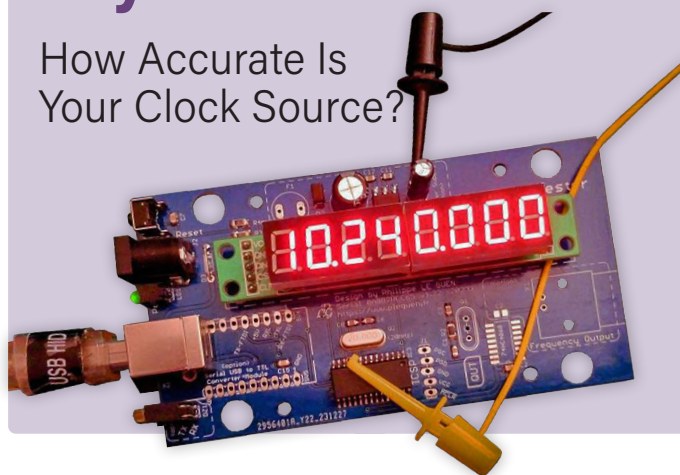
Your Instruments
Need to Be Both!

28

Stand-Alone Crystal Tester

72

How Accurate Is
Your Clock Source?



Industry

- 46 **embedded world 2025**
- 53 **Elektor Academy Pro Box**
Book + Online Course + Hardware
- 58 **The Next Leap in Semiconductors**
Onward Toward 1.4 nm
- 61 **Through-Hole Technology Connectors**
The Best of Two Worlds

Next Editions

Elektor Magazine July & August 2025

As usual, we'll have an exciting mix of projects, circuits, fundamentals, and tips and tricks for electronics engineers and makers. Our focus will be on IoT & Sensors.

- > OBD2 Sensor Dashboard
- > OBD2: Retrofitting Rev Counter and Shift Light
- > Meshtastic: A Mesh of LoRa Radios
- > Elektor Mini-Wheelie: A Self-Balancing Robot
- > Contact-Free E-Field Measurements
- > Ultrasonic Obstacle Detector
- > Analog Audio Frequency Generator
- > Radar Sensors
- > The New I3C Standard in Practice

Elektor Magazine's July & August 2025 edition will be published around **July 9, 2025**.

Arrival of printed copies for Elektor Gold members is subject to transport.

BONUS CONTENT

Check out the free Test & Measurement
bonus edition of Elektor Mag!

- > PWM Measurement with a PIC
- > Review: Fnrisc 2C53T with Two Channels and 50-MHz Bandwidth
- > FPGA-Based Voltmeter: MAX1000 and VHDPlus Make It Simple
- > Infographics: Test and Measurement

[www.elektormagazine.com/
test-measurement](http://www.elektormagazine.com/test-measurement)



**B-Field Integration
Magnetometer**
With Home-Made
Sensors

18

PbMonitor v1.0

A Battery-Monitoring System for UPS and Energy Storage Applications

By Saad Intiaz (Elektor)

Monitor your battery bank's health with PbMonitor. This simple, low-cost solution tracks voltage, current, and temperature in real time, sending data via MQTT to Home Assistant for smarter battery monitoring in UPS, renewable energy, and off-grid applications.

Battery banks are widely used in UPS systems, renewable energy storage, and off-grid applications. However, monitoring their health and performance remains a challenge, as traditional UPS systems treat multiple batteries as a single unit, leading to charge imbalance and inefficiencies. PbMonitor is a simple yet effective monitoring solution that can track voltage, current, and temperature of batteries (Lead-Acid and Li-Ion) in real-time. The collected data is transmitted via MQTT to Home Assistant, allowing for real-time monitoring and automation. This article describes a simple and inexpensive battery monitoring system that helps you keep track of your batteries, providing a cost-effective and scalable solution for battery management in various applications.

A Practical Solution

Energy storage systems, particularly those utilized in UPS configurations, frequently employ multiple batteries in series to meet voltage requirements. While this topology simplifies power delivery, it introduces significant challenges related to charge balancing, individual cell health monitoring, and state-of-charge (SoC) disparities. Conventional UPS systems treat the battery bank as a singular

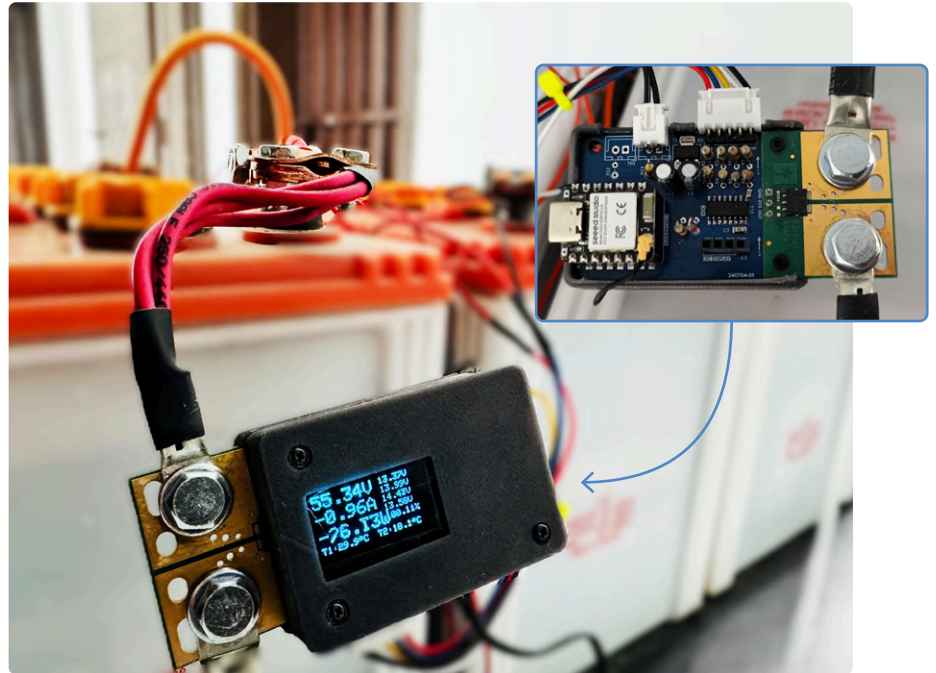


Figure 1: PbMonitor system connected to the 48-V battery bank for real-time monitoring.

entity, lacking the granularity required to detect cell-to-cell imbalances, which can precipitate premature failure and reduced operational efficiency.

The impetus for developing PbMonitor (Figure 1) grew from a practical necessity: my UPS system comprises four 12-V lead-acid batteries connected in series to yield a nominal 48-V supply (Figure 2). However, without a dedicated monitoring system, individual battery voltages remained unknown, introducing risks of overcharging or undercharging specific batteries. With nearly one and a half years of operation, it became imperative to implement a system capable of diagnosing battery health, identifying performance deviations, and facilitating proactive maintenance.

The motivation for this project stemmed from my own use of lead-acid batteries, which highlighted the need for a dedicated

monitoring solution. This inspired the name *PbMonitor*, with "Pb" representing lead, aligning with the core focus of tracking and optimizing lead-acid battery performance.

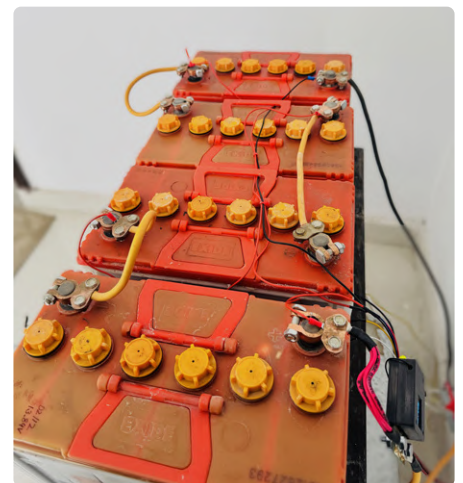


Figure 2: The 48-V battery pack configuration comprising four 12-V lead-acid batteries in series.

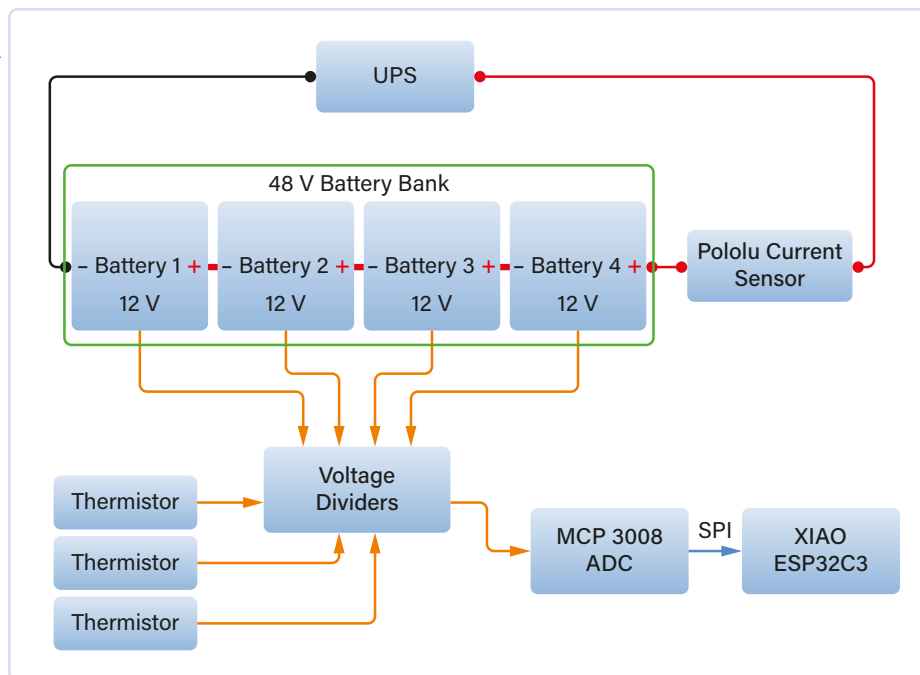


Figure 3: Block diagram of the project.

Beyond UPS environments, PbMonitor's application extends to renewable energy systems. By implementing a monitoring framework, battery lifespan can be extended, predictive maintenance strategies can be optimized, and the overall reliability of energy storage solutions can be improved.

System Architecture

The PbMonitor system is built around the ESP32-C3 microcontroller, which provides reliable Wi-Fi and BLE communication capabilities for real-time monitoring. To ensure high-resolution voltage and current measurements, the MCP3008 external ADC by Microchip [1] is integrated, offering 10-bit precision across eight channels and interfacing with the ESP32-C3 via SPI, as shown in the block diagram (Figure 3).

Voltage readings are obtained using a network of precision resistors forming a voltage divider, allowing compatibility with the ADC's input range. For current measurement, the system employs the Pololu ACS72981 module (chip by Allegro MicroSystems) [2], a high-precision Hall-effect current sensor module that enables bidirectional current tracking, critical for monitoring charge and discharge cycles. This current sensor module was also used in the AmpVolt project series and proven to be a reliable sensor for current measurements [3].

Additionally, thermistors are integrated to continuously monitor both environmental and battery temperature, providing crucial thermal

data for battery longevity and operational safety. The monitored battery bank consists of four 12 V lead-acid batteries arranged in series to create a 48-V system, requiring discrete voltage tracking to prevent imbalance.

The Schematic

The schematic (Figure 4) follows a structured approach to battery parameter acquisition. Each battery node is connected to a voltage divider circuit (R1 to R8) to scale the voltage down to a range suitable for the MCP3008 ADC (IC1), which operates at a 10-bit resolution with an input voltage limit of 0 to 3.3 V.

In a series-connected battery system like this, measuring individual battery voltages can be challenging due to the shared connection points. To simplify the measurement process and reduce hardware complexity, voltage is measured at the positive terminal of each battery relative to the system ground. This method provides cumulative voltages (e.g., 12 V, 24 V, 36 V, and 48 V). The MCP3008 ADC (IC1) sequentially samples these voltages, which are then processed in the firmware by subtracting each battery's voltage from the one before it.

This approach eliminates the need for complex differential voltage sensing circuits, reduces the number of ADC channels required, and avoids floating ground issues that can arise when measuring individual cell voltages directly. Additionally, it ensures accurate readings without introducing isolation challenges that come with direct cell measurement, making it a practical and reliable solution for multi-battery monitoring systems.

The Pololu ACS72981 ± 50 -A current sensor at K2 is used to measure bidirectional current flow, providing real-time data for both charging and discharging cycles. This Hall-effect sensor outputs an analog voltage that varies with current. At 0 V, the output voltage is 1.67 V, serving as the reference point. When current

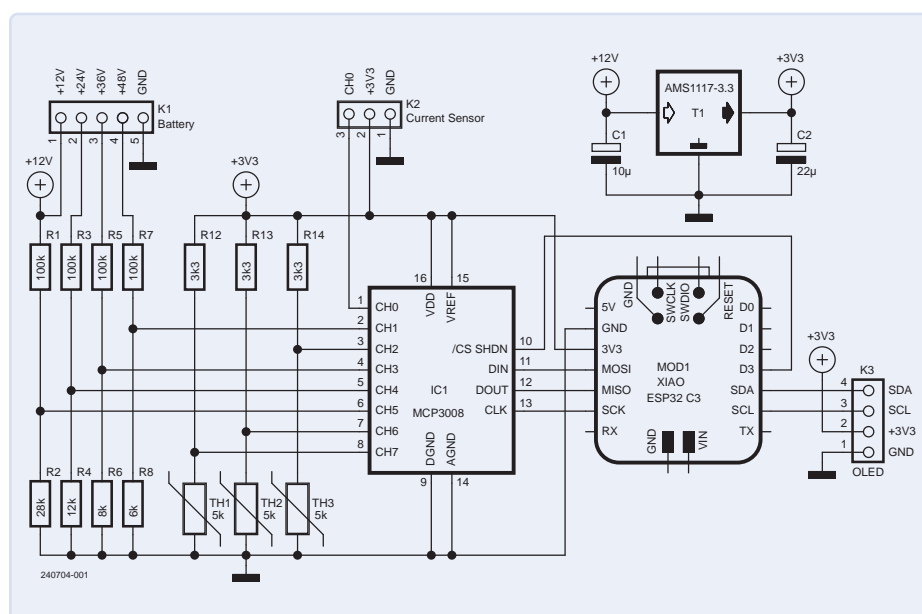


Figure 4: The schematic diagram.

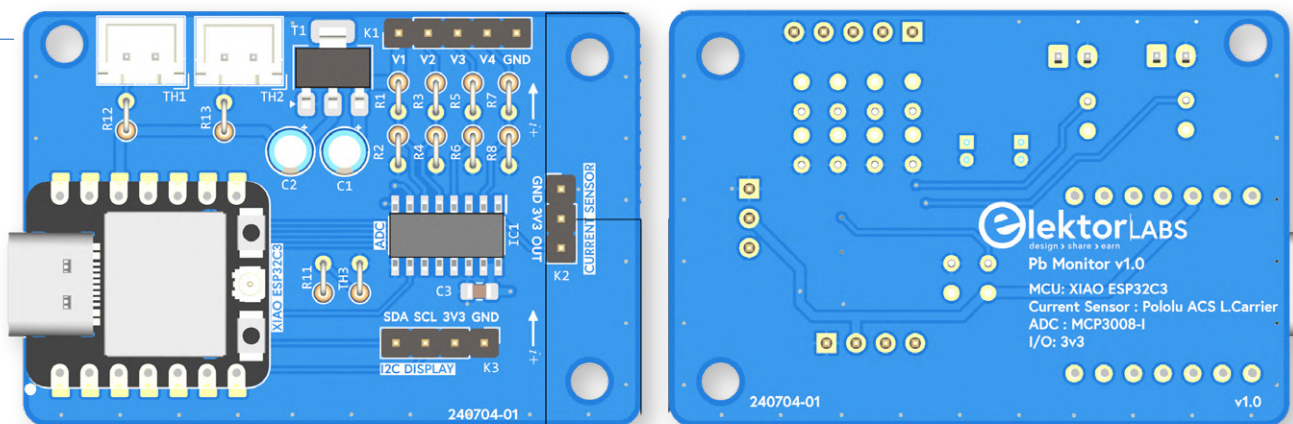


Figure 5: Front and back rendering of the PCB.

flows clockwise (discharging), the voltage increases at a rate of 0.0264 V per amp, while a counterclockwise (charging) current results in a proportional decrease in voltage.

The sensor's output is read by the MCP3008 ADC, which digitizes the voltage for processing. The MCP3008 ADC, operating at 3.3 V, is configured for single-ended mode, using all eight channels for battery voltage, current, and temperature measurements. The ADC has a 10-bit resolution and a maximum sampling rate of 75 kSamples/s at 3.3 V; but, when all eight channels are used, the effective per-channel sample rate is reduced due to sequential sampling. With an SPI clock of 1 MHz, the ADC can achieve a practical sample rate of approximately 9 to 10 kSamples/s per channel, ensuring reliable real-time monitoring.

Temperature monitoring is implemented through thermistors (T1...T3) also connected to the MCP3008, ensuring continuous tracking of both environmental and battery conditions. The thermistor operates as part of a voltage divider, where its resistance changes with temperature, altering the measured voltage. To obtain the thermistor resistance, the voltage across it is measured using a voltage divider circuit. The resistance is calculated using the formula:

$$R_{\text{thermistor}} = R_{\text{fixed}} \times \left(\frac{V_{\text{measured}}}{V_{\text{ref}} - V_{\text{measured}}} \right)$$

where R_{fixed} is the known series resistor value, V_{measured} is the voltage drop across the thermistor, and V_{ref} is the reference voltage. This voltage is used to calculate resistance, which is then converted to temperature using the Steinhart-Hart equation:

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3$$

where T is temperature in Kelvin, R is thermistor resistance, and A , B , C (provided in the thermistor's datasheet) are thermistor-specific coefficients. The temperature is finally converted to Celsius. Using thermistors is more effective than using 1-wire-sensors, as they are fast and accurate.

Moving to the MOD1, the XIAO ESP32-C3 by Seeed Studio [4] was an obvious choice for this project due to its compact size, powerful ESP32-C3 architecture, and well-designed form factor. Seeed Studio has done a quite good job in creating these XIAO series boards that are not only small but also highly versatile, with a good balance of IO options for integration into various applications. Its ability to be directly soldered onto a PCB or plugged in using 2.54 mm headers makes it extremely flexible for both prototyping and final implementations.

The XIAO ESP32-C3 (MOD1) is connected to the ADC via SPI, applies calibration algorithms, and transmits data via MQTT to a Home Assistant server. An OLED screen connection K3 (2.54-mm pitch) is also provided for showing the real-time data on site. Finally, an AMS1117 (T1) LDO by UMW is used to power the entire system.

PCB Layout

The PCB was designed to be compact, measuring 53.5 × 36 mm. The goal was to keep the board as small as possible so that it does not appear overwhelming when connected to the system. As seen in **Figure 5**, the PCB layout strategically positions the Pololu current sensor module on the right side. To facilitate voltage sampling, 2.54-mm pitch horizontal JST connectors were used, ensuring secure and reliable connections. The same connector type was used for the thermistors.

The OLED display connector was placed in the middle of the board for optimal visibility, while the XIAO ESP32-C3 module was positioned on the left side to ensure efficient space utilization and signal integrity. The layout was optimized to minimize noise, ensuring stable ADC readings and reliable operation in demanding environments.



Component List

Resistors

R1, R3, R4, R5, R7 = 100 kΩ
R2 = 28 kΩ
R4 = 12 kΩ
R6 = 8 kΩ
R8 = 6 kΩ
R11...R13 = 3.3 kΩ
TH1...TH3 = 5 kΩ

Capacitors

C1 = 10 μF
C2 = 22 μF
C3 = 0.1 μF

Semiconductors

IC1 = MCP3008-I
T1 = AMS1117-3.3V

Modules and Connectors

K1 = Pin Header 1×5, 2.54 mm Horizontal
K2 = Pololu Current Sensor
K3 = I2C OLED Screen

Software Integration and Home Assistant

The firmware for PbMonitor is developed using the Arduino framework and is compatible with the Arduino IDE and PlatformIO, ensuring ease of development and deployment. The sketch, along with the complete hardware files, is available in the project's GitHub repository [5]. The firmware relies on several essential libraries, including *WiFi.h* for network communication, *SPI.h* for interfacing with the MCP3008 ADC, *Adafruit_MCP3008.h* for ADC handling, *MQTTPubSubClient.h* for MQTT communication with Home Assistant, as well as *Adafruit_GFX.h* and *Adafruit_SSD1306.h* for real-time OLED display updates.

The firmware also calculates a State of Charge (SoC) using the battery voltage thresholds and is designed for future State of Health (SoH) estimation, which will involve tracking charge/discharge cycles and degradation over time. Additional planned features are discussed in the **Future Improvements** section later in this article.

Various parameters in the software are configurable, including ADC calibration factors,

MQTT update intervals, and threshold values for battery alerts. The firmware applies calibration algorithms to convert raw ADC values into real-world voltage and current measurements, using predefined scaling factors based on voltage dividers and sensor characteristics. The precision of voltage measurements is within ± 0.05 V, while current measurements, obtained from the ACS72981 sensor, have a resolution of 0.0264 V per A, allowing accurate detection of even small current variations.

The processed data are transmitted to Home Assistant [6] via MQTT, allowing seamless integration into a smart home environment.



*Beyond UPS environments,
PbMonitor's application
extends to renewable
energy systems.*

The MQTT topics are structured for easy integration, allowing real-time monitoring of battery voltages, current, and temperature.

Figure 6 shows how the data is displayed on the Home Assistant Dashboard, where users can track battery performance and receive alerts when critical thresholds are met, by creating automations in Home Assistant.

For users looking to set up MQTT integration with Home Assistant, it has been previously documented the step-by-step procedure in another article, where I implemented the same method for an ESP32-based Energy Meter [7]. That guide can be referenced for setting up MQTT configurations, authentication, and data visualization within Home Assistant.

Monitoring Batteries with PbMonitor

UPS systems charge lead-acid batteries using a multi-stage charging process to maintain battery longevity and efficiency. The charging process typically consists of three stages: bulk charge, absorption charge, and float charge. During the bulk charge phase, the UPS supplies a high current to rapidly charge the batteries until they reach approximately 80%

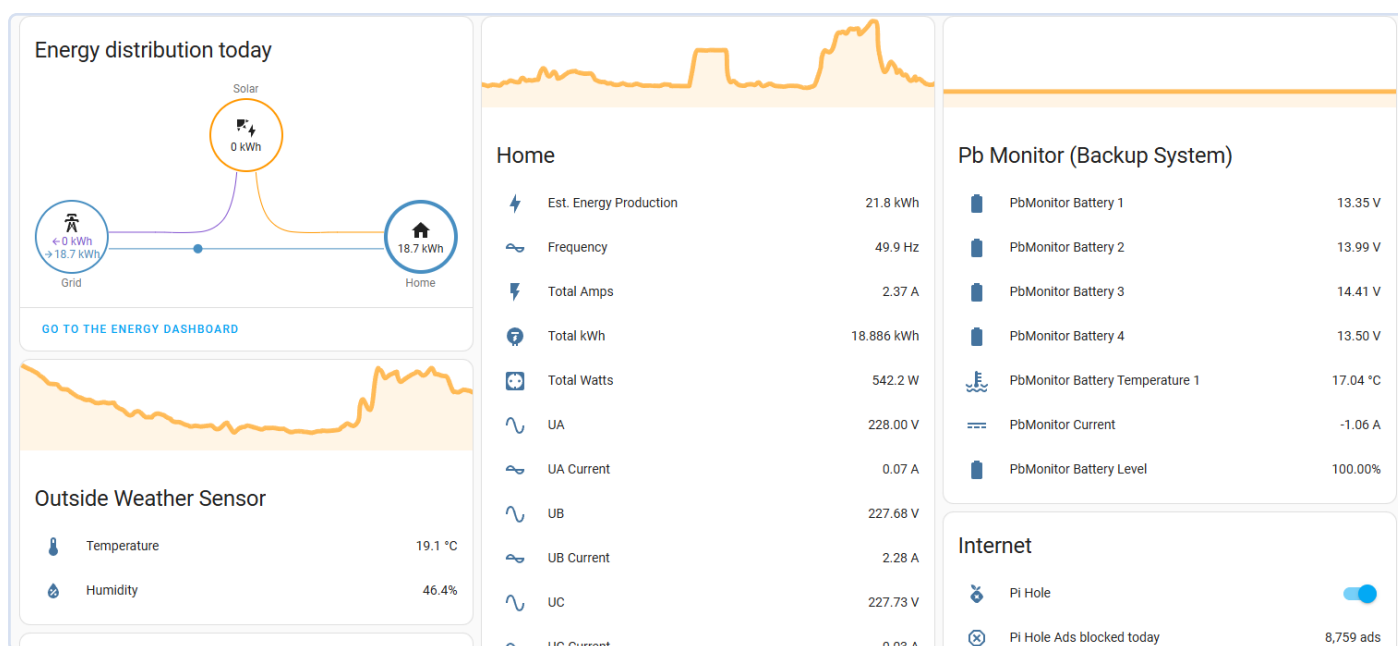


Figure 6: Real-time battery monitoring data displayed on the Home Assistant dashboard.



Figure 7: Charging and discharging trends of the battery bank, visualized using the Home Assistant History dashboard.

of their full capacity. In the absorption phase, the voltage is held at a constant level while the current gradually decreases, allowing the batteries to reach full charge without overheating. Finally, in the float charge stage, the UPS maintains a trickle charge to compensate for self-discharge, ensuring the batteries remain fully charged without overcharging.

Despite the batteries being fully charged, the UPS continuously supplies a 1 A trickle current. This small current prevents sulfation, a common issue in lead-acid batteries where lead sulfate crystals form on the battery plates, reducing capacity and efficiency. By maintaining this low-level charge, the UPS ensures that the batteries remain in optimal condition for standby use.

To test the charging and discharging cycle, I fully charged the battery setup and ran it on backup mode. The UPS provided a backup duration of 4 hours and 40 minutes before depletion. The collected data during this full charge cycle (**Figure 7**) showed that Battery 1 and Battery 4 reached a charge voltage of 13.5 V, while Battery 2 and Battery 3 charged up to 14.23 V and 13.9 V, respectively. This discrepancy indicates an imbalance in

charging, which could be due to variations in internal resistance or differences in aging between the batteries.

During discharge, notable voltage differences were observed. Battery 1 dropped to 8 V approximately 40 minutes before the end of the discharge cycle, while Battery 4 dropped to 8 V just 10 minutes before full discharge. In contrast, Battery 2 and Battery 3 maintained higher voltages, each stabilizing at 11.5 V by the end of the discharge cycle. These differences suggest that Battery 1 and Battery 4 may have reduced capacity or higher internal resistance, leading to earlier voltage drops compared to the other two batteries. This imbalance emphasizes the importance of real-time monitoring to detect weak batteries and take corrective action before the system performance is compromised.

Future Improvements

Several enhancements can be made to PbMonitor to improve accuracy, reliability, and usability. On the software side, as seen in Figure 7, the current readings fluctuate significantly, bouncing between zero and the actual value. Since clamp meter measurements remain stable, this issue needs

further investigation. Implementing filtering techniques such as moving averages or adjusting the ADC sampling rate could stabilize the readings. Additional software refinements include integrating a button to toggle between different display views, refining the SoC (State of Charge) and SoH (State of Health) algorithms for better accuracy, and implementing internal logging to track historical data. Adding predictive analytics to estimate battery backup time and performance degradation trends would also enhance the system.

For hardware, the current PCB layout is compact but has only three mounting holes, which may not provide sufficient structural stability. Increasing the board size slightly and incorporating four mounting holes would improve enclosure mounting. Additionally, adding two more thermistors would allow monitoring of all four batteries instead of just the middle two, providing a more complete thermal profile. Future iterations could also include a real-time clock (RTC) for precise logging of charge and discharge cycles, along with EEPROM or SD card storage for long-term data retention. Expanding connectivity options, such as Bluetooth or LoRaWAN,



could enable remote monitoring, while integrating a buzzer or LED indicators could provide real-time alerts for voltage anomalies. Furthermore, while keeping the Home Assistant MQTT feature, a web server feature should be added for locations where Wi-Fi or Home Assistant is inaccessible, providing a small dashboard on the ESP32 web server to keep track of system status. ◀

240704-01

Questions and Comments?

If you have questions about this article, feel free to email the author at saad.imtiaz@elektor.com or the Elektor editorial team at editor@elektor.com.

About the Author

Saad Imtiaz, Senior Engineer at Elektor, is a mechatronics engineer who has extensive experience in embedded systems and product development. His journey has seen him collaborate with a diverse array of companies, from innovative startups to established global enterprises, driving forward-thinking prototyping and development projects. With a rich background that includes a stint in the aviation industry and leadership of a technology startup, Saad brings a unique blend of technical expertise and entrepreneurial spirit to his role at Elektor. Here, he contributes to project development in both software and hardware.



Related Products

- > **Seeed Studio XIAO ESP32C3**
www.elektor.com/20265
- > **0.96" OLED Display (Blue, I²C, 4-Pin)**
www.elektor.com/18747
- > **Hioki DT4256 Multimeter** ◯
www.elektor.com/20208



WEB LINKS

- [1] MCP3008 2.7 V 10-Bit A/D Converters, Microchip: <https://www.microchip.com/en-us/product/mcp3008>
- [2] ACS72981 Current Sensor Large Carriers, Pololu: <https://www.pololu.com/category/317/acs72981-current-sensor-large-carriers>
- [3] Saad Imtiaz, "AmpVolt v2.0 Project Update," Elektor 1-2/2025: <https://elektormagazine.com/240672-01>
- [4] XIAO Board Series by Seeed Studio: <https://www.seeedstudio.com/xiao-series-page>
- [5] Pb Monitor GitHub Repository: <https://github.com/ElektorLabs/Pb-Monitor>
- [6] Home Assistant: <https://www.home-assistant.io/>
- [7] Saad Imtiaz, "Project Update #4: ESP32-Based Energy Meter," Elektor 11-12/2024: <https://elektormagazine.com/240349-01>



**WÜRTH
ELEKTRONIK**
MORE THAN
YOU EXPECT

WE are here for you!

Join our free webinars on:
www.we-online.com/webinars

Power Up with Würth Elektronik!

With our wide portfolio of high-quality components, a toolbox full of technical guidelines, our design platform REDEXPERT, books, and application-based content, we provide the right solution for every aspect of your design.

Our team of experts provides you with personal and digital support for all questions relating to power conversion. Power up your efficiency and design-in time with us.

www.we-online.com

Highlights

- Wide range of power inductors, input- and output capacitors, EMI solutions, thermal materials and electromechanical components
- Design platform REDEXPERT
- Application notes and reference designs
- Reference guide DC/DC converter handbook
- Ready-to-use simulation models

#PowerUp

Solar Charge Controller with MPPT (1)

Basic Principles of a Solar Controller for Stand-Alone Systems

By Roland Stiglmayr (Germany)

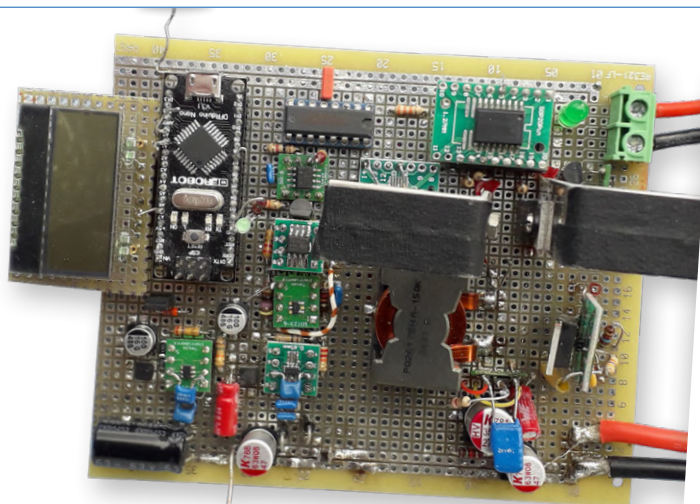
Off-grid solar power systems provide electricity in places without mains power, like clubhouses or garden sheds, using a battery for storage. A charge controller with maximum power point tracking (MPPT) optimizes charging, and this article explains its basics, benefits, and design.

Off-grid solar power systems are used in places where no mains power is available, for example at club houses or garden sheds. In contrast to balcony power plants, they always require an energy storage device in the form of a battery, otherwise there would be no electricity without the sun. Of course, it is desirable to charge the battery with the maximum power. This is done by a charge controller with maximum power point tracking (MPPT), the design of which is described here. This first part of the article deals with the basics, advantages, how it works, and the design, before you can start heating up the soldering iron in the second part.

MPPT Controllers: Maximizing Solar Efficiency

An off-grid system basically consists of photovoltaic modules (PV), a charge controller, and a battery with a protective device at the output. The charge controller regulates the charging process of the battery and ideally optimizes the energy flow between the solar modules and the battery. The protection device includes load shedding, which disconnects the load in the event of undervoltage to protect the battery from deep discharge.

Low-priced charge controllers use the current source characteristics of the PV modules. They charge the battery with the current that the module is currently providing. The battery is directly connected to



A prototype with 250 W charging power.

the solar modules. When the end-of-charge voltage is reached, the connection is interrupted until the voltage at the battery falls below a threshold value and the charging process starts again. Depending on the load current, there is a certain on/off ratio, which is why these controllers are also referred to as PWM controllers.

A substantially more complex, and therefore more expensive, but highly efficient solution is known as an MPPT controller. MPPT stands for Maximum Power Point Tracking. This means that the controller searches for the operating point of the PV modules at which the highest power is delivered and this power is then used entirely to charge the battery. The MPP is not a fixed point, but depends on external conditions such as solar radiation and temperature. To understand how an MPPT controller works, the characteristics of a PV module must be analyzed.

MPP of a Solar Module

The I-V diagram in **Figure 1**, which shows the solar current as a function of the voltage, very clearly shows a maximum power output at a certain operating point. This is the MPP. The diagram shows the current I_{PV} with a load change from short circuit to no load, each at an irradiation E of 50%, 75% and 100%. To the left of the MPP, the current is

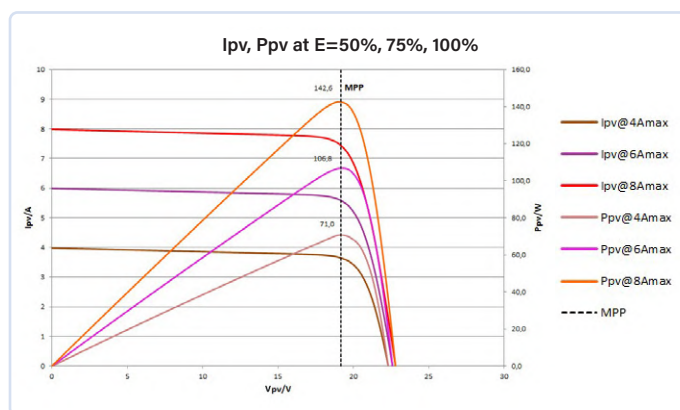


Figure 1: I_{PV} , P_{PV} as a function of solar voltage when the radiation intensity E is varied.

almost independent of the load. In this range, the module behaves like a current source. The current is directly proportional to the irradiation E . To the right of the MPP, the curve resembles that of a voltage source with the open-circuit voltage $V_{PV,OC}$. In very simple terms, a PV module can be regarded as a current source with a Zener diode connected in parallel to limit the voltage. For more information, see [1] and [2].

The diagram also contains the curves of the resulting power, which is obtained from the product $I_{PV} \times V_{PV}$. The clear maximum of the power occurs at the voltage $V_{PV,MPP}$. $V_{PV,MPP}$ is only marginally dependent on the irradiation E . $V_{PV,MPP}$ only increases slightly at very low intensity.

Figure 2, which will be discussed in more detail later, shows the I-V diagram as a function of temperature. A very interesting effect can be seen here. The colder the solar modules are, the more power they generate. This effect is very helpful in winter, when irradiation is low ([3]).

What Does an MPP Tracker Do?

The task of an MPP tracker is to find the MPP and supply the maximum power to the battery by transforming the current. The comparison of a PWM controller with an MPPT controller shows the advantages of the latter.

For this, one starts from the module's specification at 100% irradiation and a temperature of 20°C, for example, $P_{PV,MPP} = 150 \text{ W}$; $I_{PV,SC} = 8.1 \text{ A}$ (short-circuit current); $I_{PV,MPP} = 7.9 \text{ A}$; $V_{PV,OC} = 23.4 \text{ V}$; 36 cells. Dimensions that refer directly to the PV module are labeled with the index PV .

The charging power $P_{\text{Charge,PWM}}$ of the PWM controller is calculated as follows: ($V_{\text{Bat}} \times I_{PV}$). For a lead battery with six cells, we get:

$$P_{\text{Charge,PWM}} = 12.3 \text{ V} \times 8 \text{ A} = 98.4 \text{ W}$$

In contrast, the performance of an MPPT controller at 96% efficiency is:

$$P_{\text{Charge,MPPT}} = 150 \text{ W} \times 0.96 = 144 \text{ W}$$

In this example, the PWM controller is wasting 32% of the power. If the distance from $V_{PV,OC}$ to V_{Bat} is even greater, the loss can be 50%

or more. Provided that the cost expectations allow it, you therefore should always use an MPPT controller.

In short, the MPPT transforms the solar current into a charging current that is just high enough to ensure that the resulting solar current is at the MP point. At this point, the MPP voltage is then reached.

The system is designed so that the solar voltage is always higher than the battery voltage. Therefore, a step-down converter (buck converter) is used for the transformation, whose output current and output voltage can be regulated. It is typical for buck converters that their output current is always higher than the input current. The ratio of input voltage to output voltage is the increasing factor. Assuming that no losses occur (efficiency η approaches 1), the following applies:

$$\begin{aligned} I_{\text{Out}} &= I_{PV} \times (V_{PV}/V_{\text{Bat}}) = (I_{PV} \times V_{PV}) \times 1/V_{\text{Bat}} = P_{PV}/V_{\text{Bat}} \\ P_{\text{Out}} &= I_{\text{Out}} \times V_{\text{Bat}} \\ V_{\text{Bat}} &= \text{const} = c \quad (\text{the battery voltage } V_{\text{Bat}} \text{ is constant}) \end{aligned}$$

$$I_{\text{Out}} = I_{PV} \times V_{PV}/c = (I_{PV} \times V_{PV}) \times 1/c = P_{PV}/c \quad (I_{\text{Out}} \text{ is proportional to } P_{PV})$$

$$P_{\text{Out}} = (I_{PV} \times V_{PV}/c) \times c = (P_{PV}/c) \times c = P_{PV}$$

The remarkable result is that the output current curve is completely identical to the power curve generated by the solar module. This means that the maximum output current and the MPP coincide, i.e., they occur at the same voltage.

In Search of the MPP

To put it very simply, to find the MPP, you need a control loop that regulates the output current to its maximum value. The only available control variable is the output current itself. This means that no fixed value can be defined for the setpoint, which is unusual for a control loop and makes things very difficult. No matter which algorithm you follow, it ultimately comes down to trial and error.

Because the output current behaves like the power, the setpoint of the output current can be increased in steps until the actual current I_{Out} no longer follows the setpoint but decreases. This is the criterion that the MPP has just been exceeded. If I_{Out} decreases, the setpoint current is reduced until the actual current rises again. Subsequently, the system oscillates around this point, which represents the MPP. This sounds simple, but in practice it is subject to a number of problems, as will be seen.

Voltage Method

Another possible solution for finding the MPP is known as the voltage method. As the name suggests, the voltage is regulated to a specific value. The major advantage is that voltage regulation is relatively easy to implement. The diagram in Figure 1 shows that the solar voltage $V_{PV,MPP}$ is relatively independent of the intensity of the irradiation. This is utilized in this process. In most cases, $V_{PV,MPP}$ is specified in the panel's technical data. However, this voltage is highly temperature-dependent, as shown in **Figure 2**.

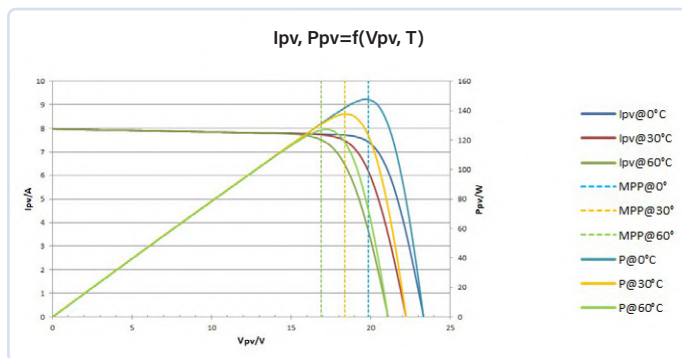


Figure 2: Current and power at constant radiation intensity, at 0°C, 30°C, 60°C.

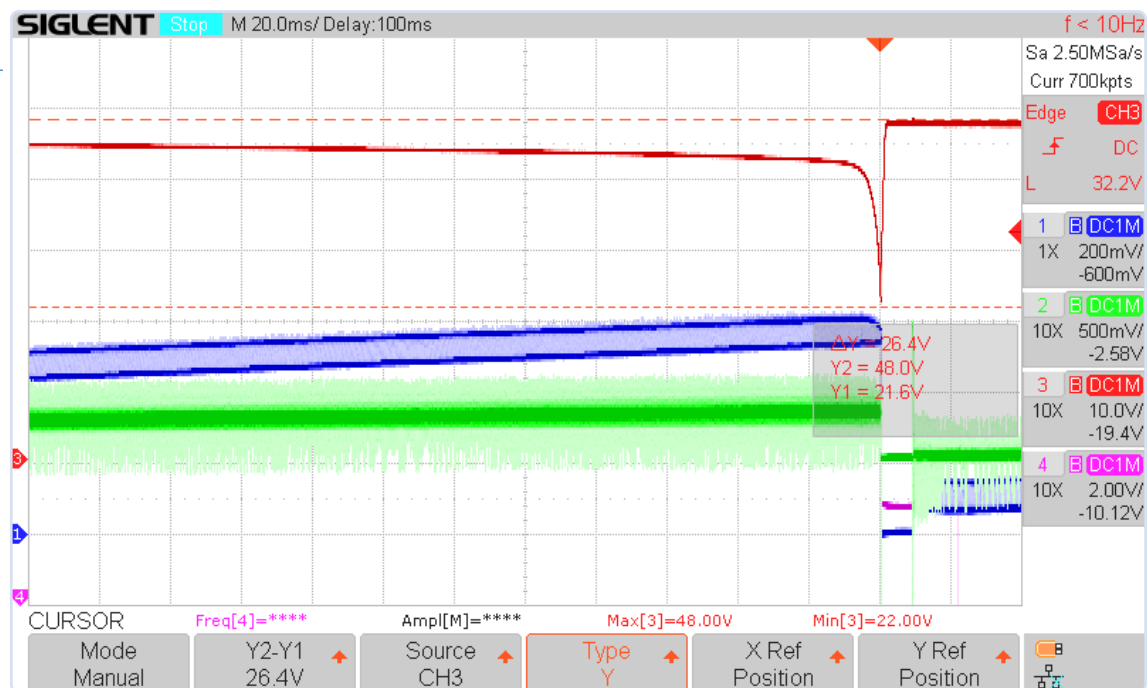


Figure 3: Voltage drop when crossing the MPP (CH1 = blue: Target Current, CH2 = green: Output Current 50 mV/1 A, CH3 = red: Input Voltage, CH4 = pink: Low Voltage Trigger).

This is due to the temperature dependence of the diode voltage. For solar modules, the typical temperature coefficient is -1.6 mV/K per cell. The resulting temperature coefficient of a panel with n cells is $-1.6 \text{ mV} \times n/\text{K}$. The open-circuit voltage changes with temperature by exactly this factor. Fortunately, the MPP shifts in proportion to the open-circuit voltage. Typically, the MPP voltage is 82% of the open-circuit voltage.

The strategy is to measure the open-circuit voltage. From this, a voltage for $V_{PV,MPP}$ is calculated, which is slightly below the 82% of $V_{PV,OC}$ to compensate the scattering of $V_{PV,OC}$ of the modules. This voltage is the first setpoint. Now the current is increased in steps until either the calculated $V_{PV,MPP}$ is reached or the maximum charging current of the battery is exceeded. If you are very lucky, the calculated value will be slightly higher than the actual $V_{PV,MPP}$. Then you are done. However, it is quite likely that the input voltage will suddenly collapse at a certain current value.

The reason for this is that the MPP has been exceeded and the region of the current source is reached. A current source has a very high internal resistance, causing the voltage to drop extremely sharply under load. This effect is intensified by the capacity of the solar panel and the capacitors that are absolutely necessary at the input of the buck converter. For a few clock cycles of the converter, the capacities are still able to maintain the input current of the converter. However, once the stored charge has been dissipated, the converter increases the duty cycle and loads the panel with a higher current. As a result, the operating point shifts towards short circuit and the input voltage drops dramatically within a very short time.

Figure 3 shows a measurement of this effect. The output current is 7 A, the system's cycle time is 4 ms. It can be seen that the system detects the impending collapse shortly beforehand and reduces the setpoint current, but the event can no longer be prevented. Furthermore, extreme noise is evident, caused by the high switching currents ($>20 \text{ A}$) of the switching controller. This makes it very difficult to measure small signals. This behavior has been explained in such detail because

it represents the biggest challenge for every tracking method. The method mentioned at the beginning also usually fails due to this effect.

Search Process

The voltage method makes a virtue out of this necessity. As Figure 3 shows, the voltage dip is several volts and is, therefore, easy to detect. After the occurrence of a dip, the setpoint of $V_{PV,MPP}$ is increased by a small amount, for example by 0.8% of $V_{PV,OC}$, and then a new attempt is started. If necessary, this process is repeated until the system is operating stably.

Figure 4 shows a complete search process at around 20% irradiation, which is a very unfavorable starting condition. The value found is slightly above the actual MPP voltage, but this hardly affects the power yield. Even after reaching the target voltage, the control process is continued by reducing the target current when the target voltage is not reached and increasing it when it is exceeded. In this way, the voltage oscillates around the target value.

Since external conditions such as irradiation and temperature are constantly changing, the system must be able to react to them. To do this, it is necessary to cyclically determine the present value of the open-circuit voltage and initiate a new search process.

Determining the Slope

Another method uses a mathematical relationship. If you look at Figure 1, you can see that the power curve is a function of the voltage, $P_{PV} = f(V_{PV})$ or $I_{Out} = f(V_{PV})$. We know that the slope of a function, also known as the gradient, is zero at the vertex. The function value at the vertex is the maximum and, thus the MPP. The slope of a function is determined by calculating the differential quotient. To do this, the quotient of voltage change and power change is determined at each current step. If the result is 0, the maximum power has been reached. The voltage at this point is $V_{PV,MPP}$. Another sure indication that the MPP has been passed is when the sign of the differential quotient changes.

Since the progression of the output current corresponds exactly to

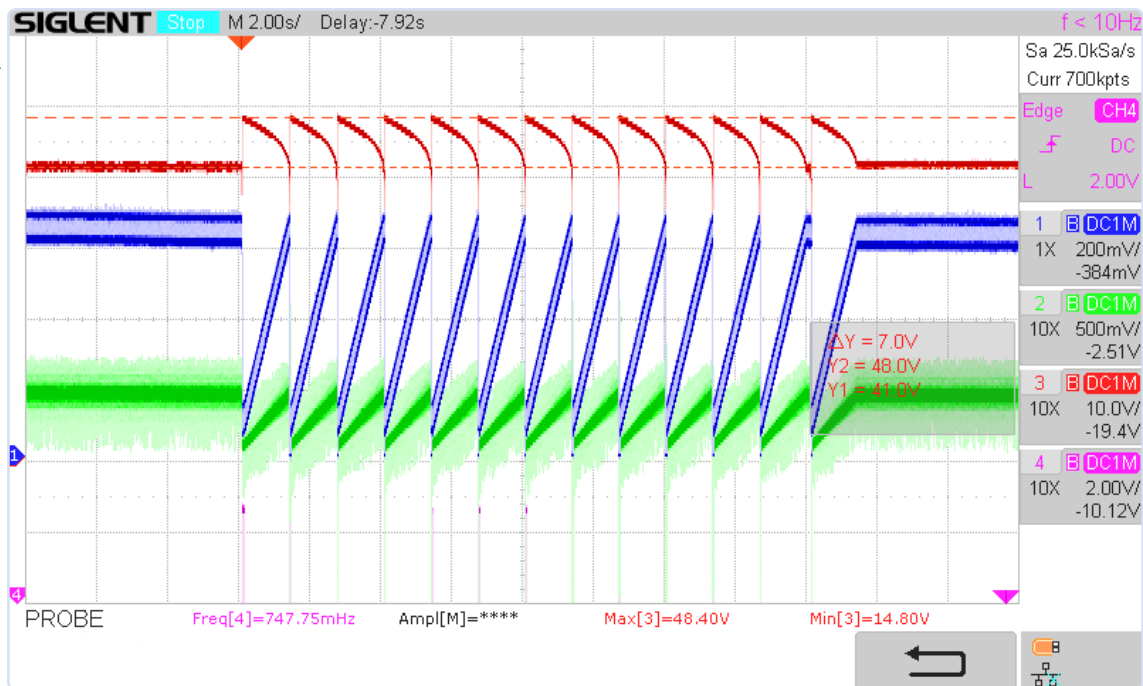


Figure 4: MPP search at $E \leq 0.2 \times E_{max}$; $I_{out} \leq 8 \text{ A}$; $V_{PV,MPP} \approx 0.84 \times V_{PV,OC}$ (CH1 = blue: Target Current, CH2 = green: Output Current 50 mV/1 A, CH3 = red: Input Voltage, CH4 = pink: Low Voltage Trigger).

that of the power, the differential quotient can also be calculated from ΔV_{PV} and ΔI_{out} . The output current is increased successively, starting at 0, and the quotient ($\Delta V_{PV}/\Delta I_{out}$) is calculated at each step. If the quotient is 0 or if its sign changes, then you have reached the MPP. **Figure 5** shows the relationships, including the similarity of P_{PV} and I_{out} .

Due to the limited number of measuring points and the very small signals, the current change at the MPP disappears, which means a division by 0. In practice, only current changes that are greater than 0 are accepted. It should also be mentioned that the problem of voltage dips also occurs here. For this reason, and also because of the division by 0, one tries not to cross the MPP, but only to approach it. In doing so, advantage is taken of the fact that the absolute value of the differential quotient increases sharply just before the MPP, which can be easily evaluated.

In general, to prevent overloading the battery, the incrementing of the output current is stopped when the maximum charging current is reached.

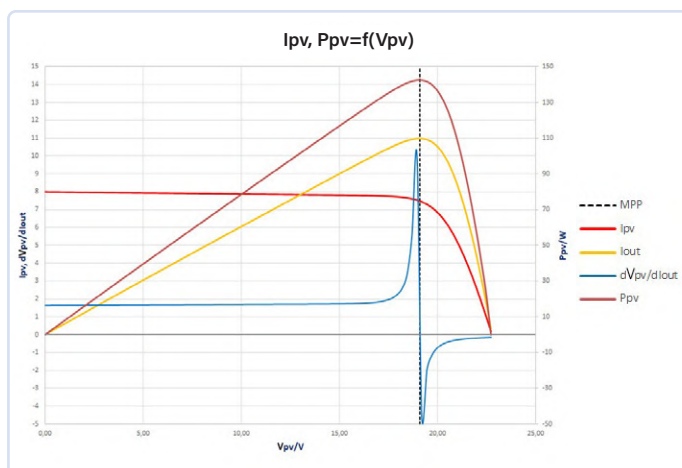


Figure 5: I-V diagram with the differential quotient $\Delta V_{PV}/\Delta I_{out}$.

The methods described can also be combined. For example, the differential quotient can also be formed for the voltage method to accelerate the search or to avoid cyclic searching. Since it is not the voltage that is controlled, but the output current, a diagram with V_{PV} as a function of the output current is more suitable. For more in-depth considerations, **Figure 6** is, therefore, very useful.

Practical Implementation

The first step is to define the main components and their functions. A microcontroller with a suitable A/D converter is used to capture all measured values, to control the hardware, and for visualization. If the computing power is sufficient, the controller can also generate the PWM signal for the buck converter. To do this, two control loops must be implemented in the software. A buck converter transforms the input voltage into the battery voltage and simultaneously regulates the output current. The principle resembles that of a voltage regulator with a controllable current limiter. The output voltage is the charging end voltage, the output current is the charging current of the battery.

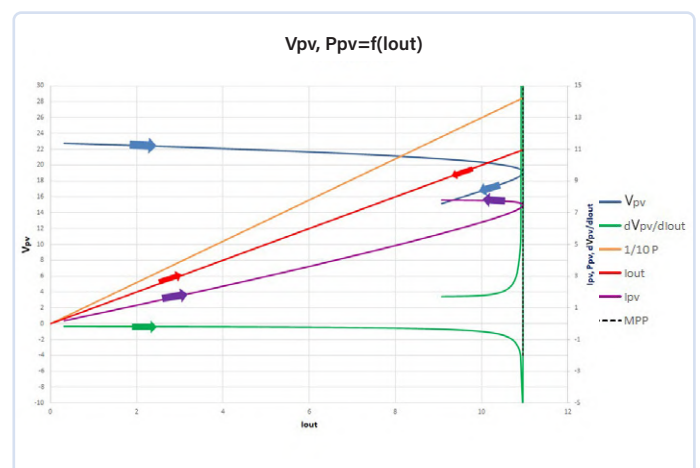


Figure 6: Voltage at the solar module as a function of the output current.

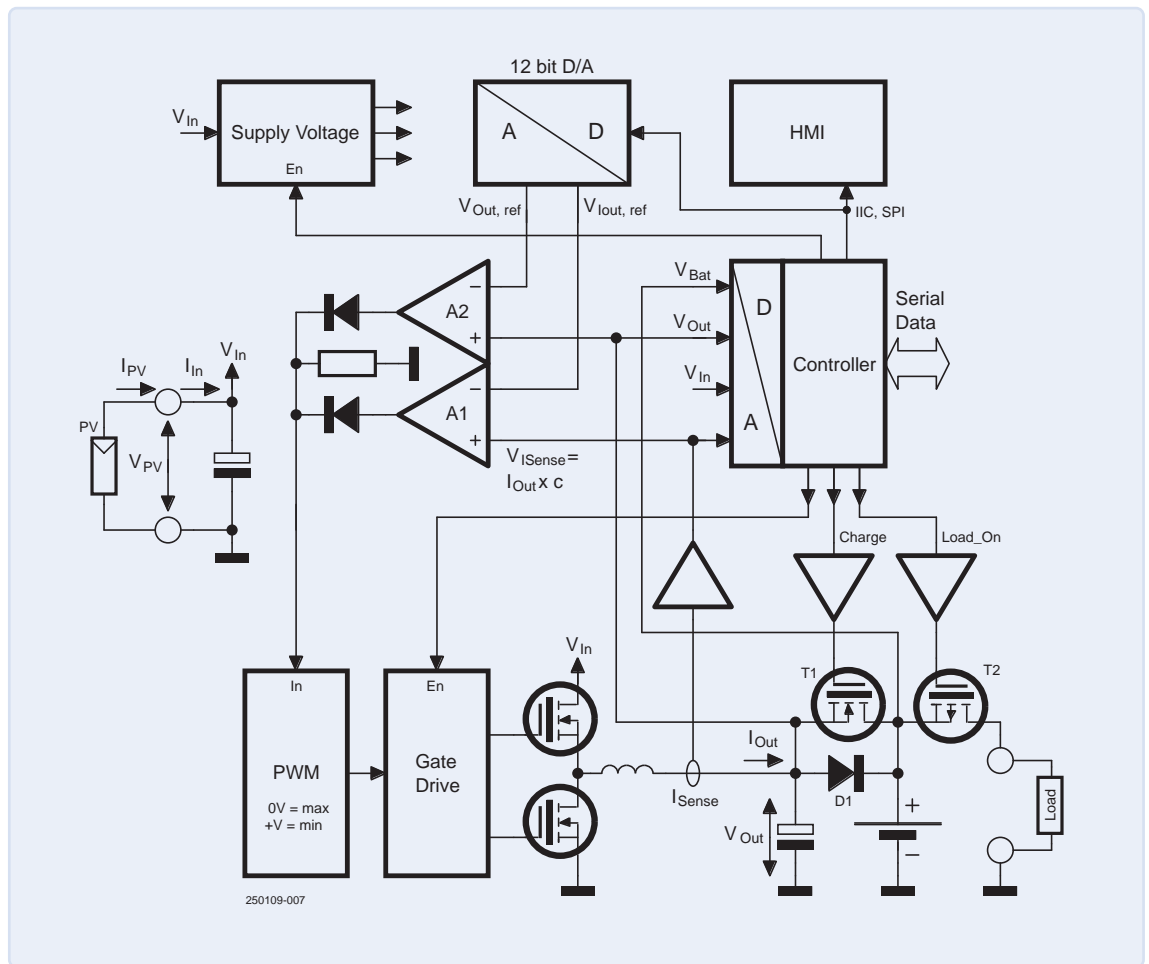


Figure 7: Block diagram of a charge controller with MPP tracking.

The block diagram (**Figure 7**) shows how such a design can look in detail. One immediately notices the two op-amps A1 and A2 that control the PWM block. A1 calculates the difference between V_{ISense} and $V_{IOut,ref}$. V_{ISense} is supplied by the current sensor I_{Sense} and represents the output current I_{Out} . A D/A converter supplies the setpoint $V_{IOut,ref}$. The amplified output signal from A1 controls the PWM block, thus closing the control loop of the output current. A similar principle applies to A2, which calculates the difference between the output voltage V_{Out} and the setpoint $V_{Out,ref}$. The output of A2 also supplies the control voltage for the output voltage V_{Out} to the PWM block. The outputs of the two control amplifiers are connected via two diodes in such a way that the higher output signal always controls the pulse width. Since an increase in the control voltage reduces the pulse width, the control loop that requests the smaller pulse width is always active. So, during charging, the current control comes out on top, even though the voltage control is requesting a high pulse width because the target voltage has not been reached. When charging is complete, the charging current is reduced, and the voltage control takes over. This is exactly the desired behavior when charging the battery.

The PWM block controls the gate driver of the MOSFETs in the half-bridge of the buck converter. A synchronous rectifier is a must

here due to the high power. The current sensor I_{Sense} , which is based on the Hall effect to reduce losses, is located at the output of the converter. As long as the buck converter is inactive, V_{Out} is zero. During this phase, diode D1 prevents the battery voltage from reaching the output of the Buck converter and thus discharging. When the Buck converter is active, MOSFET T1 bypasses the diode and reduces power dissipation.

The load is disconnected from T2 when the battery voltage is too low. This load-shedding protects the battery. The load-shedding is controlled by the controller, which constantly monitors the battery voltage. Therefore, the controller must also be in operation when it is dark. When the controller detects that there is no input power available, it deactivates all non-essential components and switches itself to a low-power mode, in which power is supplied only by the battery. When the load is shed, the controller also switches off, which means that no more current flows.

The "Supply Voltage" block generates the internal voltages for powering the digital and analog components. Step-down converters are used here as well due to the wide input voltage range of 20 V to 60 V. In low-power mode, they are switched off (except for the controller supply) so that no power is drawn from the battery.



The HMI block is responsible for visualizing the operating status. To avoid wasting energy here as well, an LC display must be used. All relevant data is transmitted via the serial interface of the controller. ◀

Translated by Jörg Starkmuth / Edited by Rolf Gerstendorf — 250109-01

Questions or Comments?

Do you have questions or comments about this article? Email the author at 1134-715@online.de, or contact Elektor at editor@elektor.com.



Related Product

- **Waveshare Solar Power Management Module**
www.elektor.com/20488

About the Author

Roland Stiglmayr studied information technology in the 1970s and has over 40 years of experience in research and development. His work focused on the development of computer mainframes, fiber-optic-based data transmission systems, RRHs for mobile communications and contactless power transmission systems. Today, he works in an advisory capacity. Here, he is particularly interested in imparting knowledge.



FEATURED TOPIC

Visit our **Power Electronics & Energy** page for articles, projects, news, and videos.

www.elektormagazine.com/power-energy



WEB LINKS

- [1] H. Föll, "The Real Solar Cell," Chair of General Materials Science, University of Kiel [German] : <https://t1p.de/a9gyy>
[2] Peter Kroll, "Solar Module Simulator," Elektor 1-2/2025: <https://www.elektormagazine.com/magazine/elektor-395/63536>
[3] Elektor Special: Solaranlagen und Photovoltaik, p. 30, Figure 8 [German]: <https://www.elektor.de/20596>

Ignite Your Electronics Innovations with

ElektorLabs

- Free Project Sharing
- Expert Support
- Collaboration Opportunities
- Access to Exclusive Resources
- Get published in Elektor Magazine



Share Your Projects Now!
www.elektormagazine.com/e-labs

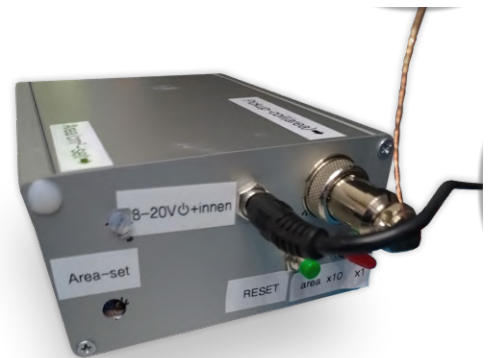


B-Field Integration Magnetometer

With Home-Made Sensors

By Michael Monkenbusch (Germany)

Detecting a magnetic field is a relatively easy task, but measuring its intensity is far more complex! The author of this article describes a project that performs this measurement based on the integration method. The project is covering a wide range of intensities, from a few Gauss up to several Teslas, with the possibility to use self-made, inexpensive sensors.



Looking for a means to measure the magnetic field of single and arrangements of NbFeB magnets, I remembered a kind of "archaic" method, i.e., the integration of the voltage pulse that is induced when a probe coil is placed into the unknown field. In the old days, such measurements were performed by "ballistic" or "creep" galvanometers that reached the (astonishingly high) sensitivity of a few μVs . For further information, you can consult Chapters VI to IX of the ancient — yet very valuable — text at [1]. So, my challenge was to try to realize such a measurement using more modern electronic means, namely an operational amplifier-based integrator.

For the hobbyist, the advantage of this method is the possibility to home-make the required sensors (sensor coils) to cover ranges from a few Gauss (earth field strength is about 0.5 Gauss) up to several Tesla (where 1 Tesla = 10,000 Gauss; 1...2 T is probably the ultimate field strength that can be encountered at home). Whilst the ranges of a few Gauss are, nowadays, commonly available in smartphones, and uncalibrated

cheap Hall sensors extend to a few 100 Gauss, typically further extending the range requires adopting more professional (expensive) equipment.

Besides this, it is also just a fun project to test the limit of what can be achieved with readily available modern electronic parts. Also being educational to understand electromagnetic fields and induction phenomena. Finally, the result is a practically usable magnetometer.

Principle of Operation

Magnetic fields are described by the magnetic field strength, usually denoted by H or the magnetic induction B . The units of H are A/m and the units of B are $\text{T} = \text{Vs}/\text{m}^2$. Outside magnetic materials (i.e. vacuum, air, etc.) H and B are related by the proportionality constant $\mu_0 = 4\pi \times 10^{-7} \text{ Tm}/\text{A}$:

$$B = \mu_0 H$$

In the following, the "magnetic field" term is used as a synonym for B .

If a conductor is moved within a magnetic field (and crosses magnetic field lines) a voltage is induced. This phenomenon is leveraged in several devices, from electric generators to dynamic microphones.

In particular, if a close conductor loop spanning an area A is transferred from a zone without field into a region with field B , a voltage pulse $U(t)$ is induced and may be measured. Before starting and after finishing the transfer, the voltage is zero, but during the transfer, nonzero readings with an arbitrary time function are observed. The key point is that the integral

$$\int U(t)dt = B A$$

or

$$B = \frac{1}{A} \int U(t)dt$$

(Note that this has the proper units: $1/m^2 Vs = Vs/m^2 = T$!)

Technically, the integration $\int U(t)dt = B A$ can “easily” be realized using a combination of one opamp, a capacitor and a resistor.

This is the basic operation principle of the described magnetometer. The loop area A can be — and typically is — increased by using a coil that represents many loops (N = windings) in series with the coil area, thus providing a larger $A = N A_{\text{loop}}$.

Challenges

The main challenge of the presented method pertains the tiny voltages involved. As an example, we may consider a moderate probe coil with 500 turns of 1 cm^2 , i.e. $A = 0.05 \text{ m}^2$ and a moderate field of $100 \text{ Gauss} = 0.01 \text{ T} = 0.01 \text{ Vs/m}^2$ (200 times the earth field). In that case, we expect a pulse with an integral of

$$\int U(t)dt = BA = 0.01 \times 0.05 \text{ Vs} = 0.5 \text{ mVs}$$

If the transfer is done within a second, the average induced voltage during this action is 0.5 mV . Even if this does not look too bad, we want to have time during which the reading is stable enough to represent the field within a (few) percent. If we define this period as, for example, 100 s and set a goal of 1% deviation, we have to reduce offset voltages down to

$$0.5 \text{ mVs} \cdot 0.01 / 100 \text{ s} = 50 \text{ nV}$$

which is a challenge even for the use state-of-the-art

zero-offset op-amp (an OPA2388 by Texas Instruments) for which the datasheet claims a typical offset of $\pm 0.25 \mu\text{V}$ and a guaranteed offset of voltage below $5 \mu\text{V}$. Even if this is not too far off, it still requires extra measures, in particular since there is another source of voltage offsets: thermocouple voltages.

Any contact of different conductor-materials is a source of a temperature dependent voltage (which is utilized when thermocouples are used to measure temperature). Typically, these voltages are of the order of $10 \mu\text{V/K}$ for different materials. This applies, for instance, if resistors with iron leads are connected to copper wires (strictly to be avoided!). Even then, inhomogeneities in wires and gradients across solder joints may contribute, however, to lower values.

Thus, special care must be taken to complete the whole probe coil connection and resistor circuitry at the op-amp with copper only. Further, the critical joints should be thermally shielded and kept to a uniform, constant temperature. A single hetero-material joint would probably only need a temperature difference of a few mK to become annoying!

Furthermore, the selection of the integrating capacitor is critical. This point is discussed in a separate chapter in this article.

Realization

At the heart of the circuit, we have a zero-offset opamp with a very low offset voltage, the probe coil is connected via a resistor to the (current-)integrator input (i.e., the “-” input of the opamp) the capacitor in the feedback loop then integrates the voltage of the probe coil. To further reduce the offset effects, the “+” input of the op-amp is connected to the heavily attenuated output of a DAC, that is set by a microcontroller to the “exact” offset zero setting.

To achieve this, the probe coil must be kept in a reference position and a zeroing sequence is initiated at start (reset) or at an explicit push of a button. During the zeroing phase the integrator voltage is observed via the connected ADC (either directly or amplified by 100 to be more sensitive around zero) and according to the raise during a time interval of about 30 s the DAC voltage is corrected.

This process is iterated several times and hopefully yields a perfect offset compensation that stays valid for the following measurement period. Then the probe coil is (manually) moved to the measurement position and the result may be read off the display. The measured value always represents the field *difference* between the field at a reference position and the field at an actual probe position. For most exact absolute field measurement,

case) are 10 times larger. While these are already excellent values, for ultimate stability of the integrator output a kind of additional "dynamic" zeroing is included, using U3 DAC output (an MCP4725 by Microchip) after a million-fold reduction as corrective offset.

The value of R4 and eventually the choice of connecting it to V+ or to GNDD depends on the individual offset of the actual opamp, and must be determined after assembly. The ultimate adjustment is orchestrated by the ATmega328P microcontroller upon start or push-button request through SW1.

After thermal stabilization, effective residual offsets as low as 50 nV could be achieved. The calibration constant (i.e., the effective area) of the used probe coil is entered using the potentiometer RV1, the wiper voltage of which is read by a genuine analog input of the microcontroller. An imminent improvement of the practical realization could be the inclusion of RV1 into the probe connector, such that it might have the proper value for each different probe.

The relay is used to reset the integrator by discharging C1. The power supply is an arbitrary external 8...20 V which is stabilized to 5 V by U2 and the level splitter U1 supplies a virtual ground GND1 for the amplifiers.

The used opamp has a typical (datasheet) offset voltage $\pm 0.25 \mu\text{V}$ and typical bias current $\pm 30 \text{ pA}$ (translating to $\pm 0.3 \mu\text{V}$ across the $10\text{-k}\Omega$ resistor R11). The limits (worst



The purpose of D2 is to indicate when the zeroing process is complete and the system is ready to start a new measurement. In principle, it could be considered redundant, since the same information can also be inferred from the LCD. The function of the buttons and switches will be explained below.

The realization of the circuit is shown in **Figure 2**. The functional groups are glued/mounted on a copper clad ground plane, ADC and DAC units are preconfigured Arduino shields. Mounting such that all joints and parts in the primary signal channel are at the same temperature is essential.

The LCD, as well as the input voltage regulator (LM317), are attached to housing parts and not shown here. Further, the main integration capacitor C1 has later been replaced by a (mechanically larger) better type.

Choice of Integration Capacitor

Besides the amplifier, a central part that requires special attention is the integration capacitor, in particular regarding a kind of non-ideal behavior. For the present application, it is important that the voltage after injection of a given charge stays constant over an extended period of time (at least long enough to finish the probe coil positioning to the field of interest and the time needed to read off the result).

Furthermore, after discharging, a capacitor should neither exhibit memory effects nor build up a residual voltage. Both

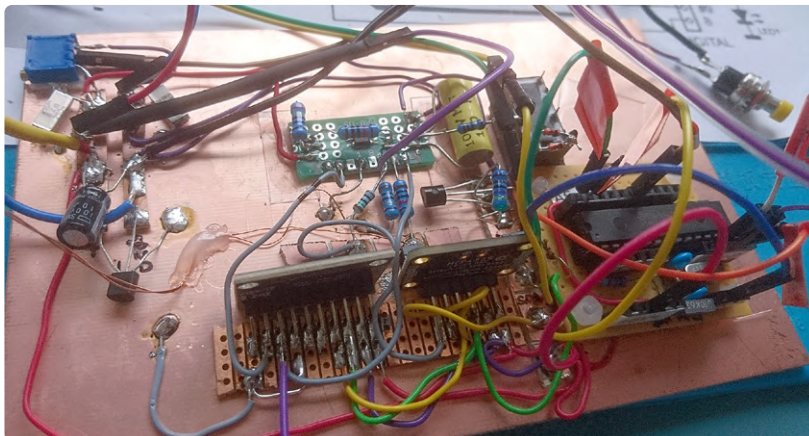


Figure 2: Realization of the circuit before its installation in the housing. Note also that the (yellow) capacitor later was changed to a different type (see text).

effects are connected and commonly occur to more or less extent. They are related to gradual leakage of charge into the dielectric and, after discharging, the gradual reverse flow of the leaked charge out of the dielectric.

As illustrated in **Figure 3**, a good (best) choice would probably be a high-quality high voltage polypropylene film capacitor. To my surprise, the high quality (and expensive) mica capacitors are awful in this respect; however, their high-frequency losses are exceptionally low. But for the present application, this, as well as ESR, are not an issue.

Test Setup

To assess different capacitors, I used the simple circuit of **Figure 3**, the capacitor is first charged to the positive supply voltage (7 V) and then only connected to the "+" input of the very high impedance opamp MAX7642 by Analog Devices and kept there for about 1 h, while the output voltage drop is monitored by a DMM. Then the capacitor is discharged via 1 k Ω resistor connected to ground for about 5 s. Subsequently,

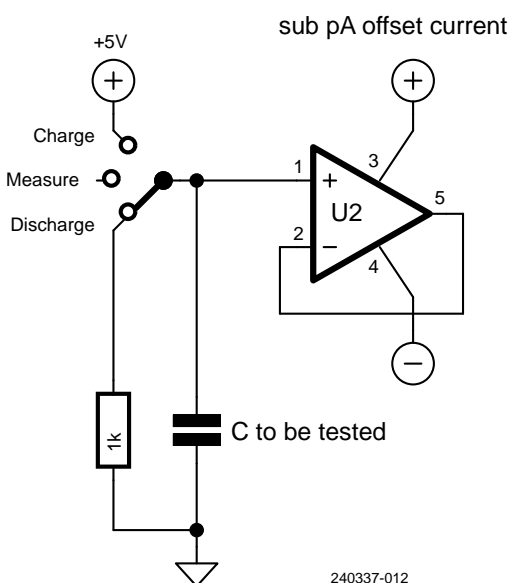
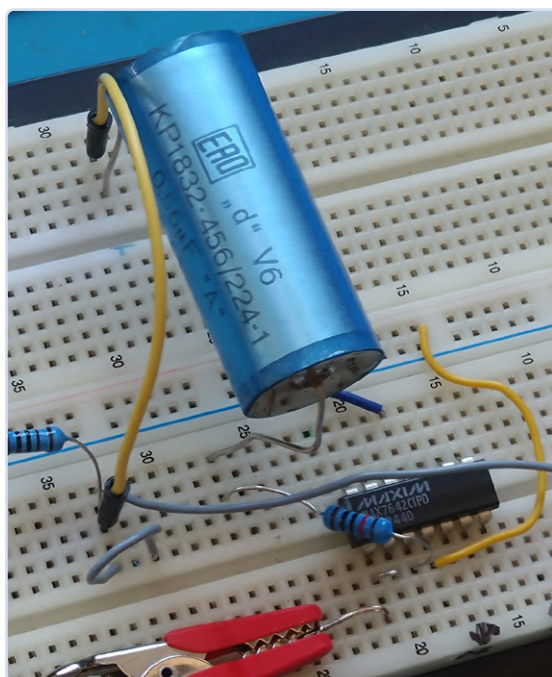
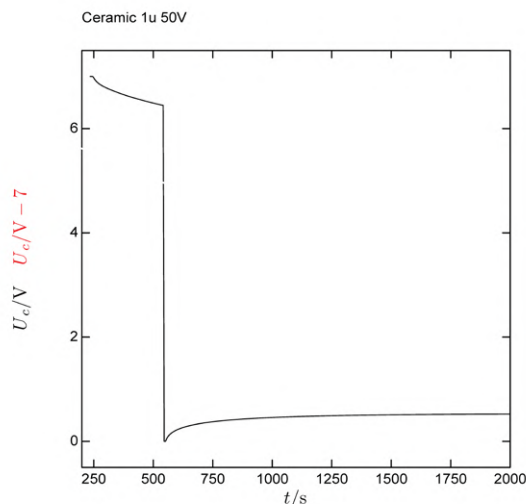


Figure 3: Test setup for the charge loss/buildup of a capacitor. The voltage at the op-amp MAX7642 output is monitored using a six-digit DMM. Left: the setup with the "winning" capacitor installed. In the actual version of the instrument, two of these 560 nF capacitors were connected in parallel.

Figure 4: A small, 1- μ F/50 V capacitor, obviously a bad choice for the integrator.



the voltage rebound is monitored during (about) another hour. I tried different capacitors from my "collection." Some key results are shown in the next paragraph.

Results

Just to illustrate how bad a choice for C1 could be, the test result for a tiny drop ceramic capacitor is shown in **Figure 4**. Within minutes, the voltage changes considerably! In contrast, much better choices, namely high voltage polypropylene film capacitors (here a vintage ERO KP1832, 560 nF or a new WIMA 1 μ F 2 kV) perform much better, as illustrated in **Figure 5**.

Figure 5: Voltage development during test of high-quality, high-voltage film capacitors. Effects are only visible in the enlarged display on the left graph. Best performance was that of the ERO 560 nF (blue = decay from 7 V, green = build-up after discharge).

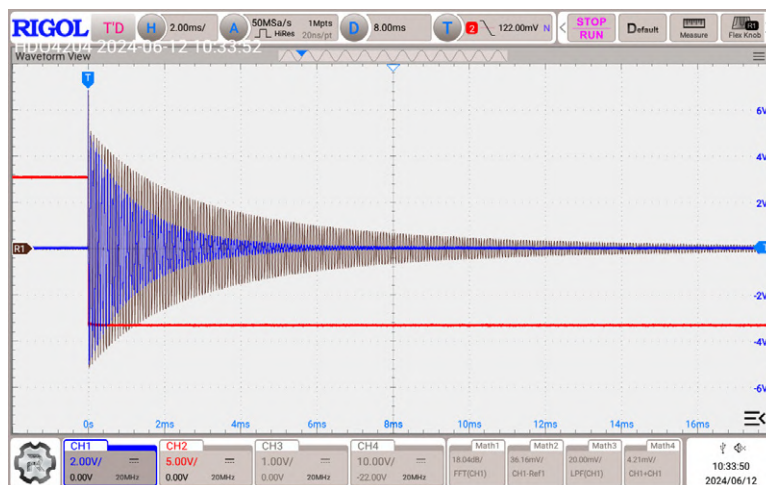
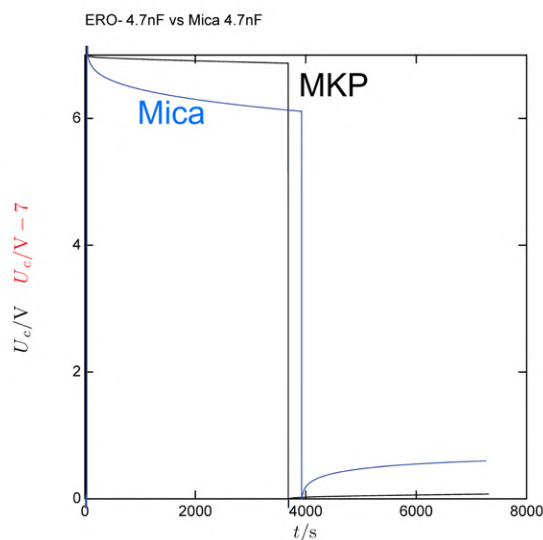
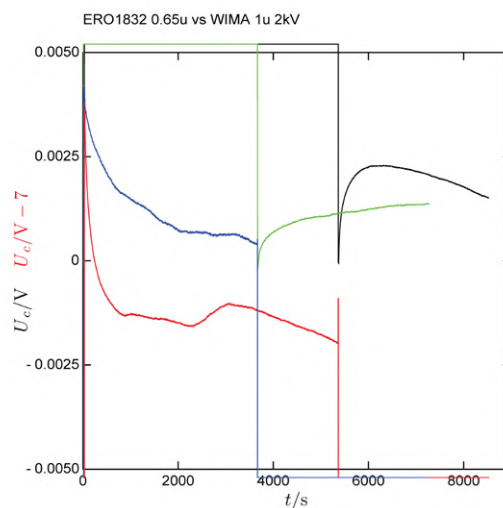
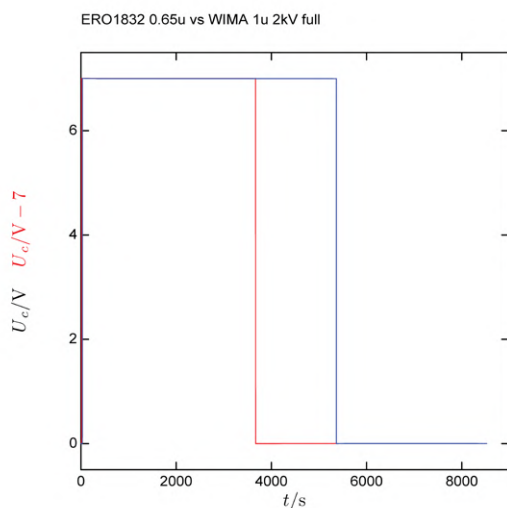


Figure 6: The blue curve shows the test results of a 4.7 nF/500 V Mica capacitor, while the black curve is that of a 4.7 nF/500 V MKP foil capacitor. On the other hand, the right screenshot shows the oscillation damping of the foil capacitor (here blue) compared to the reference trace mica.

A side remark: high-quality mica capacitors perform badly in these scenarios. However, their high-frequency losses are exceptionally low, as shown in **Figure 6**.

Probe Coils

The sensor part of the magnetometer consists of (a number of) probe coil(s). An advantage for the hobbyist is that these can be home-made with relatively little effort. With a bit of care, even before calibration their sensitivity (= effective area A) can be inferred within an accuracy of 10 %.

The principal layout of the probe coils is shown in **Figure 7**. The sensitivity is determined by the total enclosed area A of all windings. For a flawlessly-wound coil this area can be derived from the geometric parameters, in particular the inner and outer winding radii, R_i and R_o , and the number of turns:

$$A = \frac{\pi}{12} \frac{R_o^3 - R_i^3}{R_o - R_i}$$

However, some care must be (of course...) taken on the accuracy of measurements of the effective inner and outer radii, also considering that "hand-wound" coils will exhibit some amount of elliptic windings — which will cause some error in the determination of A . Nevertheless, an accuracy in the range of 10 % should be achievable from scratch. Better values may be obtained by further calibration.

To make a number of coils, I used the following method:

- **Support:** take a gasket ring of suitable size.
- **Side-walls:** glue some plastic sheets to the side walls.

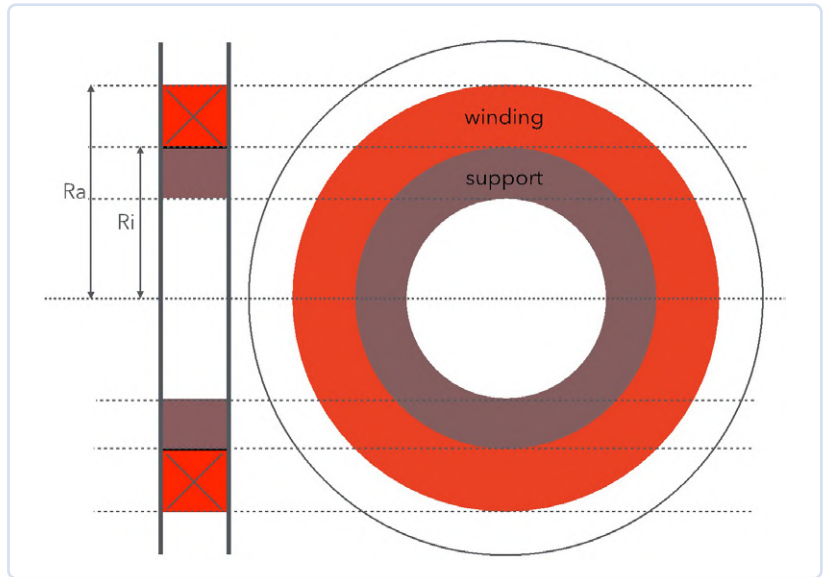


Figure 7 Drawing of the probe coil.

- **Wire:** 0.1 mm ... 0.15 mm diameter CuL.
- **Winding:** using a DC motor with gear and a counter attached as shown in **Figure 8**.

If a 3D printer is available, more professional solutions may be available.

Probe data/sensitivities

The maximum range of the integrator circuit is 20,000 μ Vs = 0.02 Vs. To convert this to field values in Tesla it has to be divided by the probe coil area A . Thus, an area of 100 $\text{cm}^2 = 0.01 \text{ m}^2$ allows measuring field up to 2 T — which is the absolute maximum you would probably ever encounter outside specialized laboratories. Enlarging A reduces the maximum range but increases the sensitivity. With $A = 2,000 \text{ cm}^2$ it is possible to detect field changes as low as 0.01 Gauss.

Avoid thermocouples!

It is essential that the connection from probe coil to integrator are of the same wire type (i.e. copper only). The connector (if used, must be copper). Solder joints should be very localized and thermally shielded.

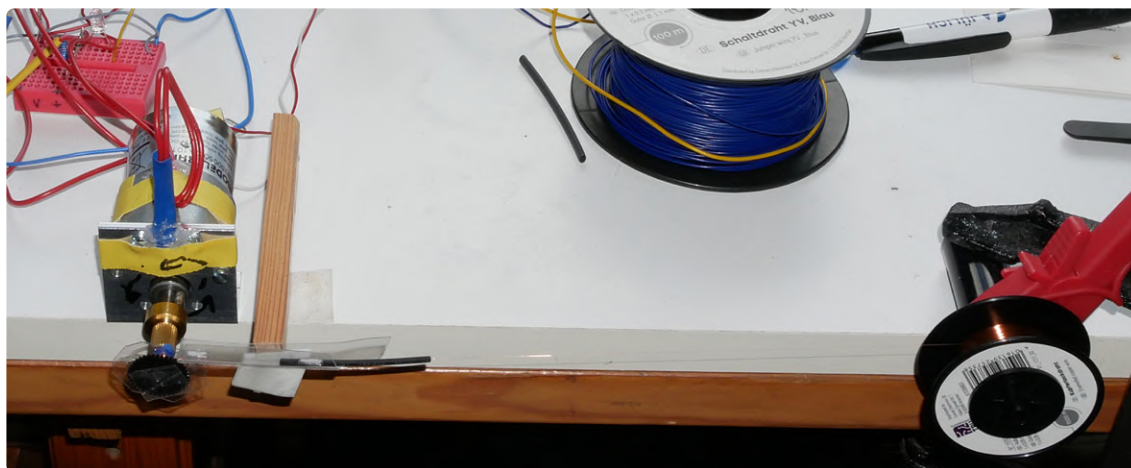


Figure 8: Probe coil winding contraption.

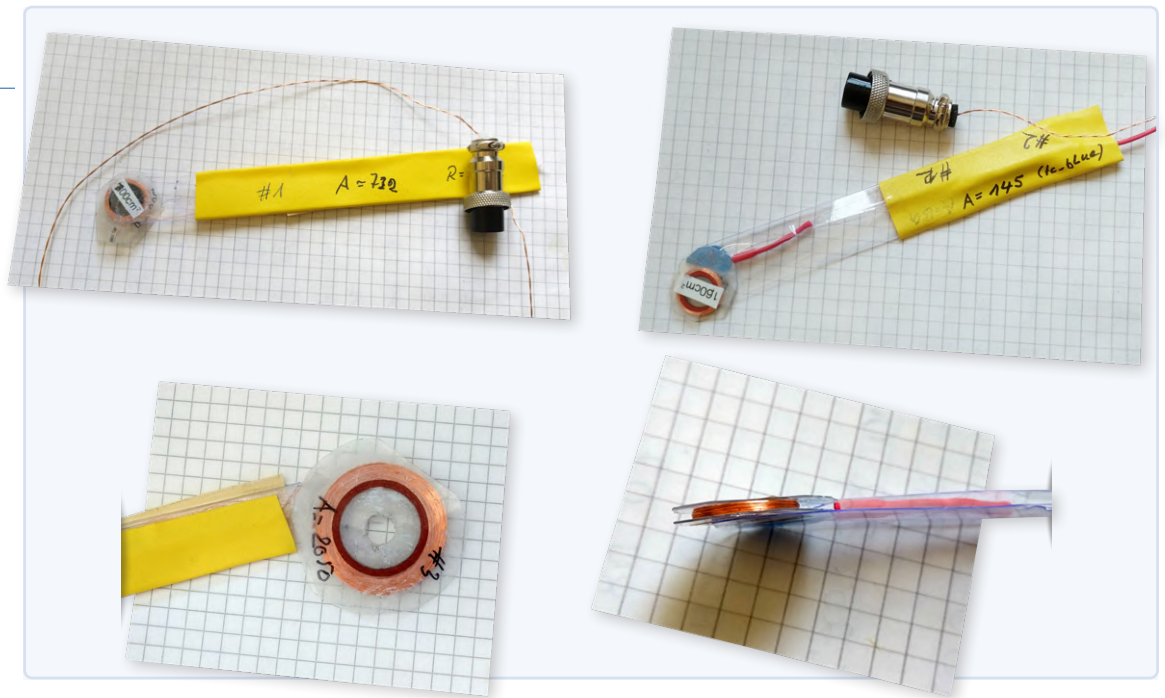


Figure 9: Some probe coils with A from about 150 cm² to 2000 cm².

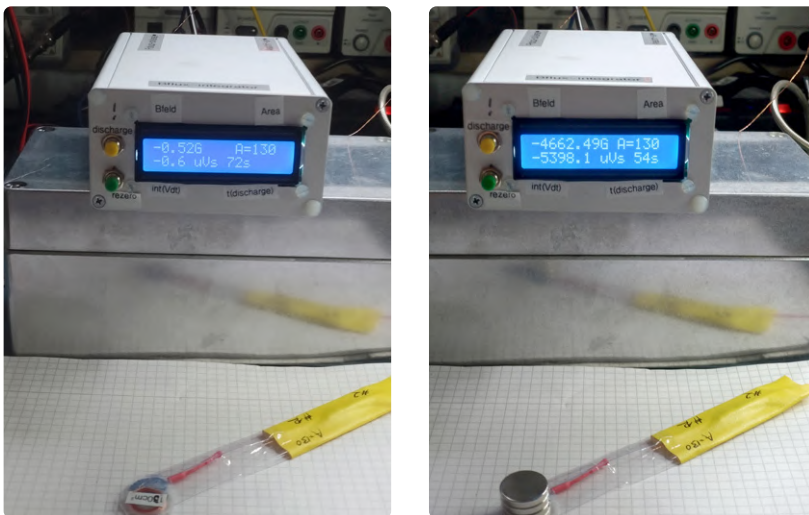


Figure 10: Measurement of the field value at the pole of a stack of strong NdFeB magnets.

Remember: we aim at offset voltages < 50 nV, while typical voltages generated by the coupling of different metals are in the order of 10 μ V/K!

Measurements

As an example here, only the effect of putting a stack of NdB magnets on top of a probe coil is illustrated in **Figure 10**.

Calibration

Even if the geometry data of the probe coils allow the computation of the sensitivity factor A, an independent calibration to account for winding errors and dimensional uncertainties is desirable. Usually, access to a known

standard field is difficult. Therefore, in this chapter, I describe a way to homebrew a known field of 100...200 Gauss that can be used for a decent calibration. Still, the simple described method is not super accurate, but can be improved if a designated test coil is wound.

But here I took the lazy road and bought a 100 m coil of wire (with access to the inner end) and inferred the field from the outer winding geometry and the wire length. My example coil had the following parameters: number of turns $N = 590$, inner winding radius $r_i = 0.018$ m, outer winding radius $r_a = 0.036$ m and total winding length $L = 0.045$ m. With that, the field of the coil can be computed.

Let's consider the "simple" formula for the central field of a cylindrical coil. For such a plain cylinder coil to be used as a calibration aid, one may readily compute the field along its axis, in particular at the center.

For any ratio of length L to radius and inner r_i to outer radius r_a of the winding with a number of N turns, the field at the center of the coil can be expressed as shown in the text box below.

Ideally, the position for the probe coil calibration would be the center of this coil. At the center, the field value is only slightly sensitive to the exact probe position. The inserts in **Figure 11** show how the field changes along the axes and illustrate that in the center of the

$$B(\text{center}) = \frac{NI\mu_0}{4r_a - 4r_i} \ln \left(\left(\sqrt{1 + \frac{L^2}{r_a^2}} + 1 \right) \left(\sqrt{1 + \frac{L^2}{r_i^2}} - 1 \right) \left(\sqrt{1 + \frac{L^2}{r_a^2}} - 1 \right)^{-1} \left(\sqrt{1 + \frac{L^2}{r_i^2}} + 1 \right)^{-1} \right)$$

The field is proportional to the current I through the coil.

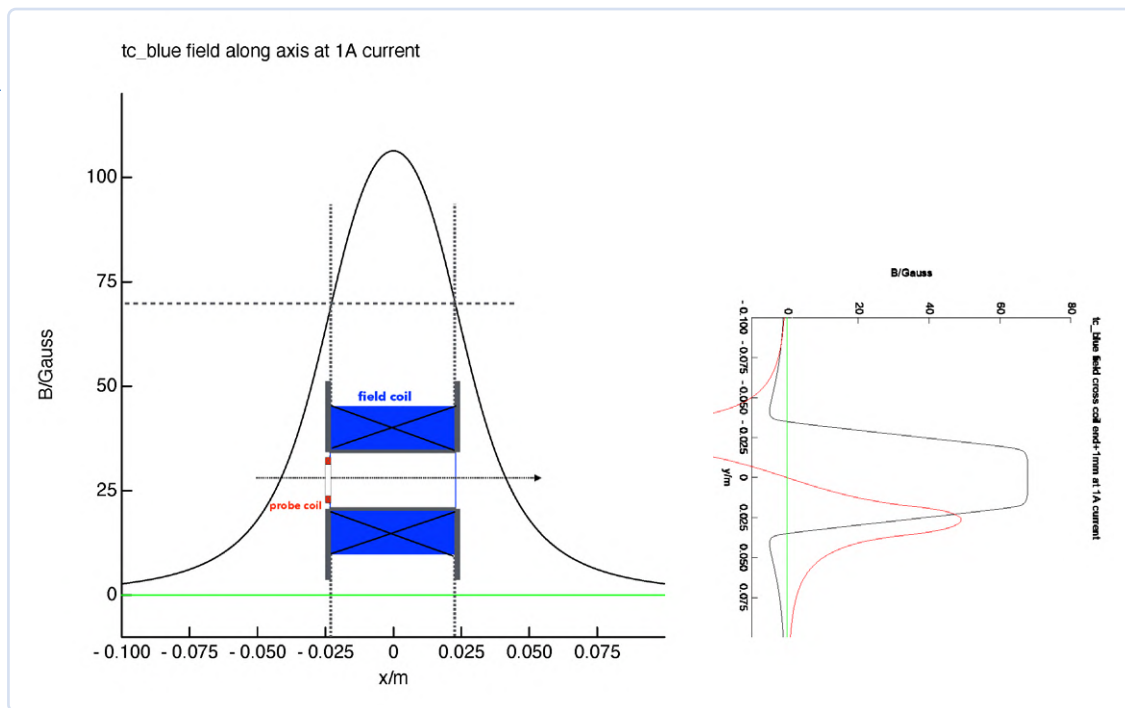


Figure 11: Field of the cylindrical (calibration) coil along the axis for the following coil parameters: $N = 590$, $r_i = 0.018$ m, $r_a = 0.036$ m, $L = 0.045$ m.

coil that field does not change much with small deviations of probe position from the center, and that at the ends the changes due to positioning errors are larger. However, depending on the construction geometry of (my) probe coils, this position is not possible.

Fortunately, if we think of stacking two of the coils, we can conclude that the field at the end of one of the coils contributes exactly one half of the field at the center of the stack of coils, therefore we may use

$$B(\text{end}, L) = B(\text{center}, 2L)$$

to yield the field at the coil end. Unfortunately, there the field is very position sensitive (see Figure 11) and therefore the accuracy of such a calibration is limited.

Inserting the data of my coil in the field expression, one obtains:

$$B(\text{end}, I = 1 \text{ A}) = 0.00705 \text{ T} = 70.5 \text{ Gauss.}$$

Note: if you want to use a different coil, don't be intimidated by the "simple formula" shown above for B, which can readily be evaluated with a pocket calculator with scientific function (ln).

First note values for $a = 1 + (L/r_a)^2$ and for $b = 1 + (L/r_i)^2$, and with

$$f = \frac{(a + 1)(b - 1)}{(a - 1)(b + 1)}$$

we can come to:

$$B = \frac{NI\mu_0}{4r_a - 4r_i} \ln(f)$$

All dimensions must be entered in meter, current I in Ampere, then the field is given in Tesla $T = \text{Vs/m}^2$ (1 Tesla = 10,000 Gauss and $\mu_0 = 4\pi \times 10^{-7} \text{ Tm/A}$).

Program

The program for an Arduino-type microcontroller — available at [2] — takes care of exact zeroing of the integrator, initiating integrator resets (i.e., capacitor discharge), occasionally re-zeroing, and finally performs the conversion of the integrator results to magnetics field values, with their display.

Upon start, first a sequence of five iterations is executed, to fine tune the offset voltage such that the result stays constant within a preset 20 s integration period. Thereafter, further zeroing sequences can be manually initiated by pressing SW3 button connected to D6 (GPIO6).

Multiple push-button actions increase the integration time for zeroing, by increments of 10 s up to a maximum of 100 s; after that, the selection restarts at 10 s. In case of no use of that button, a re-zeroing sequence is automatically performed every 1000 s.

During the whole zeroing period, the sensor probe must be kept in a reference position at a constant field! If available, a mu-metal zero chamber would be ideal for that, but keeping the position fixed in the ambient field is also ok, the absolute field values may then have an offset < 1 Gauss, due to the earth field. For best results, one may wait for full temperature stabilization before measurements.

Plain zeroing of the displayed values (i.e., integrator reset) is initiated by pressing SW1 push-button connected to D7 (GPIO7). The buttons must be pressed long enough for the action to be accepted.

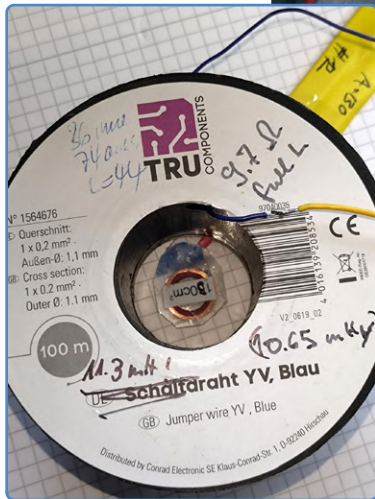


Figure 12: Calibration attempt using the wire reel as "calibration coil" at a current of 2 A (for a few seconds only!). Left: the probe coil is positioned at the center of the coil end.

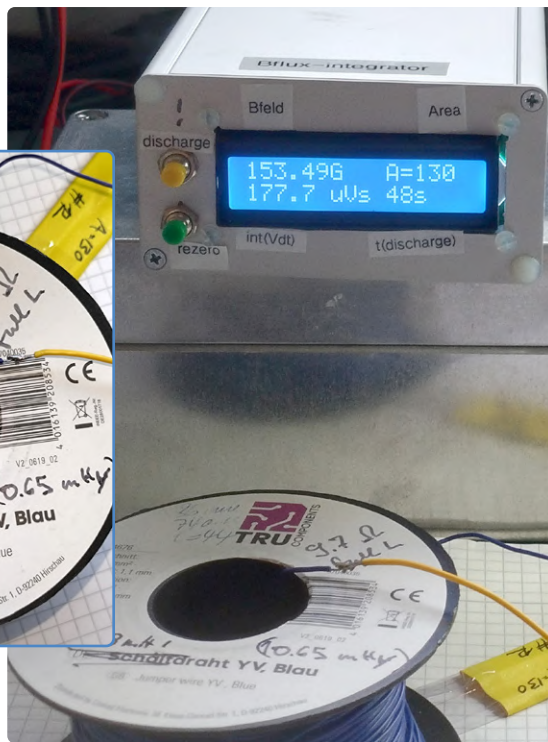


Figure 13: LCD during last iteration of zeroing, short display of new and previously applied offset corrections, zero state (no applied field) after some time (72 s) after discharge and value probe moved to pole (pole of a NdFeB magnet) at 22 s after last discharge.

Probe area A (sensitivity) for the used probe coil can be communicated to the program by setting the potentiometer RV1 such that the A value shown on the display (in cm^2) coincides with the value of the probe coil. The standard range is 1...1024. To enable also larger A values, a switch (SW4) connected to GPIO3 must be toggled, to extend the maximum value up to 10,240.

The display is a standard 2×16 LCD, if a 4×20 display is connected, the comment in front of the **large display** definition should be removed. Then a bit more internal information is shown.

But the small display contains all the necessary output, it:

- Informs when zeroing is under way (zero count down).
- Shows magnetic field values and the sensitivity factor A (first line).
- Shows the actual plain integral in μVs and the actual integration time in s. ◀

240337-01

Questions or Comments?

Do you have technical questions or comments about this article? Please contact the author at michael.monkenbusch@googlemail.com or the Elektor editorial team at editor@elektor.com.



About the Author

Michael Monkenbusch is a retired physicist who worked in the fields of neutron scattering, instrumentation, and soft-matter physics. Reviving an old electronics hobby led to the project presented here.



Related Products

- **PeakTech 3440 True RMS Graphical Multimeter**
www.elektor.com/18157
- **AE970D Soldering Station (80 W) Incl. Extra Soldering Tip**
www.elektor.com/20731

WEB LINKS

- [1] R. W. Pohl, "Einführung in die Elektrizitätslehre", Springer-Verlag, 1927: <https://tinyurl.com/44zrey6p>
- [2] Elektor Labs page for this project: <https://tinyurl.com/ycy689kk>

FREE Download Now!
Mac, Windows and Linux

Blackmagicdesign



Edit and color correct using the same software used by Hollywood, for free!

DaVinci Resolve is Hollywood's most popular software! Now it's easy to create feature film quality videos by using professional color correction, editing, audio and visual effects. Because DaVinci Resolve is free, you're not locked into a cloud license so you won't lose your work if you stop paying a monthly fee. There's no monthly fee, no embedded ads and no user tracking.

Editing, Color, Audio and Effects!

DaVinci Resolve is the world's only solution that combines editing, color correction, visual effects, motion graphics and audio post production all in one software tool! You can work faster because you don't have to learn multiple apps or switch software for different tasks. For example, just click the color page for color, or the edit page for editing! It's so incredibly fast!

Professional Editing

DaVinci Resolve is perfect for editing sales or training videos! The familiar track layout makes it easy to learn, while being powerful enough for professional editors. You also get a library full of hundreds of titles, transitions and effects that you can add and animate! Plus, DaVinci Resolve is used on high end work, so you are learning advanced skills used in TV and film.

Creative Color Correction

DaVinci Resolve's color page is Hollywood's most advanced color corrector and has been used on more feature films and television shows than any other system! It has exciting new features to make it easier to get amazing results, even while learning the more advanced color correction tools. There's PowerWindows™, qualifiers, tracking, advanced HDR grading tools and more!

Designed to Grow With You

DaVinci Resolve is designed for collaboration so as you work on larger jobs you can add users and all work on the same projects, at the same time. You can also expand DaVinci Resolve by adding a range of color control panels that let you create unique looks that are impossible with a mouse and keyboard. There's also edit keyboards and Fairlight audio consoles for sound studios!

DaVinci Resolve 19 **Free**
DaVinci Resolve Micro Color Panel **Only £425**



Learn the basics for free then get more creative control with our accessories!

Learn more at www.blackmagicdesign.com/uk

SRP is Exclusive of VAT

Download free on the
DaVinci Resolve website



NO SUBSCRIPTIONS • NO ADS • NO USER TRACKING • NO AI TRAINING

Precise or Accurate?

Your Instruments Need to Be Both!

By Roberto Armani (Elektor)

When editing or translating engineering texts, it often happens that the authors use the definitions “precision” and “accuracy” interchangeably, as if they were synonyms. They are not! In this article, we will try to bring some clarity to the terms, with a few practical examples and a (very) minimum of calculations and formulas.

In writing technical articles, we often risk losing some language accuracy, and this becomes more evident in rigorous fields — such as metrology — where each term has a very specific usage and, generally, inappropriate terminology is not well-tolerated.

Measurements

A measurement is, by definition, a numerical relationship resulting from the comparison of a given quantity to another, homogeneous to it and of known value, considered as a *reference*.

Regardless of the unit of measurement, this measurement, comparison, and the resulting calculations cannot be error-free. Error is inherent to measurement; it can be reduced, but not eliminated. What can be done — to live with it — in the technical and scientific sphere, is to quantify its magnitude and consider it within the context of the overall measurement process.

Generally, in the T&M field, quality pays: The finest and most expensive measuring instruments can help reduce the margin of a measurement error, thus improving the quality of our work. However, this comes at a cost; technicians and engineers must



evaluate, before purchase, the quality of the instrument they are going to buy, perhaps resorting to trade-offs between price and performance.

Terms such as *precision*, *accuracy*, *repeatability*, *long-term stability*, *sensitivity*, *resolution*, *readiness*, and *full-scale value* have become commonly used and can help one make a choice among the plethora of the devices available off the shelf.

However, in this scenario, two of them are the most illustrious victims: precision and accuracy, which are often used interchangeably and improperly (which does not necessarily cause much harm).

An Initial Definition

What do Accuracy and Precision really represent in metrology? Let's start with two concise definitions:

- **Accuracy:** An instrument is defined as *accurate* if it is capable of providing readings consistently as close as possible to a *reference value*, or *true value*.
- **Precision:** An instrument is defined as *precise* if it is capable of providing readings that correlate as consistently as possible with the *average value* of the readings taken.

Having briefly defined them, we must note something important: With only one reading available, we can define the accuracy of an instrument (albeit approximately), but not its precision, for which we need an average value or, possibly, a number such that the calculations are statistically significant and provide a reliable result.

Before going further into detail, however, it will be worth spending a few words defining the concept of *error*.

Measurement Errors

In a *single measurement*, error can be defined as the difference between *measured value* X and *true value* θ (or *reference value*, if we assume it as being *true*) of the quantity under consideration. In this case, it's generally defined as an *absolute error*, or $\eta = X - \theta$.

The absolute error value is merely part of the information, as it doesn't give us any clue as to the ratio of this error versus the quantity we are comparing. That's why the magnitude of an error is often expressed as the ratio of absolute error η to the true value θ , thus defined as *relative error*, or $\epsilon = \eta / \theta$.

This error can be calculated only if the reference value, θ , is known. The value of ϵ is usually expressed as a percentage, by multiplying it by 100. For this reason, the percentage error is generally indicated for each individual range of a measuring instrument.

Types of Error

There may be many causes of errors in an experimental measurement, including procedural ones (operator errors, for instance). With analog measuring instruments, in particular, operator error becomes more significant, as described in the text frame **Analog vs. Digital Readout: Resolution Matters!**

Here, however, we will assume that there is no operator error and that any errors present depend mainly on random factors and on the device's features.

We may divide these errors into two macro-categories:

- **Systematic errors**, which constantly bias measurements in one direction. This type of error can be reduced or eliminated, in some cases. It mainly affects the *accuracy* of an instrument.
- **Random errors**, determined by the technical properties of the instrument, the quality of its design and on the components used, and by all the unpredictable and/or uncontrollable

events that might occur during measurement. This type of error, which affects mainly the *precision* of an instrument, cannot be eliminated, but it can at least be estimated by repeating the measurement several times and analyzing the resulting statistics.

In an electronic measuring instrument, systematic errors can be caused, for example, by long-term variations in the characteristics of the components used (derating), or by short-term variations, such as an out-of-range device operating temperature. On higher quality equipment, one or more calibration options are usually provided to reduce the magnitude of these errors — either via hardware or in software.

Random errors, on the other hand, can be introduced by electromagnetic interference (EMI) conducted by probe cables, or by the inherent instability or poor quality of the signal conditioning and/or sampling front-end. They can be partially reduced by shielding, but nothing can be done to correct a design that is fundamentally deficient or poorly functioning.

A First Practical Example

Let's suppose to have 4 multimeters on our laboratory shelves and we want to check their performance in terms of reliability. In the lab, we have a high-end reference current generator available, and we set it up for a preliminary *go-no-go* test on one of those instruments.

To run the test, we set the current generator to 2.000 A, and we consider this value as *true value*, *reference value*, or θ .

Analog vs. Digital Readout: Resolution Matters!

The resolution of a measuring instrument is the smallest variation in the measured quantity that the device can reliably detect and display.

In the example in the image, a vintage Cassinelli & C. analog milliamperemeter (circa 1960) and a latest-generation Siglent SDM3055 digital multimeter (2024) are connected in series to a current generator set at 30 mA. The upper image shows the 0...50 mA dial of the vintage instrument — whose accuracy, BTW, is absolutely remarkable, despite being more than 60 years old! — with 1 mA unit large divisions and 0.5 mA small divisions. Thus, the resolution of this instrument is 0.5 mA (500 μ A).

Here, the critical aspect is in assessing (optical targeting) the position of the needle in the areas between the divisions, with evaluations that are unavoidably personal, arbitrary and widely variable from operator to operator.

With the digital instrument, the display gives a reading of 29.690 mA on five digits — where the least significant digit represents the smallest quantity that the instrument is capable of measuring and displaying, i.e., 0.001 mA (1 μ A). It goes without saying that resolution does not allow for any conclusions about accuracy or precision!



The Gaussian Distribution, in Short

Often in science, the most important discoveries are the result of a combination of the study of several scientists. This is the case of the *Gaussian Distribution*, whose concept arose from various observations, starting in the late 1700s with Laplace's fundamental central limit theorem [3], to follow with the insight of Abraham de Moivre who noticed that as the number of events increased, their distribution followed a rather smoothed profile, until the work of Carl Friedrich Gauss who defined the formula for calculating the curve that later took his name.

An interesting historical overview describing this course of study, which is the basis of modern statistics, can be found here [4].

Figure A illustrates a typical Gaussian distribution curve. In a normal distribution, within the analyzed sample there are values that symmetrically approach the μ value — the arithmetic mean of all the data — with a greater frequency than others, and this frequency contributes to giving the curve its characteristic bell-shaped form.

The area below the curve represents the distribution of the sample data. The Y axis indicates the frequency with which data of the same value occurs.

The distance between μ and the inflection point of the curve (where the curvature change in the other direction) represents what is commonly referred to as *standard deviation* or *sigma* (σ), *root-mean-square deviation*, or *root-mean-square error*.

In other words, sigma can be considered as an index of the dispersion of our data in respect to the average μ . Sigma and precision are inversely proportional, i.e. when the dispersion of the data decreases, their precision increases and the Gaussian

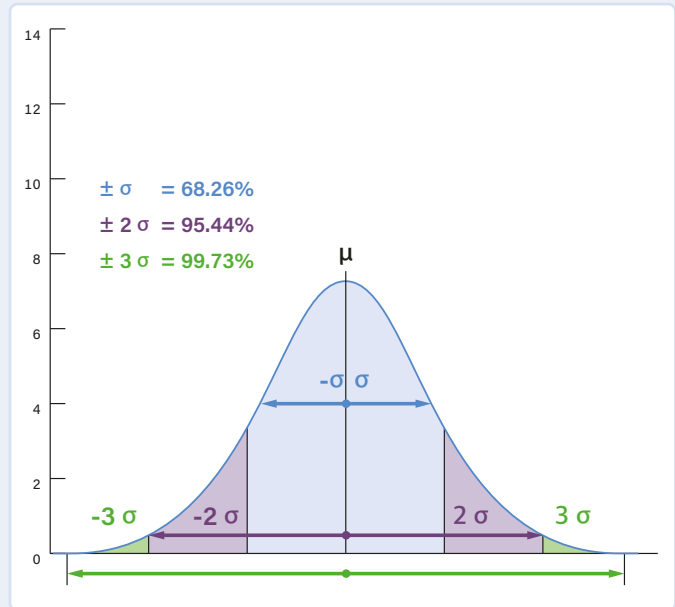


Figure A.

curve becomes peakier. Conversely, with high sigma values, the curve becomes particularly flat and its base gets wider.

An empiric rule that's commonly applied to a standard distribution curve states that the sample data have a 68.26% probability of falling in the $[-\sigma... \sigma]$ interval (blue area, see Figure A), 95.44% in the $[-2\sigma... 2\sigma]$ interval (blue + purple areas) and 99.73% in the $[-3\sigma... 3\sigma]$ (blue + purple + green areas).

Sigma is not a dimensionless quantity, rather, it assumes the unit of measurement of the quantity under consideration (in the case of our four multimeters example it is ampere/A).

Suppose that our first (and only) reading from that multimeter — from now on, defined as *measured value*, or X — indicates 2.0052 A. Then, as anticipated, we have an absolute error:

$$\eta = X - \theta$$

$$\eta = 2.0052 - 2.0000 = 0.0052 \text{ A}$$

Knowing η , we can then calculate relative error ε , which, expressed as a percentage, will be:

$$\varepsilon = (\eta / \theta) \times 100$$

$$\varepsilon = (0.0052 / 2.0000) \times 100 = 0.26\%$$

This figure represents a dimensionless number (i.e., without an associated unit) and means that our multimeter, in this specific reading and selected range, shows a relative error = 0.26%.

Building Statistics

The single measurement we've done so far on the first of our multimeters does not tell us much about the overall reliability of our instrument, nor does it allow us to determine whether it is affected by systematic errors, random errors, or both. So, for more meaning-

ful insights, we take the matter more seriously and build up four databases of measurements.

We perform 100 consecutive readings on each multimeter, taking care not to vary the test conditions between one measurement and the next on any of the instruments. Each one of the measurements generates an X_n value.

Then we calculate the arithmetic mean of those readings and define it as the *average measured value* μ (with $n = 100$, in this case):

$$\mu = (X_1 + X_2 + X_3 \dots + X_n) / n$$

Knowing μ , however, is not sufficient; although it gives us some clues about the average *accuracy*, it does not allow us to quantify the error's random component, thus preventing us from determining the *precision*.

Therefore, on the same acquired data, for each of the four databases, we assume that our measurements follow the law of normal distribution and calculate the standard deviation, σ , of readings $X_1... X_{100}$, using the following formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

So, we can check how far the readings fall from the mean value obtained previously.

If you are not familiar with the concept of standard deviation, you can either take a look at [1] or, for a more formal, mathematical approach, at [2]. You may also take a look at **The Gaussian Distribution, in Short** insert in this article.

Interpreting Data

Multimeter 1

With the hundred measurements taken with multimeter 1, we calculate the respective average value μ and standard deviation σ as mentioned above. The resulting chart is shown in **Figure 1**. The curve shows the normal distribution of measurements around the mean value, μ , which for this instrument was found to be $\mu = 1.9$. The plot of the distribution around μ is very “sharp,” indicating a high concentration of measurements with values very close to the average. However, this is distant from the true value $\theta = 2$. The value $\delta = (\mu - \theta) = -0.1$ is the expression of the *average inaccuracy* of this instrument.

This is a typical case in which calibration of the instrument can help to reduce even substantially this error component, lowering δ to negligible values (close to 0) and restoring its expected accuracy.

The interval between the two worst readings (the base of the

bell-shaped curve, indicated by the red arrow) is also referred to as the maximum *measurement uncertainty*. In measurements, uncertainty and precision are inversely correlated.

With $\sigma = 0.042$ A, (i.e. a standard error of $\pm 2.2\%$ on average μ), it can then be concluded that the instrument is precise, but being the systematic component of the error ($\delta = -0.1$ A) high, it's not accurate, with an average error = -5% to reference.

Multimeter 2

The chart of the second multimeter (**Figure 2**) shows a drastically different scenario. The mean value μ of the readings was found to be $= 1.998$, but the maximum measurement uncertainty of this instrument is far greater than multimeter 1.

With $\sigma = \pm 0.116$ A ($\pm 5.8\%$ error on average μ), the random component of the error is large, with highly distributed (not concentrated) measurement values. Since it cannot be completely corrected, this random error component makes multimeter 2 unprecise, although its apparently astonishing accuracy $\delta = -0.012$ A (-0.6% error on reference) might be misleading us.

All in all, a very accurate yet widely imprecise instrument that cannot be used in the laboratory, as the probability of obtaining unrealistic measurements is high. Using a different, yet common terminology, this instrument shows a very low *repeatability*, a synonym for precision.

This case shows how accuracy alone is not enough for obtaining a trustworthy measurement.

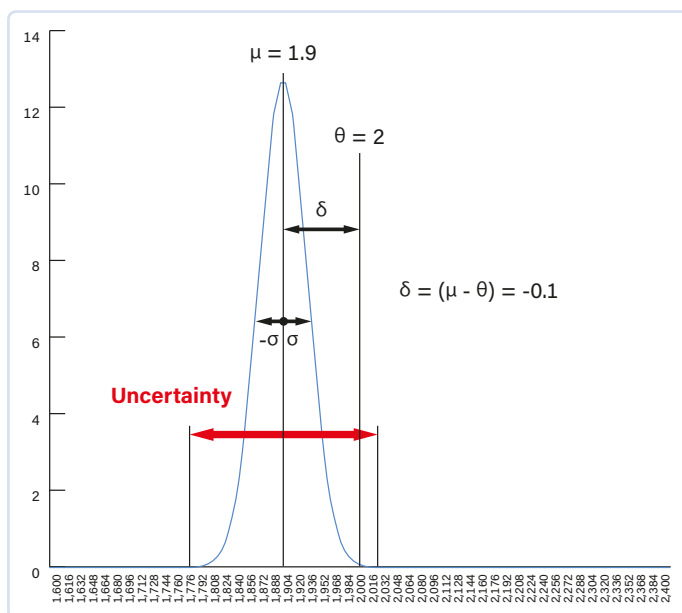


Figure 1: Multimeter 1 shows to be potentially a good measuring instrument. It just needs to be calibrated using a reference instrument, since its δ component is not negligible.

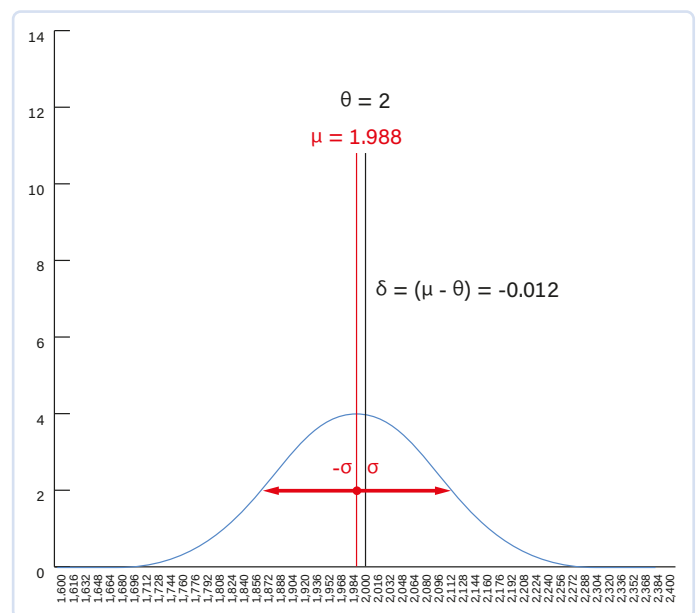


Figure 2: This multimeter's high level of accuracy is rendered completely useless by the significant error in precision.

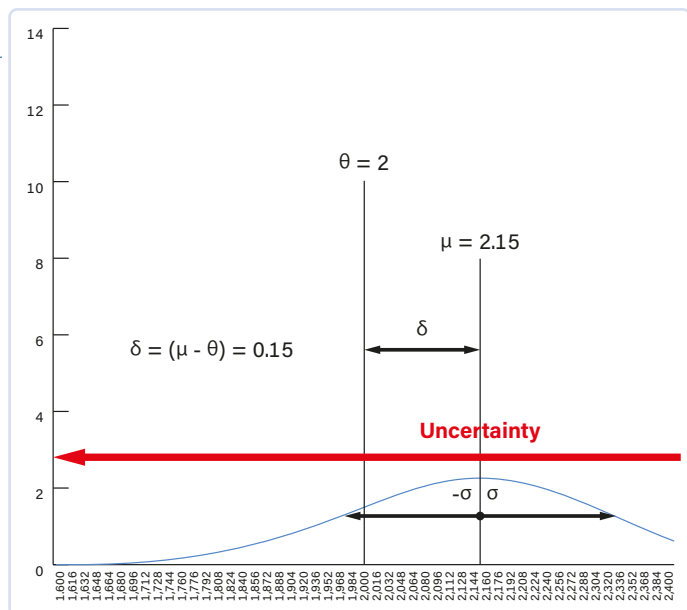


Figure 3: Where's the bin? This instrument shows a relevant bias of +7.5% on reference, worsened (if possible) by a widely spread uncertainty (very high σ).

Multimeter 3

The situation with multimeter 3 is even worse, if that's even possible. The graph of **Figure 3** indicates a calculated mean value $\mu = 2.15$, with $\delta = 0.15$ A (+7.5% error on reference) that could well be significant on its own. Furthermore, with $\sigma = \pm 0.19$ A ($\pm 8.8\%$ error on average μ), the distribution curve reveals a random error with an extremely flattened curve. The overall uncertainty is so great that it falls off the chart on the right-hand side. So, this third instrument results coarsely unprecise and inaccurate.

Multimeter 4

The plot of the fourth instrument (**Figure 4**) appears to be the best of the series. The error distribution is concentrated on the mean of $\mu = 2.006$, the resulting $\delta = 0.006$ A (+0.3% error on reference value) can be considered as negligible and the value of the standard deviation $\sigma = \pm 0.04$ A ($\pm 2\%$ on average μ) is low. The conclusion that can be drawn is that this is both a precise and accurate instrument, and because of its features, it can also be defined as *exact*.

A More Handy Representation

For a more practical depiction, the summary table in **Figure 5** may be useful. In it, the above examples' measurement distributions are represented on four targets. This is a widely used example in these cases, because of the very immediacy of its interpretation. In addition, the green and blue frames indicate the systematic and random error components' areas, partially overlapping, in the worst case — multimeter 3, which is affected by high levels of both.

Ideally, the Cartesian axes could be superimposed on this drawing, where the Y axis represents precision, the X axis accuracy, and the straight line function $Y = X$ in the first quadrant is the *exactness* of measurement, i.e., the combination of both.

Reference Value or True Value?

At the beginning of this article, we wrote that we considered the *reference value* to be the *true value*. This is because we relied on the

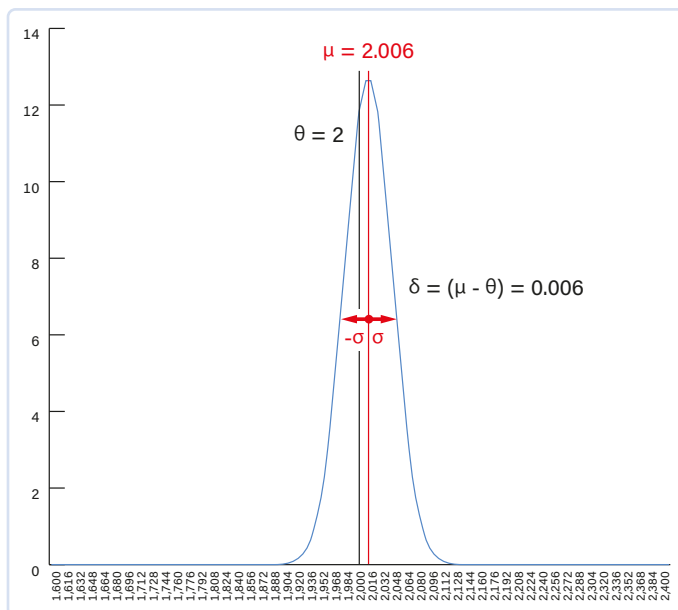


Figure 4: Finally, a trustworthy multimeter! With an accuracy error of +0.3% and $\pm 2\%$ on precision, it will still be useful on the workbench.

characteristics of our reference current generating instrument, considering it to be exact for our standards.

In reality — I'm sorry to disappoint you — the *true value* of any quantity exists, but it is **unknown** and **inaccessible**, given that every measurement is subject to an uncertainty level of greater than 0.

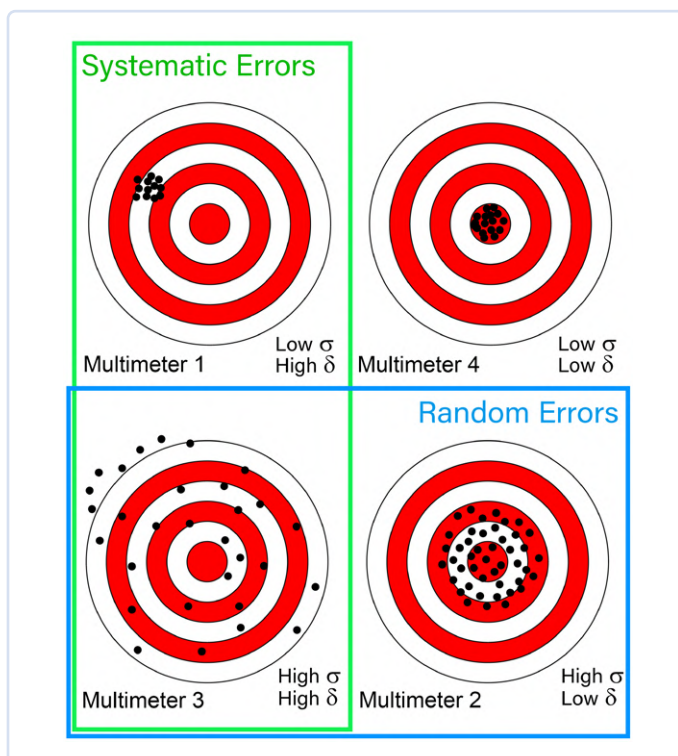


Figure 5: A much more practical, but very effective representation. In the four targets, the diameter of the perforated areas represents the dispersion (σ , the reciprocal of precision), while their distance from the center (δ) indicates inaccuracy. Multimeter 4 is not error-free, but the two error components, systematic and random, are far lower.



In other words:

- The *true value* is perfect, but only **theoretical**: impossible to determine exactly.
- The *reference value* is **the best value available**, and is used as a (local) standard in practical measurements.

The task of metrology is specifically to bridge, as far as possible, the distance between the *true value* and the *measured value*.

This distance is narrowed by using different standards, of increasingly higher level of accuracy, which constitute the so-called *traceability chain* of a measuring instrument [5]. This, and many other detailed aspects of acquisition and reading errors inherent in electronic measuring instruments, will be the subject of a future article. Stay tuned! ◀

250046-01

Questions or Comments?

Do you have technical questions or comments about this article? You can write to the author at roberto.armani@elektor.com, or contact the editorial team of Elektor at editor@elektor.com.

About the Author

Roberto Armani is an electronic engineer. After his university studies at Politecnico di Milano, he gained over 35 years of experience in various sectors. Before joining the Elektor team as a Senior Editor, he worked 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.



Related Product

- **PeakTech 3440 True RMS Graphical Multimeter**
www.elektor.com/18157



WEB LINKS

- [1] About Standard Deviation, mathsisfun.com: <https://mathsisfun.com/data/standard-deviation.html>
- [2] About Standard Deviation, cuemath.com: <https://cuemath.com/data/standard-deviation>
- [3] Short notes on Laplace's Central Limit Theory, Medium.com: <https://tinyurl.com/56wbem3d>
- [4] Edel Alexandre Silva Pontes, "A Brief Historical Overview Of the Gaussian Curve," IJESI: <https://tinyurl.com/murz3nuh>
- [5] NIST Traceability chain: <https://nist.gov/image/traceability-chainjpg>

Join our Community



www.elektormagazine.com/community



AD7124

A Precision ADC in Practice

Features for Sensor Signal Conditioning

By Diego de Azcuénaga (Argentina)

Precision, high-performance A/D converters, unlike the most common ones, require some extra technical considerations for interfacing and signal conditioning of different types of sensors. This is to ensure the best utilization of the full potential they offer. Let's see how!

This article aims to propose practical applications where it can be seen how the special features of the AD7124 ADC by Analog Devices allow the signals coming from different types of sensors to be conditioned to obtain a precise and efficient measurement. Through the following case-of-use examples, the operation of these characteristics is explained, and how they help to achieve an efficient measurement system.

Case 1: pH Sensor

The output of the pH sensor is bipolar and gives a maximum signal of ± 414 mV at 25°C. The AD7124 operates from a single power supply ($AVDD = 3.3$ V); therefore, the pH probe should be biased above ground so that it is within the acceptable common-mode range of the AD7124.

Biasing a signal above ground means to apply a constant DC voltage to a signal, setting its resting level above the ground potential (usually considered 0 V), essentially giving the signal a positive offset, so that it always operates at a voltage higher than ground level.

One of the integrated features of the AD7124 is its internal bias voltage generator that sets the common-mode voltage of a channel to $AVDD/2$, or 1.65 V. This bias voltage from the ADC is applied to the pH probe shield (see **Figure 1**) and sets the output of the sensor $1.65\text{ V} \pm 414\text{ mV}$ at 25°C.

The circuit uses the ADA4661-2 precision op amp to buffer the high impedance pH probe output and to drive the ADC. Typical input bias current is 0.15 pA to minimize offset errors caused by the bias current flowing through the high output impedance — approximately $1\text{ G}\Omega$ — of the pH sensor.

An op amp voltage buffer mirrors a voltage from a high-impedance input to a low-impedance output. It's an amplifier with a gain of 1. Output is connected back to inverting input (-).

Case 2: 4-Wire RTD

A Pt100 RTD (Resistance Temperature Detector) is a temperature sensor that measures temperature based on changes in the resistance of a platinum conductor. One of the on-chip excitation currents of the AD7124 (IOUT1, see **Figure 2**) excites the 4-wire Pt100 RTD with DC current. The maximum resistance of the Pt100 is $313.71\ \Omega$ (at 600°C). If the excitation current is programmed to 250 μA , the maximum voltage generated across the Pt100 is $250\ \mu\text{A} \times 313.71\ \Omega = 78.43\text{ mV}$.

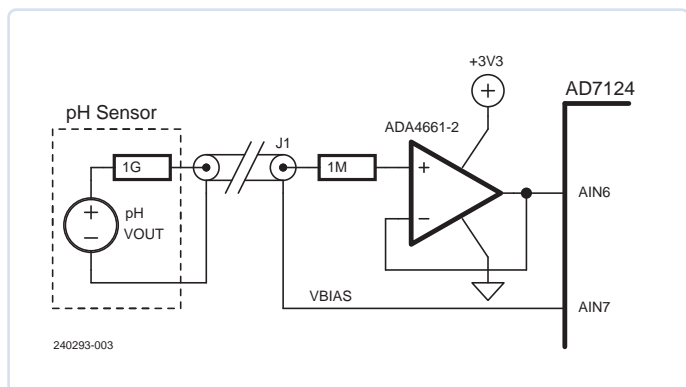


Figure 1: pH Measurement Circuit.

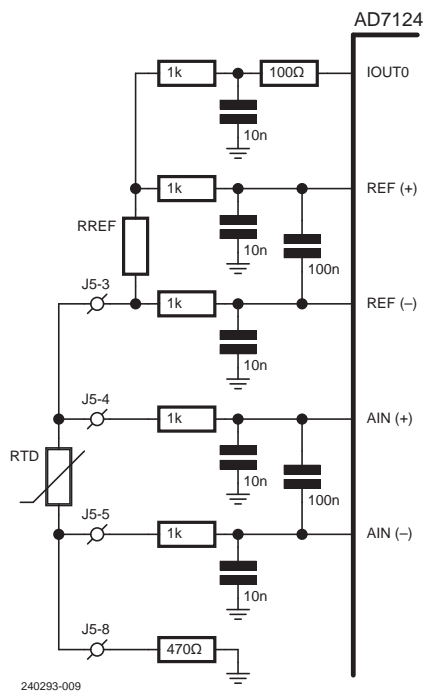


Figure 2: 4-wire RTD Measurement Circuit.

With the programmable gain amplifier (PGA) programmed to a gain of 16, the maximum voltage seen at the ADC modulator input is $78.43 \text{ mV} \times 16 = 1.25 \text{ V}$.

A $5.6 \text{ k}\Omega$ (R_{REF}) precision resistor generates the reference voltage. The value of the reference resistor, R_{REF} is chosen so that the voltage generated across it has a minimum value of 1.25 V in this circuit. The voltage “seen” at the ADC modulator input, as a result of that voltage present at the analog input, is then less than or equal to the reference voltage.

The reference voltage (V_{REF}) of an analog-to-digital converter (ADC) is the maximum voltage that the ADC can convert. It’s used to set the range of voltages that the ADC maps to binary values. The digital outputs are a ratio of the input reference voltage.

The $250 \mu\text{A}$ current that supplies the Pt100 also flows through the $5.6\text{-k}\Omega$ resistor, resulting in a ratiometric configuration. Therefore, any variation of the excitation current has no effect on the performance of the circuit. The voltage generated across the precision resistor is $5.6 \text{ k}\Omega \times 250 \mu\text{A} = 1.4 \text{ V}$.

Antialiasing filters are included on the analog input and reference input. These filters prevent any interference at multiples of the modulator sampling frequency aliasing into the band of interest.

Differential antialiasing filters are formed by parts between $AIN(+)$ and $AIN(-)$ and between $REF(+)$ and $REF(-)$. It is an RC lowpass made by $1 \text{ k}\Omega$, $1 \text{ k}\Omega$, $0.1 \mu\text{F}$.

Common-mode antialiasing filters are formed by parts between $AIN(+)$ and ground, $AIN(-)$ and ground and between $REF(+)$ and ground, $REF(-)$ and ground. It is an RC lowpass made by $1 \text{ k}\Omega$, 10 nF .

Programmable Gain Amplifier

A programmable gain amplifier (PGA) is an operational amplifier (op amp), whose gain can be changed by external control signals. The gain of this non-inverting amplifier is set by the ratio $(R_F/R_X) + 1$. To “program” it, we can change the value of R_X by switching some resistors in and out through some analog switches.

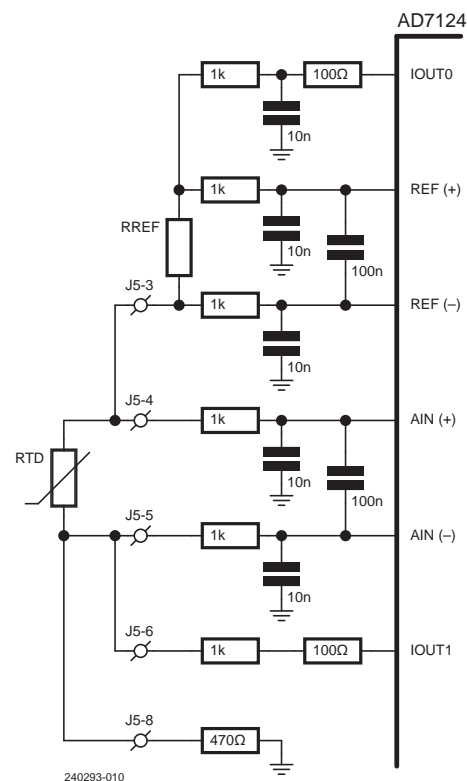
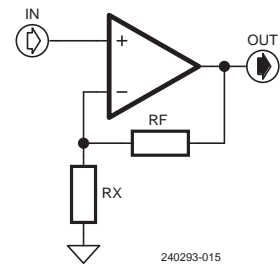


Figure 3: 3-wire RTD Measurement Circuit.

To ensure that these filters and the external resistors do not cause gain errors, the reference buffers are enabled. The analog input buffers are enabled automatically when the PGA is used. The reference buffers require some headroom (100 mV within the power supply rails). Therefore, a $470\text{-}\Omega$ resistor was included on the low side to allow this headroom.

The $250 \mu\text{A}$ excitation current requires 370 mV of headroom. The voltage seen on this pin is:

$$(470 + 5600 + 313.71) \Omega \times 250 \mu\text{A} = 1.6 \text{ V}$$

which is acceptable.

Case 3: 3-wire RTD

The AD7124 has two matched excitation currents (I_{OUT0} and I_{OUT1} , see **Figure 3**). Both are used with the 3-wire RTD so that lead wire compensation can be performed. Because the Pt100 has a

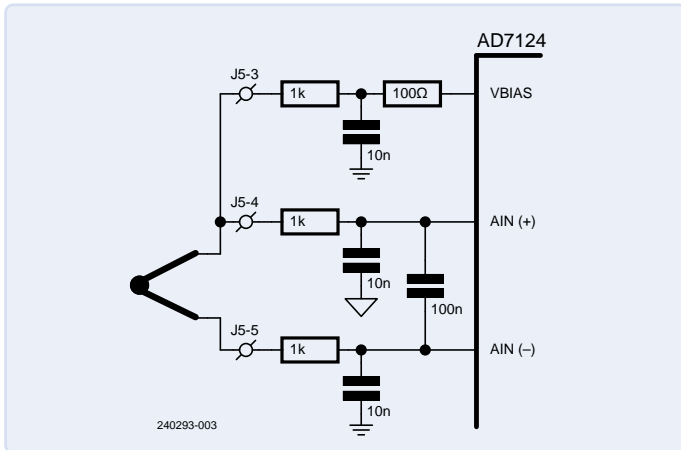


Figure 4: T-Type Thermocouple Measurement Circuit.

maximum resistance of 313.71 Ω (at 600 $^{\circ}\text{C}$), the excitation currents are programmed to 250 μA .

The reference resistor, R_{REF} is connected on the top side. Therefore, one 250- μA excitation current (IOUT0) flows through the 5.6-k Ω (R_{REF}) precision reference resistor and then through the Pt100. The second 250- μA current (IOUT1) generates a voltage due to the lead resistance, which is in opposition to the lead resistance voltage generated due to the current on IOUT0. Therefore, the error due to the lead resistance is minimized.

The voltage generated across the 5.6 k Ω reference resistor is $250 \mu\text{A} \times 5.6 \text{ k}\Omega = 1.4 \text{ V}$. The maximum voltage generated across the Pt100 is $250 \mu\text{A} \times 313.71 \Omega = 78.43 \text{ mV}$.

With PGA programmed to a gain of 16, the voltage at the sigma-delta modulator of the AD7124 is $78.43 \text{ mV} \times 16 = 1.25 \text{ V}$. This voltage is within range because it is less than the reference voltage.

The 250- μA excitation current requires 370 mV of headroom. The voltage seen on this REF(+) pin is $(470 \Omega \times 2 \times 250 \mu\text{A}) + (5600 + 313.71) \Omega \times 250 \mu\text{A} = 1.713 \text{ V}$, which is acceptable.

Case 4: Thermocouple

The thermocouple is connected between the AIN+ and AIN- terminals (see **Figure 4**). The AD7124 contains a VBIAS source that biases the thermocouple to $(\text{AVDD} - \text{AVSS})/2$, where $\text{AVDD} = 3.3 \text{ V}$ and $\text{AVSS} = 0 \text{ V}$.

The internal 2.5 V reference is used for the conversion. A T-type thermocouple can measure temperatures from -200°C to $+400^{\circ}\text{C}$, generating a voltage range of -8 mV to $+16 \text{ mV}$. The PGA is set to 128. A thermocouple's cold junction is the connection point where the thermocouple attaches to AD7124 (J5-4 and J5-5). It's also known as the reference junction. For the cold junction compensation, a 10 k Ω NTC thermistor is used, see **Figure 5**. Thermocouple cold junction compensation (CJC) is a process that adjusts the output

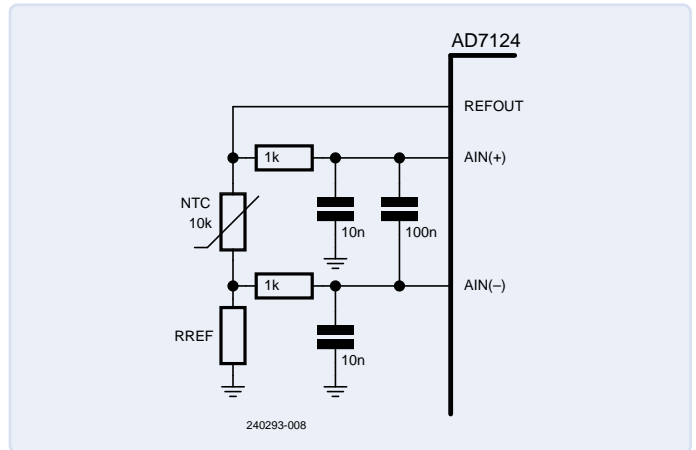


Figure 5: Thermistor for the cold junction compensation.

voltage of a thermocouple to account for the temperature of the cold junction.

The internal reference supplies the thermistor through REFOUT, the gain being set to 1 for this conversion. A 5.6 k Ω resistor, R_{REF} is used on the low side of the thermistor. It provides 100 mV of headroom for the analog input because the input buffers are enabled.

Case 5: Pressure Sensor

For the voltage-excited pressure sensor, AVDD (3.3 V) excites it (see **Figure 6**). The excitation voltage can be used directly as the

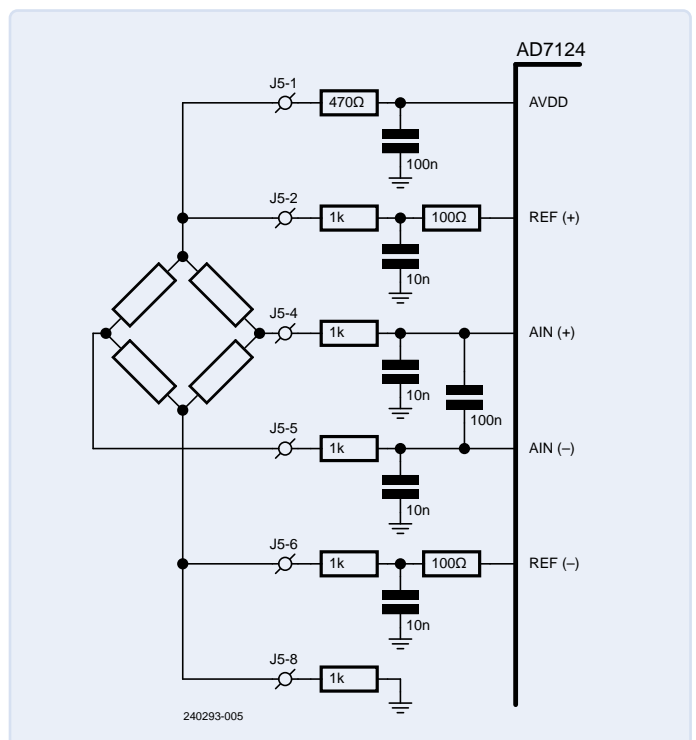


Figure 6: Voltage-Excited Pressure Sensor.

ADC reference. Note that a resistor was added on the high side and the low side of the sensor to ensure that the headroom requirements of the AD7124 buffers are met. A gain of 16 is used.

For the current-excited pressure sensor, both of the 100 μA excitation currents of the AD7124 are directed to IOUT0 (see **Figure 7**), resulting in a 200 μA excitation current for the sensor. The gain is set to 16. A 5.6 k Ω precision resistor generates the reference. This arrangement results in a ratiometric configuration, that is, any deviation in the excitation current value has no effect on the system accuracy.

Case 6: Thermistor

Exciting the thermistor with even the smallest possible excitation current available on the AD7124 is not appropriate because the thermistor tends to generate a high output voltage at lower temperatures. For example, an excitation current of 50 μA generates a voltage of $441.117 \text{ k}\Omega \times 50 \mu\text{A} = 22 \text{ V}$ when measuring a temperature of -50°C . This voltage level is outside the specified range of the AD7124.

A simple approach of measuring R_{TH} is to connect the thermistor in series with a 0.01% sense resistor (R_{SENSE}) as part of a potential divider circuit (see **Figure 8**). A constant 2.5 V excitation voltage is provided from REFOUT to generate a voltage across R_{TH} , which is then converted to a precision digital representation using the 24 bit, $\Sigma\text{-}\Delta$ ADC.

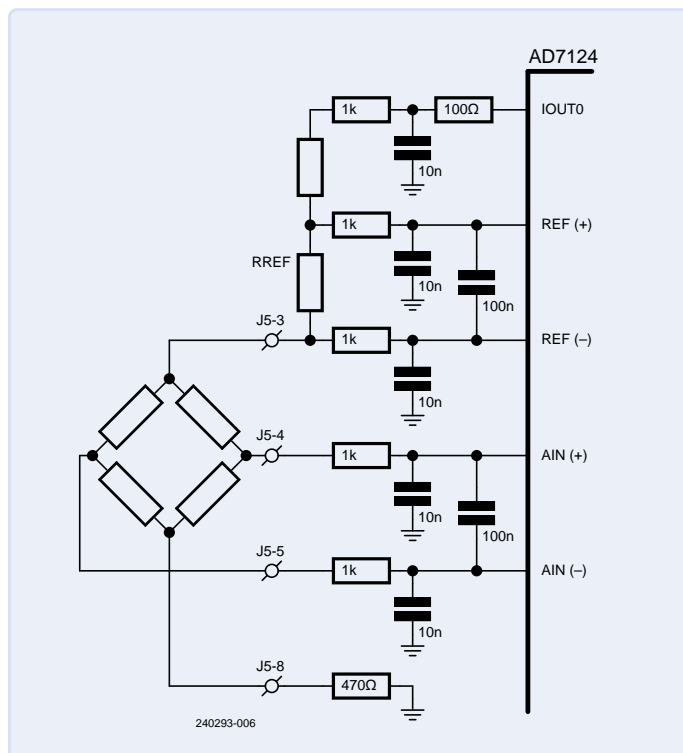


Figure 7: Current-Excited Pressure Sensor.

By using the selected reference voltage to supply the thermistor and/or R_{SENSE} and by using this same reference as the ADC reference for the measurement, errors in the excitation voltage source can be cancelled, thereby resulting in a ratiometric measurement result.

The maximum voltage generated across the 10 k Ω NTC thermistor is approximately 2.44 V. Therefore, the PGA gain is programmed to a gain of 1 to ensure that the entire sensor output range is within the allowable input range of the AD7124.

The AD7124 incorporates a high impedance analog input stage buffer amplifier. Therefore, the input can tolerate significant source impedances and is tailored for direct connection to external resistive type sensors such as thermistors. The inputs are automatically buffered when the PGA gain is greater than 1. However, the buffer can be disabled when the gain is equal to 1 as long as the source impedance is low.

In the circuit, the analog input is driven by the thermistor with a high resistance value (up to 441.11 k Ω), which can cause significant gain and offset errors. Enabling the input buffers allows a wide range of resistor values and electromagnetic capability (EMC) filtering without adding any error. The buffers require at least 0.1 V headroom above and below the supply rails.

This method also ensures that the voltage across the thermistor is always lower than the reference voltage, regardless of its larger resistance value at lower temperatures, which greatly simplifies the direct input connection for measuring the entire temperature range.

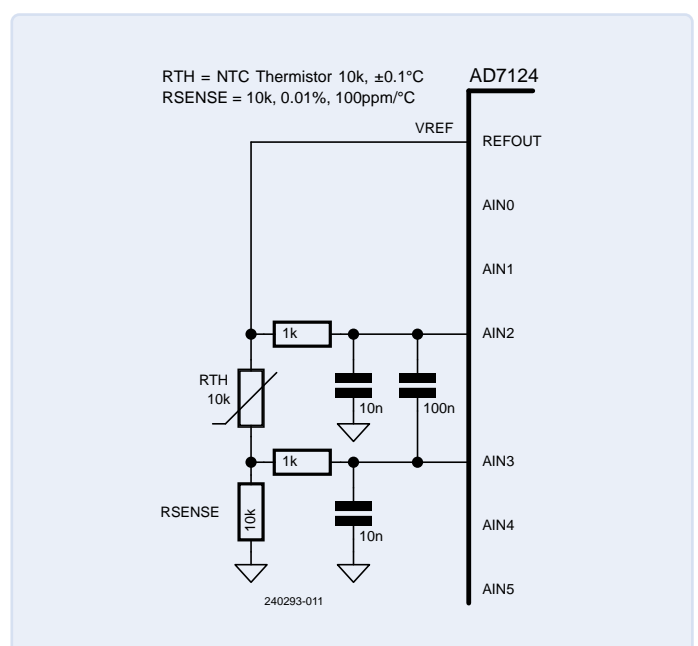


Figure 8: A 10 k Ω NTC Thermistor Analog Input Configuration.

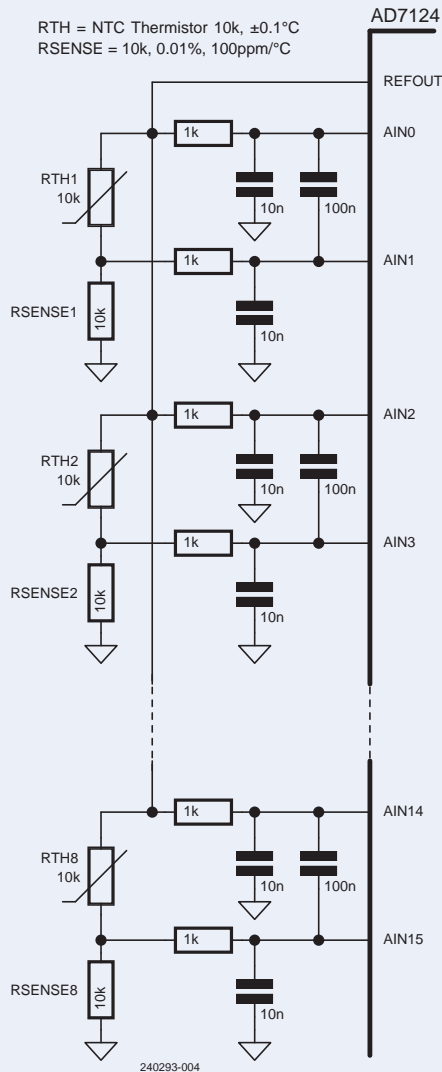



Figure 9: A 10-k Ω NTC Thermistor Multiple Analog Input Configuration.

When multiple thermistors are used, each one must have its own precision R_{SENSE} , but they can share a single excitation voltage (internal VREF, see **Figure 9**). As a minimum, each NTC thermistor requires two analog input pins of the AD7124.

Let's Start Sampling!

As could be seen, different types of sensors were configured with the AD7124 ADC, taking advantage of its special characteristics to achieve a correct measurement of its parameters.

In this way we have described the operation of bias voltage generator, excitation currents, internal voltage reference, programmable gain amplifier (PGA), reference and analog input buffers applied to each sensor in particular. 

240293-01

About the Author

Diego de Azcuénaga is a technical writer specialized in several topics of electronics and technology and an electronics engineer. He has worked on various projects designing data acquisition and control systems. Diego enjoys writing and telling stories about Power Electronics, IoT, Embedded and Reference Designs.

Questions or Comments?

Do you have technical questions or comments about this article? You can contact the author at azcuenagadiego@yahoo.com.ar or the editorial team of Elektor at editor@elektor.com.



Related Product

> **Elektor Ultimate Sensor Kit**
www.elektor.com/19104



WEB LINKS

- [1] CN-0398 Circuit Note: Soil Moisture and pH Measurement System with Temperature Compensation, Analog Devices Inc.: <https://tinyurl.com/yeznamms>
- [2] CN-0382 Circuit Note: Isolated 4 mA to 20 mA/HART Temperature and Pressure Industrial Transmitter Using a Low Power, Precision, 24-Bit, Sigma-Delta ADC, Analog Devices Inc.: <https://tinyurl.com/bdcnp6em>
- [3] CN-0545 Circuit Note: Completely Integrated, 0.1°C Thermistor Measurement System Using a Low Power, Precision, 24-Bit Sigma-Delta ADC, Analog Devices Inc.: <https://tinyurl.com/4ev4d23e>

ADVANTECH

AIR-150

EDGE AI RETHOUGHT

COMPACT, STRONG, SUITABLE FOR INDUSTRIAL USE

Compact and powerful

- Edge AI system in mini format
- Intel® Core™ (13th Gen.) + Hailo-8
- Ideal for space-saving AI applications

Superior AI inference

- 26 TOPS of computing power
- Faster than Orin Nano & Xavier NX
- Energy efficient & powerful



26 TOPS

HAILO

intel CORE

Flexible application options

- For robotics, security & inspection
- COM, USB, LAN, M.2 E-Key connections
- Flexible integration into existing systems

Robust for industrial use

- Robust housing & -20 to +60 °C
- EMC-certified (IEC 61000-6-4/-6-2)
- Suitable for harsh industrial environments

FORTEC INTEGRATED

www.fortec-integrated.de/en

Interested?



PID Control Tool

Optimize Your Parameters Easily

By Johannes Sturz (Germany)

PID control has been in use for a long time in industry, but only for the past few years has it been introduced into environments such as Arduino, with rather concise libraries and no graphical aids. This project fills these gaps by introducing a very useful GUI to play with.

There are many microcontroller projects using PID control, but most of them are not particularly suitable for experimentation. Firstly, the code has to be recompiled when a parameter is changed. Furthermore, there is no visualization of the closed-loop controller behavior, or it is inadequate — for example, there is no zoom-in function. In addition, individual adjustments of the GUI should be easy. For this reason, an application was programmed with HTML/JavaScript that runs in the Internet browser. If PID control is not entirely familiar to you, take a look at the “How Does a PID Controller Work?” text frame in this article.

Nothing better than to show this GUI tool with an example project: we want to reach a target position of a motor as smoothly and fast as possible; a motor encoder tells us repeatedly the current position. For those interested in learning more about the operating principle of a closed-loop system — with motor and encoder — this website [1] might prove to be a useful starting point. There is also an Elektor book about the subject [2].

How Does a PID Controller Work?

A PID controller is a feedback controller used in many control systems to maintain a desired output by adjusting inputs based on the error between the desired target position and the actual position. PID stands for Proportional, Integral, and Derivative, which are the three terms that make up the control algorithm. They should be optimized by following three factors: steady-state error, transient response and stability.

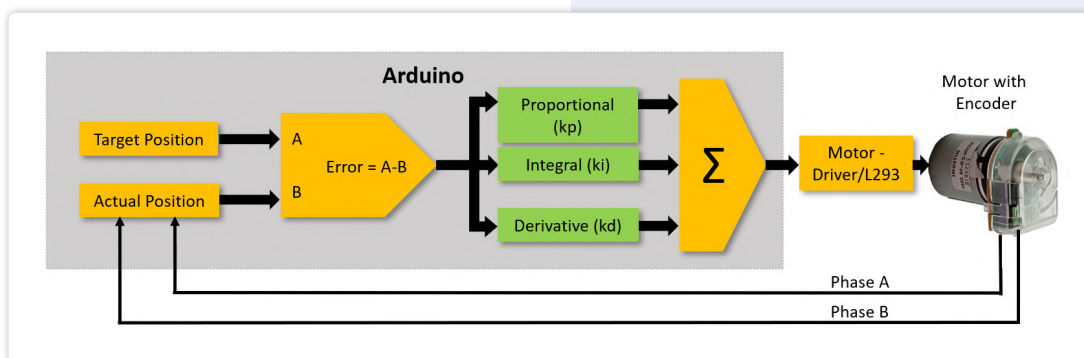
Why are three branches (P, I and D) necessary here?

The proportional branch increases the error by the factor k_p . Now you might think that the higher k_p , the more precise the control, but that has its limits. A high k_p value can lead to oscillation since, in our case, the motor cannot brake quickly enough and overshoots the target. With a moderate k_p value, the target value is not reached (called steady-state error) because the error becomes small when the target position is almost reached.

The integral branch solves the previous problem by adding up all the errors. In our case, the cycle time is 1 ms. This means that even the smallest errors are eventually compensated for. Caution is advised here; if a system reaches a limit, the integral value can become infinitely large.

The control loop can be further optimized using the derivative path. Here, the change in the error over time is considered. You can virtually *see into the future*. This path reacts strongly if the controlled variable suddenly deviates from the target position. It tends to reduce the overshoot and the settling time.

Further information can be found in this book [2].




```

|*****|
|**      PID-Tuning (1.0)      **|
|**          Menu          **|
|*****|

Select one of the following options:
[h]      - Menu
[t <value>] - Set target position
[p <value>] - Set proportional part
[d <value>] - Set derivative part
[i <value>] - Set integral part
[s <value>] - Set speed [%]
[g]      - Get parameter (t,p,d,i, psw)
[l]      - Load configuration to EEPROM
[v]      - Save configuration to EEPROM

```

Figure 1: List of available commands.

The demo project consists of three parts (all files can be downloaded at [3]):

- > **Arduino Sketch:** The program code calculates the current motor position from encoder signals, processes the PID control (with a PID Arduino library), and controls the motor with the help of an external motor driver. Additionally, the motor position is sent out repeatedly via a serial interface; in the other direction commands are received to configure the PID parameters.
- > **Tuning GUI:** The GUI is written in HTML, CSS and JavaScript. The GUI runs with any internet browser that has native serial interface support (Chrome, Edge and Opera, but not Firefox). The code can be easily modified with any editor.
- > **Hardware:** Arduino Nano; H-bridge motor-driver based on device L293D; and DC Motor with encoder.

Arduino Sketch

There is a simple command interface implemented to set the parameters. It must be noted here that a space between the parameter and the value must be inserted. **Figure 1** reports a list of all the available commands. You can set the parameters or use other commands, manually via a terminal program. However, it is more comfortable to use the graphical tool (which uses the same serial command interface).

For PID control, a good Arduino library from Brett Beauregard is used. The *PID_v2.h* can be installed from the Arduino IDE (*Tools* → *Manage Libraries*) or downloaded at [4]. A detailed description of the lib is available at [5]. It is not very complicated to code a PID controller with this library, but there are some additional conditions that still need to be considered. For example, if the desired position cannot be reached because the carriage is at the stop, then the integral component would constantly increase, which means that the integral part is much too high for subsequent positioning.

Another lib, *EEPROM.h*, allows easy access to EEPROM. All parameters will be stored into a struct named *Configuration*. This makes it easier to save them all on the EEPROM. The maximum speed is only set via PWM and not regulated. The password is necessary so

that the sketch recognizes that a configuration has already been saved after a reset or reprogramming of the Arduino. Otherwise, default values are loaded.

The configuration is not overwritten by a reset or reprogramming of the Arduino. If a new target position is set, then the sketch sends the encoder position to the serial port until the target position is settled, or 2000 samples are received.

Tuning GUI

The GUI is implemented as a web application, running in your browser. It has four parts:

- > *index.html*
- > *pid.js*
- > *serial.js*
- > *style.css*

The *index.html* file is responsible for the structure. All controls, libraries, and links are also integrated here. The *style.css* is used to lay out the web page, such as the color, font, size of text, and so on. The basic serial communication functions are located in *Serial.js* and the functions that communicate with the Arduino are in *pid.js*.

The serial communication is fully interrupt controlled, which means that no data will be lost during receiving data while still being sent data. Further information can be found in [6] and [7]. To start the GUI, all you need to do is double-click on the *index.html* file. Next, the correct port must be selected. Then all controls are visible, as shown in the screenshot of **Figure 2**.

It is important that all of the above files are in the same folder, and the referenced browser must be Microsoft Edge, Google Chrome, or Opera browser. Visual Studio Code was used as an editor because it offers interesting features to create HTML pages and write JavaScript code. As an example: Intellisense function (code completion), auto structure and a local web browser for test and development purposes. You can download Visual Studio Code for free at [8]. However, any other editor can also be used. Debugging of HTML/JavaScript code can be done with the browser itself. Activating this depends on the browser used (Microsoft Edge: *Ctrl+Shift+I* and Google Chrome: *Ctrl+Shift+J*). Further details can be found at [9] or [10]. The Opera browser is also supposed to support the serial interface, but it has not been tested.

After changing values in data fields or sliders, the respective *Send* button must be pressed. As long as the data has not been sent, the button is red. The parameter setting can also be saved to the EEPROM of the Arduino controller with a button press of *Save Conf*. A stored configuration can be retrieved by pressing the *Load Conf* button.

Logged data can be exported in .csv format. The console window displays all serial communication data. The graph allows zooming

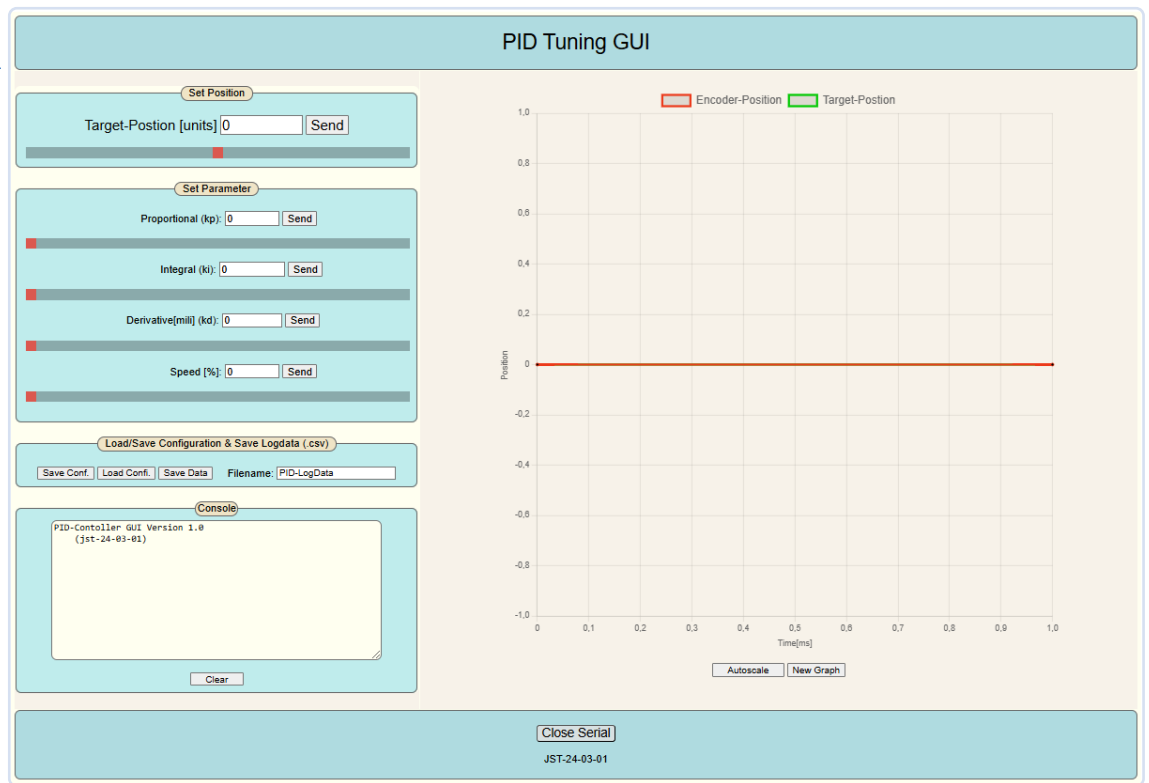


Figure 2: After a COM port has been selected, all controls become visible.

and panning with the mouse. Autoscale resets the graph settings. The toggle button Add Graph/New Graph (see **Figure 3**) appends the new data to the old one or clears the data, which were stored before. The time interval between the data points is 1 ms.

Limits of the target position and parameters can be changed in the `index.html` file. Since the `index.html` of the web application contains

various references to external libraries, an online connection is necessary:

- Chart.js (JavaScript open-source charting library)
- Serial.js (simplifies the use of web serial)

If the libraries are installed locally, it also works offline.

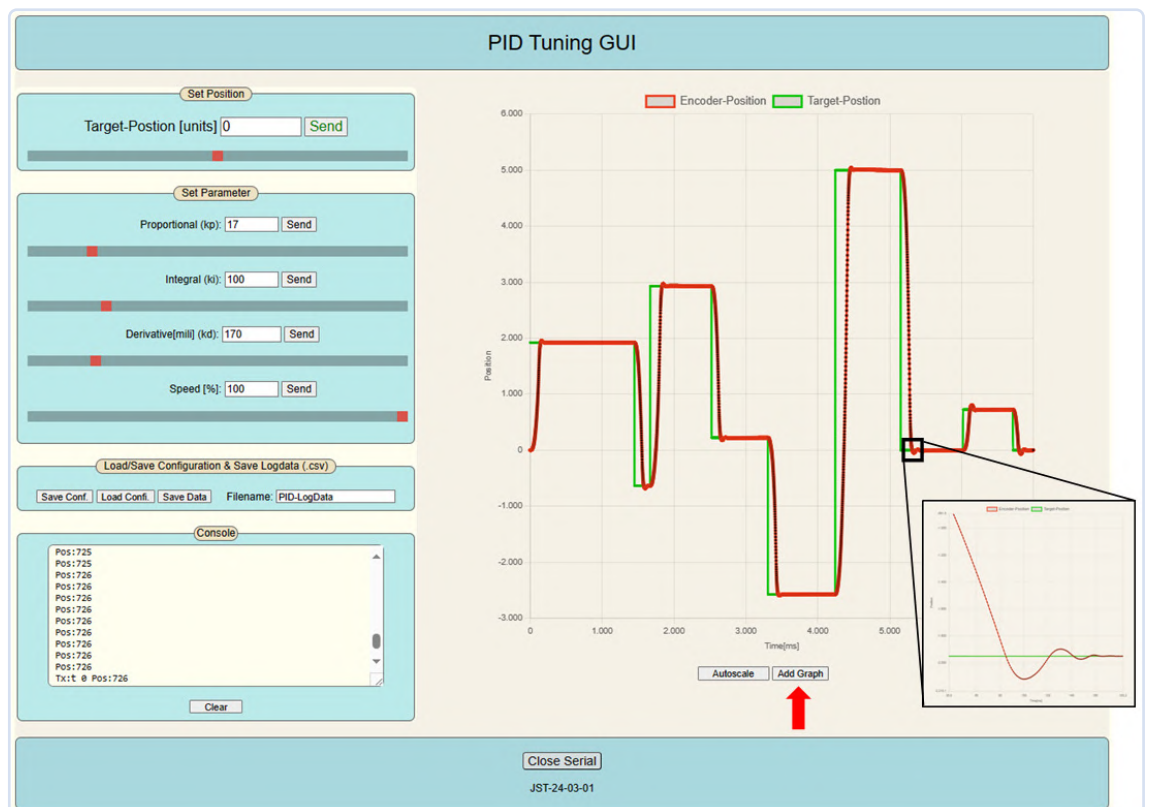


Figure 3: Here "Add Graph" is selected, which means that every new position is appended to the old one. Zoom-in via mouse-wheel.

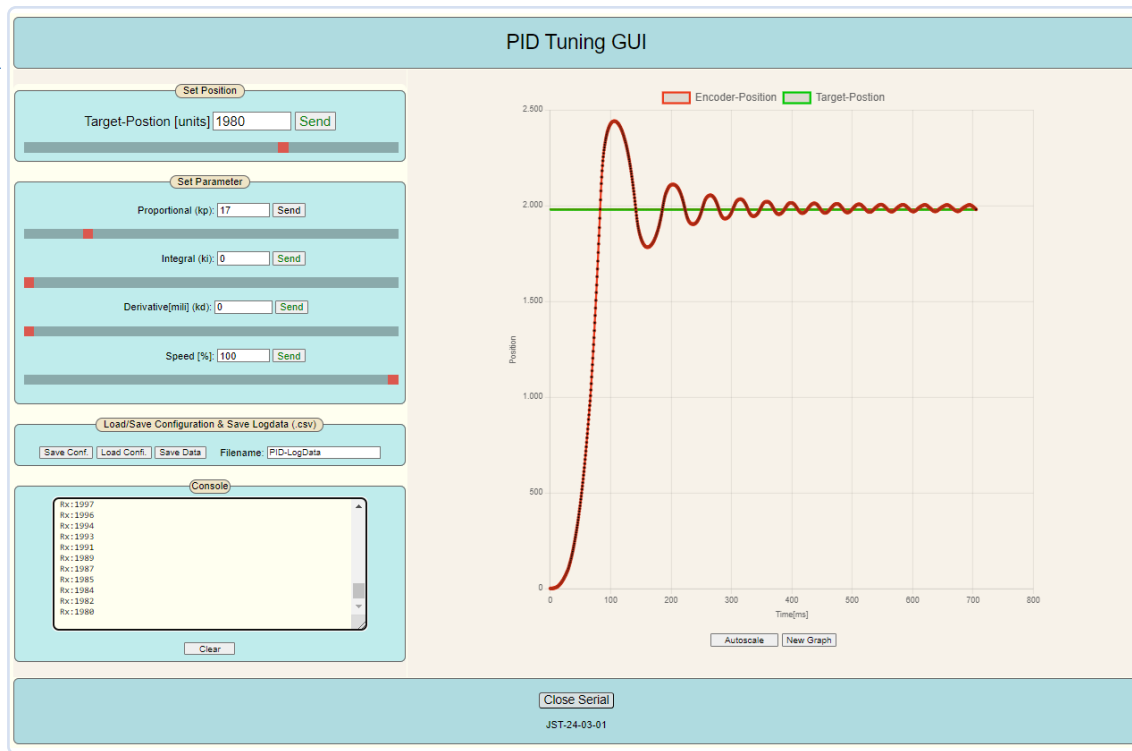


Figure 4:
The proportional component is set too high here. The oscillation never ends.

How to Start Parameter Setting

Start with $k_p = 3$, $k_i = 0$ and $k_d = 0$. Then turn up k_p until you get the pitch you want. An excessive setup value in the proportional control could lead to uncontrollable oscillations, see **Figure 4**. After this, turn up k_i until you get the set point with no (or nearly no) overshoot. Now, increase k_d and stop before the control loop starts to oscillate. An accurately tuned system should produce a graph similar to the one visible in **Figure 5**.

Circuit Diagram

Figure 6 shows the Arduino Nano with all connections. JP1 is for the motor supply voltage, and JP2 is the connector for the encoder. The voltage depends on the motor and is limited to 36 V because of the H-bridge. The current has to be limited to about 500 mA. Otherwise, the H-bridge (L293D) could be damaged. If the motor does not stop at positioning, then the polarity of the motor connections should be reversed.

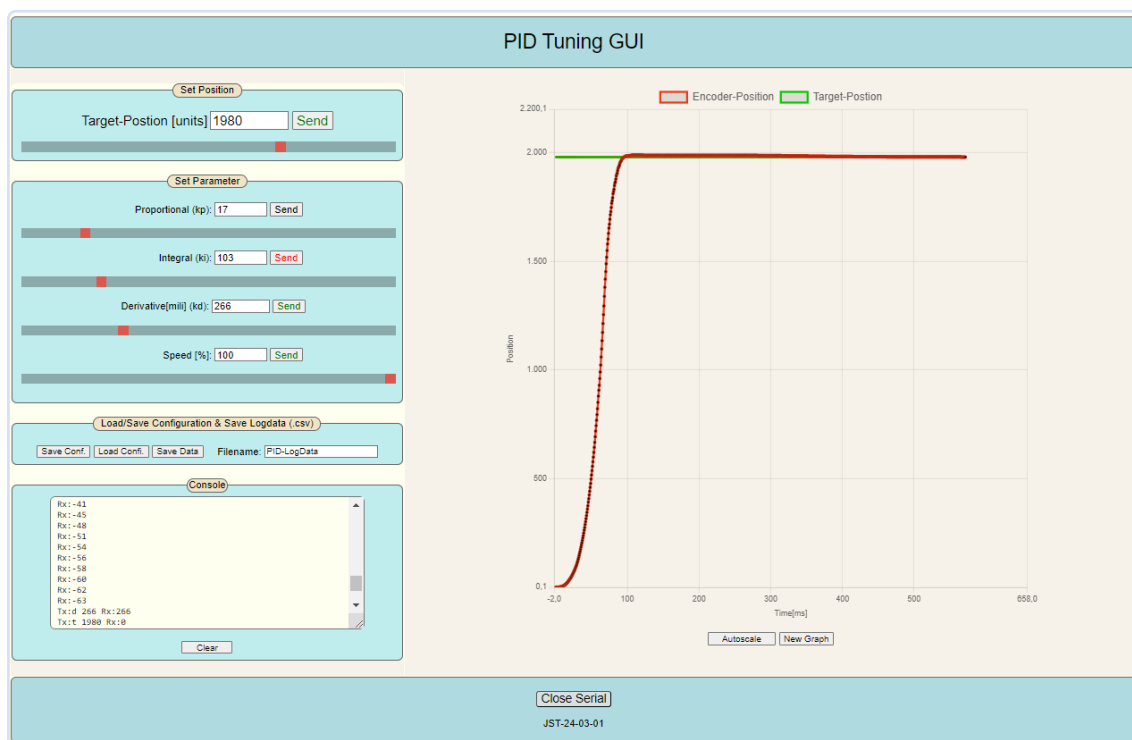


Figure 5:
Nearly perfect result (with linear system).

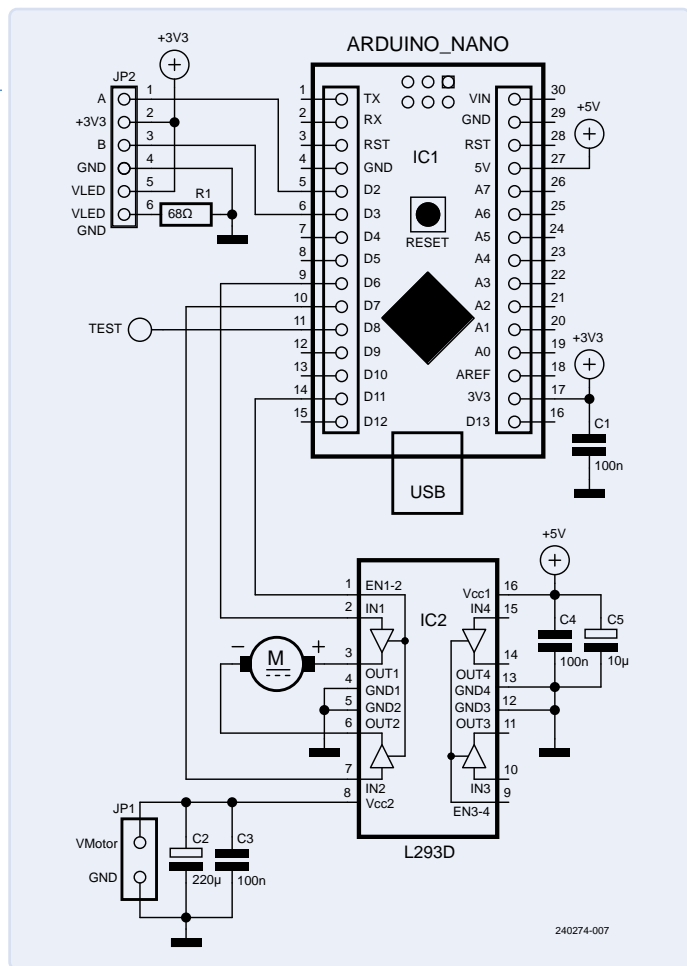


Figure 6: The H-bridge has two supply voltages, V_{CC1} (Pin 16) for the logic and V_{CC2} (Pin 8) for the motor power.

Encoder Pinning

If you use an encoder from an old ink-jet printer, is it usually easy to get the pinout on the PCB. Ground is usually the widest connection, or it has a thermal via, followed by the power connection,

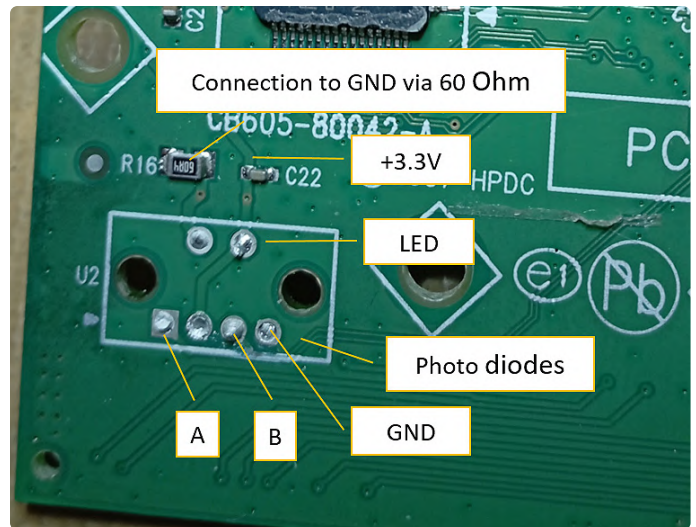


Figure 7: Backside of a printer PCB (sliding carriage).

and the narrowest are the data lines. See **Figure 7**. The other way is to search for the part at any supplier or a simple trial and error method.

Typical Encoder from Agilent/Avago

Figure 8 illustrates the structure of a typical optical encoder. On the left side is an infrared LED and the other side has two infrared receivers. The tape (in the case of the sliding carriage, see **Figure 9**) and the code wheel mounted at the back of the DC motor (**Figure 10**) have tiny stripes on them, through which light cannot pass. When this stripe or wheel blocks the IR light from the emitter, the receivers don't see it.

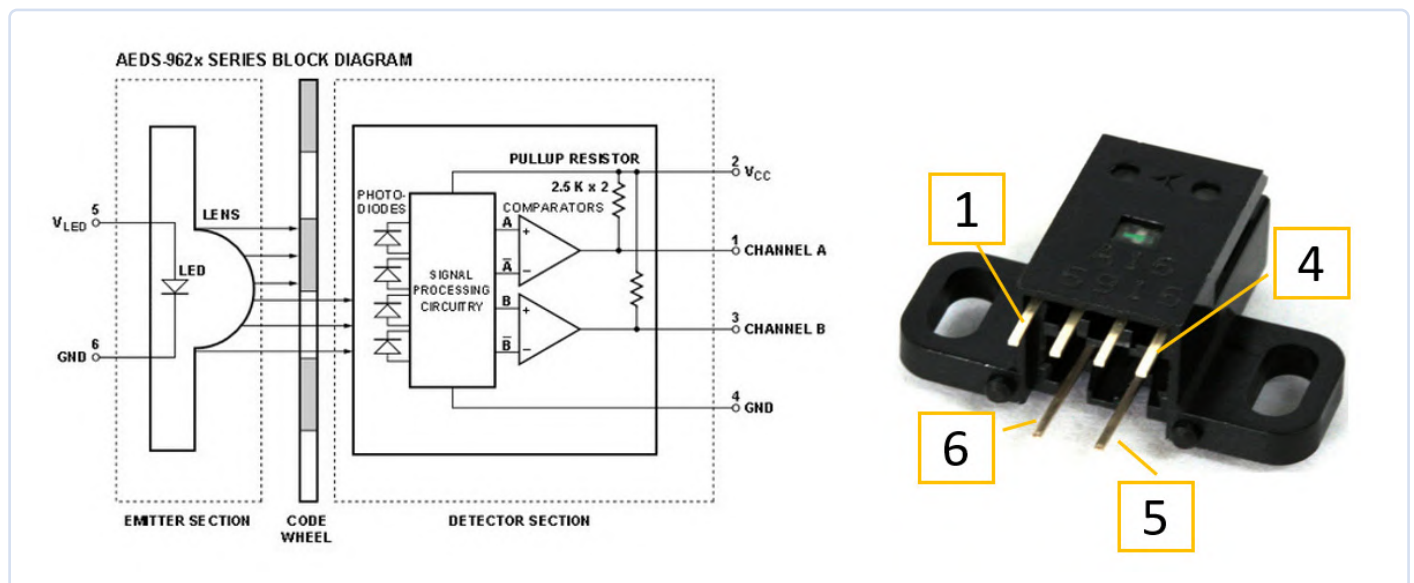


Figure 8: Example of a typical optical encoder (AEDS-962x) and the pinning on the right side. (Source: Agilent AEDS-962x Optical Encoder Module, www.farnell.com/datasheets/20481.pdf)

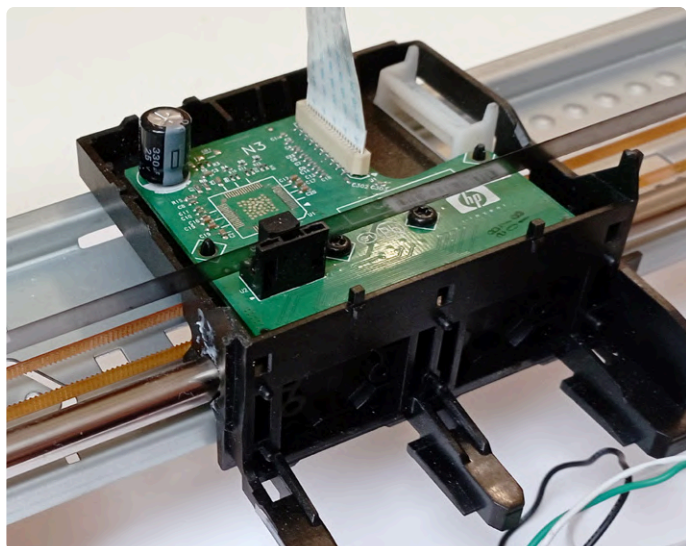


Figure 9: Print head assembly with belt, optical tape and optical sensor. The controller on the PCB was removed to avoid potential faults.



Figure 10: DC-motor with encoder wheel.



Now Give It a Try!

Experience shows that it is not so easy to find the optimal parameters for the different applications. The overshoots depend heavily on the system itself (damping) and also on the distance covered. One idea here is to change the parameters dependent on the distance. For more information, see [4].

Else than driving a DC motor, like was done in this article, BLDC motors can also be driven. In this case, however, the functions `pwmOut()`, `forward()` and `reverse()` must be adapted for the respective motor driver. A complex lib can be found at [11]. This lib also enables speed control. ◀

240274-01

Questions or Comments?

Do you have technical questions or comments about this article? You may contact the author at johannes.sturz@web.de or the editorial team of Elektor at editor@elektor.com

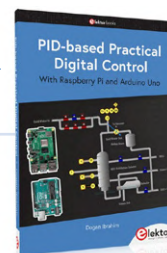
About the Author

Johannes Sturz has been passionate about electronics since he was 12 years old. His interest was aroused by electronic construction kits. He worked for a semiconductor manufacturer as a validation/characterization engineer for several decades. One part of the job was decapped ICs debugging of mixed-signal devices with a probe station, mainly RFID devices and microcontrollers. He retired at the beginning of 2024. This means he has now more time for his own projects and other activities he loves, like hiking and cycling.



Related Products

- Dogan Ibrahim, *PID-based Practical Digital Control with Raspberry Pi and Arduino Uno* (Elektor, 2022) www.elektor.com/20274
- **Arduino Nano** www.elektor.com/17002



WEB LINKS

- [1] Arduino PID DC motor control close loop system on Electric DIY Lab: <https://tinyurl.com/ys7jara5>
- [2] Dogan Ibrahim, PID-Based Practical Digital Control With Raspberry Pi and Arduino Uno (Elektor 2022): <https://tinyurl.com/52rbwy7j>
- [3] Elektor Labs page for this project: <https://www.elektormagazine.com/labs/pid-control-example-using-arduino>
- [4] Arduino-PID-Library-V2: <https://github.com/illinoistechrobotics/Arduino-PID-Library-V2>
- [5] Brett Beauregard, Improving the Beginner's PID: <https://tinyurl.com/mwms4thk>
- [6] Physical Computing, Why Web Serial: <https://tinyurl.com/y4dj5ame>
- [7] Chrome for Developers, Web Serial API: <https://developer.chrome.com/docs/capabilities/serial?hl=en>
- [8] Download link Visual Studio Code: <https://code.visualstudio.com/>
- [9] JavaScript debugging (MS Edge): <https://tinyurl.com/yca7de76>
- [10] JavaScript debugging (Google Chrome): <https://tinyurl.com/rbbnt97r>
- [11] Arduino Simple Field Oriented Control (FOC) project: <https://docs.simplefoc.com/>

RECAP

embedded world 2025

A must for everyone dealing with microcontrollers, embedded world 2025 took place from March 11 to 13 in Nuremberg, Germany. This year, there were 32,000 visitors and more than 1,100 exhibitors. As always, Elektor was present with a team of editors and engineers. We were also joined by colleagues from eeNews Europe. Here, we offer a small sampling of all the innovations seen at the fair. You can find more on our YouTube channel, www.youtube.com/@ElektorTV.



Microchip: Single Pair Ethernet

Microchip showcased an automation demo where smaller and larger microcontrollers communicate seamlessly via Single Pair Ethernet. Single Pair Ethernet is an alternative to CAN bus, requiring only two wires and screw clamps instead of standard Ethernet connectors and cables. It supports a range of up to 50 meters and a data rate of 10 Mbit/s. The standard also allows for powering devices through the same two wires.

<https://tinyurl.com/microchip-SPE>

Analog Devices: CodeFusion Studio IDE

Analog Devices was presenting their new CodeFusion Studio IDE, which is based on VS Code and open source. The integrated "System Planner" helps resource partitioning for microcontrollers, even across different development teams. It allows the configuration of two or more cores, including clocks, pin assignments, and peripherals.

<https://tinyurl.com/AD-codefusion-studio>



Mouser Electronics: Innovative Development Solutions

Every year, Mouser knows how to make their booth hugely popular. This time again, they consistently attracted long queues of attendees eager to take part in the "Spin to Win" prize giveaways, or play on the nostalgic "Pick, Pack, Stack" retro gaming kiosk, offering chances to win development kits and other prizes.

www.mouser.com



Watch this video!

Elektor TV at embedded world 2025

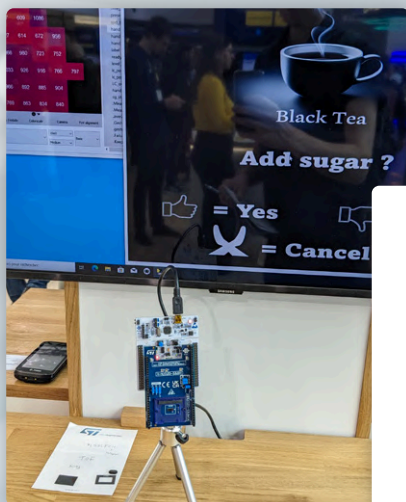
As always, we took our video equipment along, got some great footage and had very interesting conversations with exhibitors, including big names such as:

- > **Arduino:**
www.youtube.com/watch?v=NL5ocrFa8dE
- > **Espressif:**
www.youtube.com/watch?v=3A3fRV18N-o
- > **Raspberry Pi:**
www.youtube.com/watch?v=NujFl9MxbZo
- > **Texas Instruments:**
www.youtube.com/watch?v=WTlgMPQcGqA
- > **Edge Impulse:**
www.youtube.com/watch?v=bb6rACsZwhg

as well as ELATEC, QNX, AdaCore and more.

Visit our YouTube channels:

- > **Elektor TV:**
www.youtube.com/@ElektorTV
- > **Elektor Industry × eeNews Europe:**
www.youtube.com/@ElektorIndustry



STMicroelectronics: STM32 Demos and More

ST's massive booth featured numerous sub-stands dedicated to new products across a wide range of technologies. Around the microcontroller corner, one of the highlights this year was the STM32C0 microcontroller series, offering generous flash memory for graphical interfaces and integrated connectivity options such as USB and CAN. The full STM32 lineup is extensive, from the ultra-efficient STM32U3 to the high-performance STM32N6. Amongst the many live demos was an impressive gesture-controlled interface using a VL53L8 time-of-flight sensor, whose resolution almost made it look like a camera.

www.st.com/en/imaging-and-photonics-solutions/vl53l8cx.html

Würth Elektronik: WL-ICLED and More

Würth had a very busy booth at embedded world 2025, once again drawing attention with its broad portfolio of components for power electronics, automotive, wireless and other applications. A key highlight was the introduction of a reference design for a variant of Single Pair Ethernet (SPE), this time with Power over Data Lines (PoDL) to keep the wiring to its simplest form: a single twisted pair for power and data. They also presented their optoelectronics lineup with dual-wire ICLEDs, addressable LEDs using separate clock and data lines, which allows faster data transmission than regular 1-wire addressable LEDs.

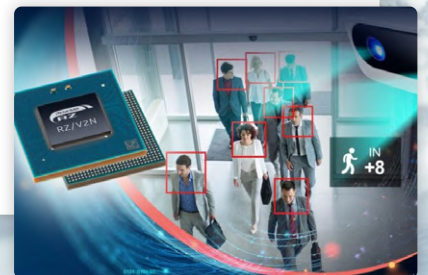
www.we-online.com/de/components/products/WL-ICLED

Renesas: DRP-AI Technology

Renesas Electronics has launched an integrated quad-core ARM microprocessor featuring its proprietary DRP-AI technology, designed for high-volume vision applications. The RZ/V2N employs the Dynamically Reconfigurable Processor DRP-AI3, delivering 10 TOPS/W and achieving AI inference performance of up to 15 TOPS through advanced pruning technology.

www.renesas.com/rzv2n

<https://tinyurl.com/renesas-DRP-AI>





Lattice Semiconductor: High-Performance FPGAs

Lattice showcased its continued expansion into higher-performance FPGAs, going well beyond its low-power roots. A standout was the Avant-X 70, one of the most powerful products in their range; it's a "mid-range" device by Lattice standards but offers an impressive 600,000 look-up tables (LUTs) and up to 28 (!) 25 Gbit/s interfaces, demonstrated here in a 5G base station application. As in previous years, Lattice was joined by numerous partners demonstrating real-world use cases. Among many of them, an interesting one was Secure-IC, a French company developing post-quantum cryptography solutions designed to resist future attacks by quantum computers.

www.latticesemi.com/Products/FPGAandCPLD/Avant-X

www.secure-ic.com

XMOS: Multipurpose Solutions

This year, the British fabless semiconductor maker XMOS proudly presented its xcore.ai platform, which are high-performance, low-latency, multipurpose processors tailored for intelligent IoT applications. The company also expanded support for high-performance audio through a new DSP Audio Processing library and example applications. Furthermore, they showcased AI-driven voice capture capabilities utilizing advanced denoise algorithms to ensure clear audio in challenging environments, and demonstrated networked audio over Ethernet for professional audio applications.

www.xmos.com/xcore-ai



Swissbit: Security Kit Upgrade

The Swiss company focuses on storage, security and embedded IoT solutions. One highlight was the Security Upgrade Kit. A Security Level 2 microSD card is a retrofit solution for Linux-based embedded systems, ensuring a high protection of critical data such as configuration credentials, licenses or externally stored data from being stolen, copied or manipulated. It also provides a reliable secure boot of crucial firmware and applications.

<https://tinyurl.com/swissbit-security-kit>



M5Stack: Modular Development Kits

M5Stack showcased its colorful, modular development kits aimed at rapid prototyping and product development. A key highlight was the new Unit C6L, combining the ESP32-C6 microcontroller with an SX1262 transceiver to enable LoRa connectivity alongside

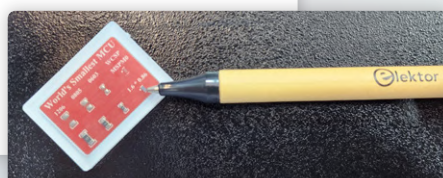
Wi-Fi 6, Bluetooth 5.3, Zigbee, and Thread support. Also on display was the Cardputer, a compact, card-sized computer featuring an ESP32-S3, keyboard, screen, and speaker designed for portable control and user interface applications. M5Stack also targets industrial applications with the StamPLC, a compact Programmable Logic Controller with opto-isolated inputs, relay outputs, and support for CAN and RS485.

<https://tinyurl.com/m5stamp-plc-controller>

Texas Instruments: Super-Small MCU

One highlight from Texas Instruments included what they were touting as "the world's smallest microcontroller," the MSPM0C1104. Looking at it next to an Elektor pen, it's hard to believe it sports I²C, SPI, 1 KB SRAM, 16 KB of flash storage, and runs at up to 24 MHz, among other features. Because it's smaller than a house ant, you would be well advised not to forget where you left it.

<https://ti.com/product/MSPM0C1104>

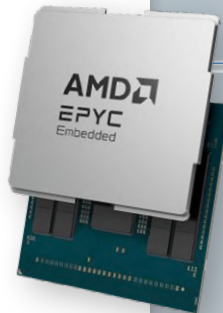


AMD: Embedded Processors

AMD has expanded its x86 embedded processor portfolio by introducing fifth-generation AMD EPYC Embedded processors. Powered by the proven "Zen 5" architecture, these embedded processors deliver leadership performance and energy efficiency, allowing networking, storage, and industrial edge systems to handle more data more quickly and with enhanced efficiency. The processors are designed for compute-intensive embedded systems with support for core counts from 8 to 192 in a single socket and a high data throughput, making the devices ideal for network and security firewall platforms, storage systems and industrial control applications.

www.amd.com

<https://tinyurl.com/amd-fifth-gen-processors>



Onsemi: iTOF Depth Sensors

The Hyperlux ID family of real-time, indirect time-of-flight (iToF) depth sensors deliver high-precision long-distance measurements and 3D imaging of fast-moving objects. The depth sensors can capture an entire scene while simultaneously processing depth measurements in real-time by employing a new proprietary global shutter pixel architecture and onboard storage. Hyperlux ID sensors enable depth sensing up to 30 metres, which is four times further than standard iToF sensors, all within a smaller form factor.

www.onsemi.com

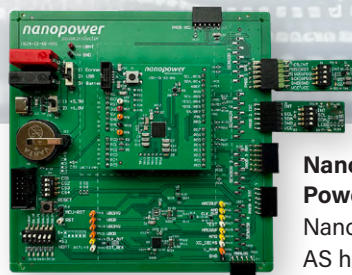
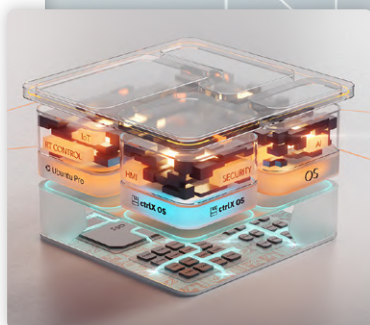
<https://tinyurl.com/onsemi-iToF>

Congatec: Computer-on-Modules

Industrial PCs form traditionally one big segment of the products showcased in Nuremberg. Bavarian company Congatec presented their aReady.COM product line, which stands for application-ready Computer-on-Modules. They come with soldered mass storage plus pre-installed, pre-configured and licensed operating systems like for example Ubuntu Pro. Pre-evaluated functional software building blocks significantly minimize design efforts for different use cases; including AI, Security, IoT, Human Machine Interface and many more.

www.aready.com

www.congatec.com



Nanopower Semiconductor: Power-Saving IC

Nanopower Semiconductor AS has announced the first evaluation kit to support its

breakthrough nPZero product, a power-saving IC which reduces the energy consumption of battery-powered applications by up to 90%. With the kit, developers can easily configure the nPZero power-saving IC to autonomously power up sensor peripherals, read data, and wake up the system as needed. The ultra-low-power nPZero IC powers down a microcontroller host, takes over the controller role and autonomously handles power-up, configuration, and reading data from up to four sensors or other peripherals.

www.nanopower.com

<https://tinyurl.com/nanopower-kit-nPZero>



Elektor Lab Talk: Live from embedded world 2025

We streamed a live episode of Elektor Lab Talk directly from the fair. Elektor editors Brian T. Williams and Jens Nickel were joined by Elektor author Stuart Cording. During the one-hour show, they shared their thoughts about the most interesting demos relating to embedded systems, and, of course, artificial intelligence (AI). This year, we also streamed a "recap" episode of Elektor Lab Talk around a week later, where we had as special guest Keith Jackson from Arduino.

> Elektor Lab Talk #31:

Live from Embedded World 2025:
www.youtube.com/watch?v=oJACP4ZET8

> Elektor Lab Talk #32:

Embedded World 2025 Recap:
www.youtube.com/watch?v=muDGTf_IswY

250162-01

Starting Out in Electronics...

...Continues with Tone Control

By Eric Bogers (Elektor)

In the previous installment, we made a tentative start on the application of active filters in tone controls.

Granted — audio puritans will now loudly exclaim that tone controls have no place in a good audio chain; that may be the case with high-end equipment, but in all other respects, there is nothing against adapting the reproduction characteristics of an amplifier to the listening room within certain limits. Anyway, tastes may differ, and Elektor simply offers “something for everyone,” so we continue to control our tones here.

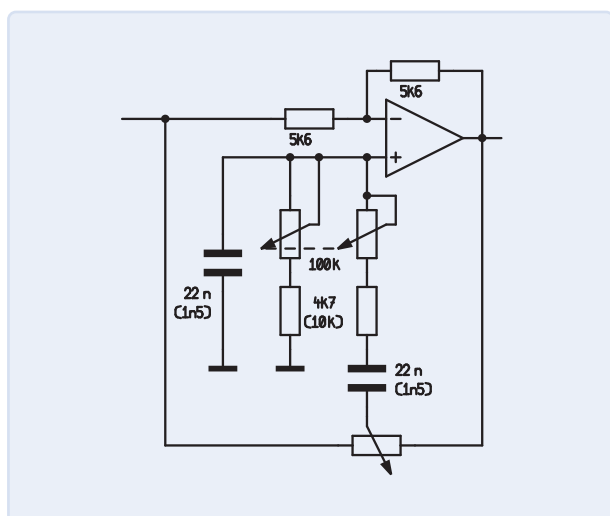


Figure 1: The midrange filter.

The Midrange Filter

After the treble filter of the previous episode, it is now the midrange's turn. The same circuit is used for both (indeed: two) midrange filters, albeit with different component values — see **Figure 1**. The component values in parentheses in the diagram apply to the high midrange.

Here, a “real” Wien filter was used as the frequency-determining network — i.e., a series circuit of two equally dimensioned RC networks (one parallel circuit and one series circuit). By means of a stereo potentiometer, both resistors can be regulated, and the center frequency of the filter is set. The resulting frequency characteristic is shown in **Figure 2**.

Basically, with this circuit, only a gain or attenuation of up to 10 dB is possible. A different value of the 5.6-k Ω resistors does not change this.

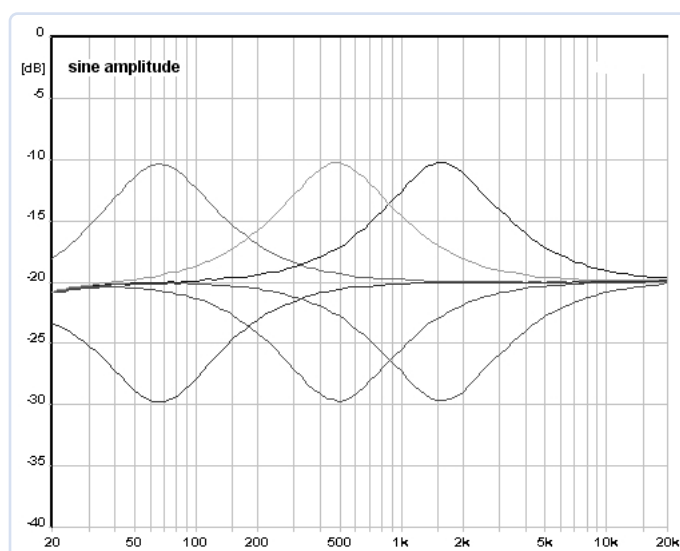


Figure 2: Frequency characteristic of the midrange filter.

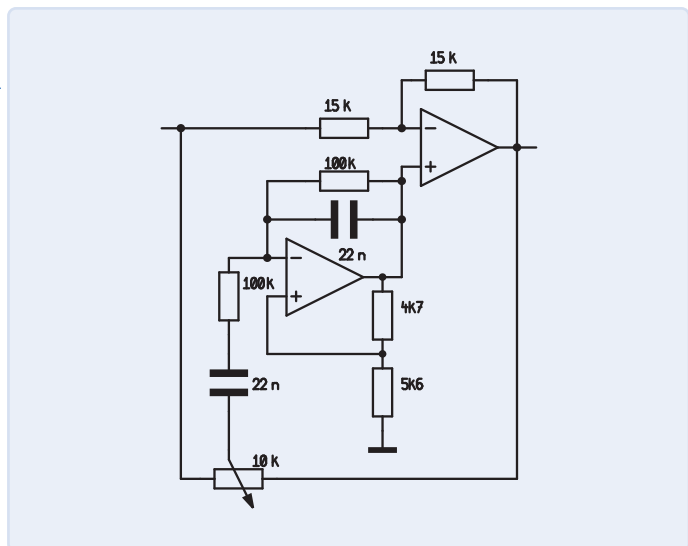


Figure 3: The bass filter.

The Bass Filter

The circuit of **Figure 3** used as a bass filter demonstrates how a wide control range can be obtained. Here, a second opamp has been used to improve the quality factor of the filter. If we would increase the resistance of 5.6 k Ω , the filter range could be increased to more than ± 16 dB.

Unlike the high-pass filter, the bass filter does not have a *shelving* characteristic (**Figure 4**). In our opinion, this would not make much sense either — why would we consider increasing the signal level in the frequency range of 20 to 40 Hz?

The State-Variable Filter

Normally, so-called state-variable filters are used for full-parametric tone control. These are universal filters that combine a high-pass, low-pass, and band-pass filter.

In **Figure 5**, you can see two characteristics for each of the three pass filters that result from a different filter quality (Q-factor).

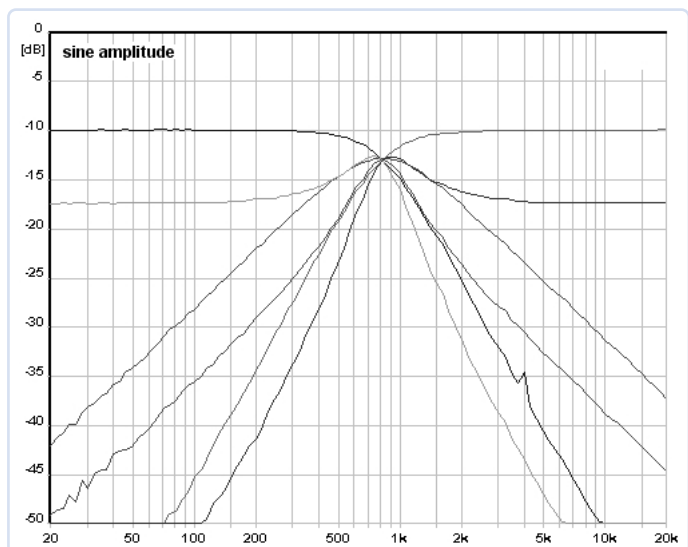


Figure 5: Frequency characteristics of the state-variable filter.

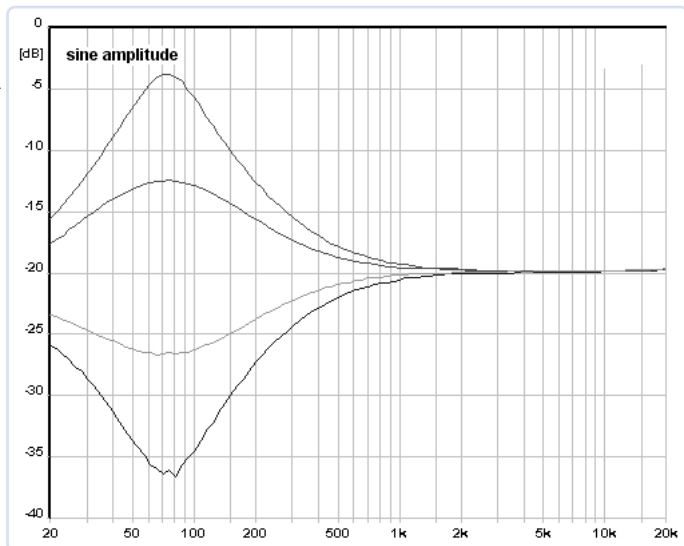


Figure 4: Frequency characteristic of the bass filter.

The slight ripple of the curves is a result of a somewhat sloppy measurement setup and is not an inherent property of this type of filter.

Figure 6 shows the principle diagram of the state-variable filter. This consists of two integrators whose signals are summed. Integrators behave like low-pass filters with, like other first-order filter circuits, a steepness of 6 dB/octave — but over the entire frequency range and not only noticeably above the cutoff frequency.

With two integrators connected in series, we obtain a second-order filter with a steepness of 12 dB/octave. When we subtract a signal filtered from the original signal in this way, the result will be a high-pass filter. And if we use the resulting signal to drive the two integrators, we obtain the characteristics of Figure 5.

The advantages of this setup are obvious:

- > Gain, filter quality and frequency can be set independently.
- > The gain and Q-factor are both set with only one resistor. For setting the frequency, we need a resistor for each filter section; thus, in the case of the usual second-order filters, a stereo potentiometer is needed.

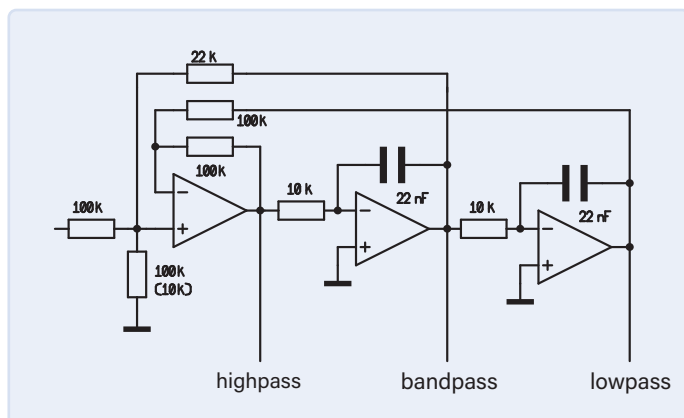


Figure 6: Principle diagram of the state-variable filter.

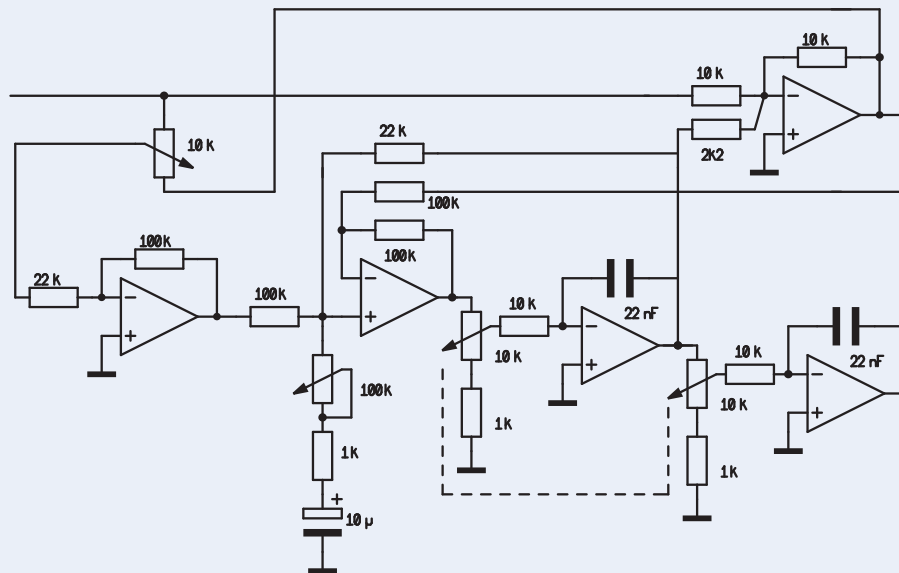


Figure 7:
Full-parametric filter.

- If the circuit is used as a crossover filter, then we always have the same cut-off frequency for the high-pass and low-pass filter, and in addition, the filter characteristics can also be changed very easily.

The State-Variable Filter as a Full-Parametric Filter

The schematic of **Figure 7** shows the state-variable filter as a full-parametric filter. Basically, we see here the normal setup of a filter, with the potentiometer runner between input and output. However, a state-variable filter is used here as a frequency-determining network that is broadly similar to the principle diagram of Figure 6. Two details are noteworthy:

- The filter quality is set with a potentiometer of 100 kΩ; the minimum adjustable value is 1 kΩ so that extremely “narrow” filters are possible.

- The center frequency is set with a stereo potentiometer; however, this does not vary the frequency-determining resistors (that is also a possibility) but rather sets the voltage with which that resistor is supplied.

Figure 8 shows the characteristic curve of the filter at different Q factors (100 kΩ, 10 kΩ, and 1 kΩ). The control range could be further extended (by setting the 22 kΩ resistor to a smaller value).

Equalizers

In principle, we could construct a graphic equalizer with the circuits discussed above. However, in PA technology, third-band equalizers are common, and if we were to connect 21 filter sections in series, we would have — despite modern low-noise op-amps — a (probably unsolvable) major signal-to-noise issue.

How can we address this problem? We will see in the next episode. ◀

Translated by Hans Adams — 250149-01

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.

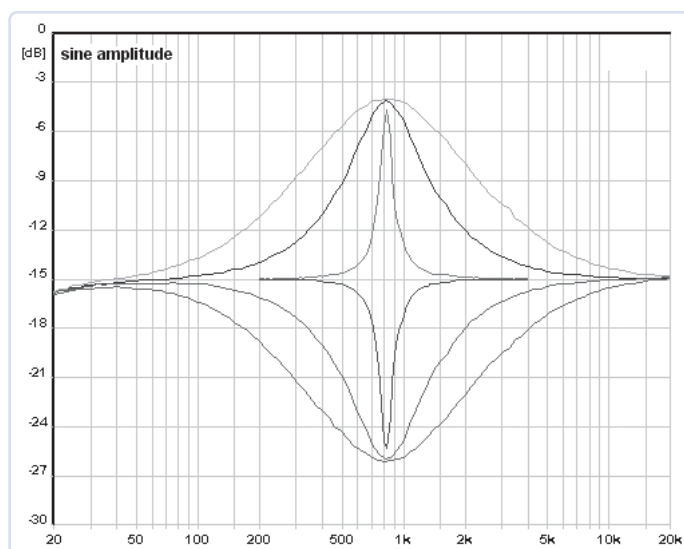


Figure 8: Different Q factors.

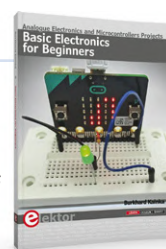
Questions or Comments?

Do you have technical questions or comments about his article? Send an email to Elektor at editor@elektor.com.



Related Products

- **B. Kainka, Basic Electronics for Beginners (Elektor, 2020)**
Book: www.elektor.com/19212
Ebook: www.elektor.com/19213



Academy Pro Box

Book + Online Course + Hardware



By The Elektor Team

Elektor is launching a unique combination of multimedia learning material and hardware: Academy Pro Box. This new all-in-a-box learning experience is aimed at electrical engineers and other professionals, as well as students and enthusiastic makers.

With decades of experience in electronics publishing and education, Elektor understands the challenges faced by engineers and makers in keeping up with rapid technological advancements. Traditional learning resources often lack real-world application, while online courses alone may not provide the hands-on experience needed to master electronics.

A New Standard in Technical Education

In today's fast-evolving world of electronics and engineering, professionals need more than just theory — they need practical, hands-on experience to stay ahead. With Academy Pro Box, Elektor brings together three essential learning components into a single, cohesive package:

- > **Expert Knowledge:** In-depth technical books and curated reading materials provide a strong theoretical foundation.
- > **Hardware:** Carefully selected electronic components allow users to apply concepts in real-world scenarios.
- > **Digital Learning:** Online courses, developed by industry specialists, offer structured learning paths, interactive content, and step-by-step guidance.

By integrating knowledge, practical application, and expert instruction, Academy Pro Box creates an engaging learning experience that is accessible, flexible, and result-driven.

Whether you want to expand your specialist knowledge, research new technologies or implement innovative projects, Academy Pro Box offers you the ideal support.

Audio Processing, Mastering Cortex-M Controllers, PCB Design with KiCad

These boxes are already available or will be available soon:

Red Pitaya Academy Pro Box

Master FPGA programming with Verilog and build a real-time audio processing system! This complete learning package includes industrial-grade hardware, a self-paced Udemy course, and step-by-step project instructions, making it the perfect choice for engineers, students, and electronics enthusiasts. Get started today and gain practical FPGA skills through hands-on learning!


NXP Academy Pro Box

Explore the power of NXP's FRDM-MCXN947 development board. Designed for embedded developers, this kit includes industrial-grade hardware, a guided learning course, and real-world projects to help you master ARM Cortex-M microcontrollers. Whether you're prototyping IoT solutions, developing real-time applications, or optimizing power management, the NXP Academy Pro Box provides the tools and knowledge to take your embedded development to the next level.

KiCad Academy Pro Box

This Academy Pro Box is your complete learning package for professional PCB design using the powerful KiCad EDA suite. The kit includes step-by-step tutorials, real-world projects, and expert guidance to help you design, simulate, and manufacture custom circuit boards. Whether you're a beginner or looking to refine your PCB skills, the KiCad Academy Pro Box provides everything you need to create high-quality designs for your next electronics project.

Elektor is already working on expanding the series, developing new learning modules, and forming strategic partnerships to further enhance the experience.

Find out more: www.elektormagazine.com/elektor-academy-pro 

250279-01



Want to Collaborate on an Academy Pro Box?

Are you an industry expert, educator, or company looking to develop customized educational material for professionals? We are actively seeking partners to co-create future Academy Pro Boxes tailored to specific technologies, industries, and training needs.

If your company needs high-quality technical training materials or if you want to contribute to the next box, reach out to us today via marketing@elektor.com!



Milliohmmeter Adapter

Uses the Precision of Your Multimeter

By Clemens Valens (Elektor)

There are several situations where you may need to measure low resistance values, down to the milliohm range. One example is locating a short circuit on a PCB, but it's also useful for testing electrical contacts, wires, and winding resistances. The adapter presented below allows you to do this together with the multimeter you already have.

Multimeters are handy devices that measure basic electrical quantities, such as voltage, current, and resistance. (I won't go into detail here, as you probably have at least one in your lab.) Typically, voltages and currents can be measured with good precision down to millivolts and microamps, but resistance measurements often stop at 1 Ω .

However, there are several situations where you may need to measure lower resistance values, down to the milliohm range. One example is locating a short circuit on a PCB, but it's also useful for testing electrical contacts, wires, and winding resistances. For this reason, I decided to build my own milliohmmeter.

The Approach

At first, I imagined an all-in-one meter with a display and advanced features, but while researching the topic, I came across an Analog Devices application note [1]. This simple circuit uses a multimeter as the display, significantly simplifying the design. Digging further, I eventually found an Elektor project from 1992 [2], which may very well have been the main inspiration for the 1998 application note from Dallas Semiconductor (which was acquired by Maxim in 2001, and

later by Analog Devices in 2021). That settled it. Instead of designing something entirely from scratch, and in the spirit of standing on the shoulders of giants, I decided to simply modernize the application note circuit.

The Circuit

The resulting schematic is shown in **Figure 1**. As you can see, it is quite simple (and not particularly creative, as it is just an updated copy of the original). Essentially, it is a programmable current source with three selectable settings: 1 A, 100 mA, and 10 mA. This provides three corresponding measurement ranges: 1 m Ω /mV, 10 m Ω /mV, and 100 m Ω /mV.

Switch S2 is the range selection switch. This 2P3T-type (or DP3T) slide switch can be found online as a hair-dryer spare part. Make sure to check the dimensions before ordering it as there appear to be several variants around carrying the same reference. An alternative is the L203011MS02 from C&K, which also fits on the PCB.

The current is applied via K1 to the resistance-under-test, such as a cable, a switch contact, a PCB trace, or even a resistor. This current creates a voltage drop across the

resistance, which can be measured using a multimeter set to its millivolt range. In a way, it serves as a practical demonstration of Ohm's law.

Note that the milliohmmeter adapter is suitable for occasional use. If you find you're using it frequently, you're probably better off investing in a dedicated milliohmmeter.

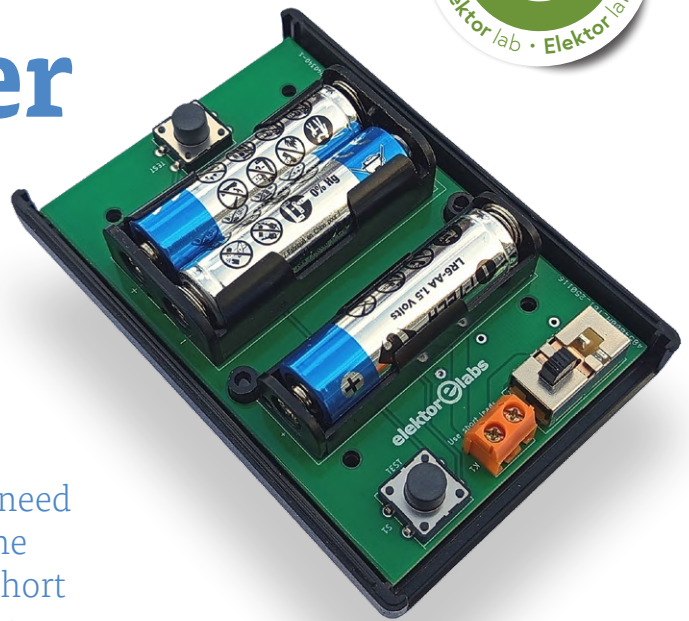
Programmable Current Source

The current source is a classic circuit built around IC2 and T1. The resistance from the source of T1 to GND determines the current. The voltage on the source of T1 should be set to 100 mV to obtain a 10 \times m Ω /mV scale (with x an integer). IC1 provides the reference voltage that can be adjusted with P1.

IC2 should have low input offset voltage. The schematic shows an MCP6401 which has a specified input offset voltage of 0.8 mV. I also tried the MAX4238 (0.1 μ V) and LTC2054 (0.5 μ V) but didn't notice any difference in performance, only in price. These op-amps all come in a SOT-23 package (6-pin for the MAX4238, 5 pins for the others) and therefore can be mounted on the PCB.

Battery Powered

The milliohmmeter adapter is powered from two 1.5-V AA cells (**Figure 2**). As the power supply is only applied when one of the *Test*



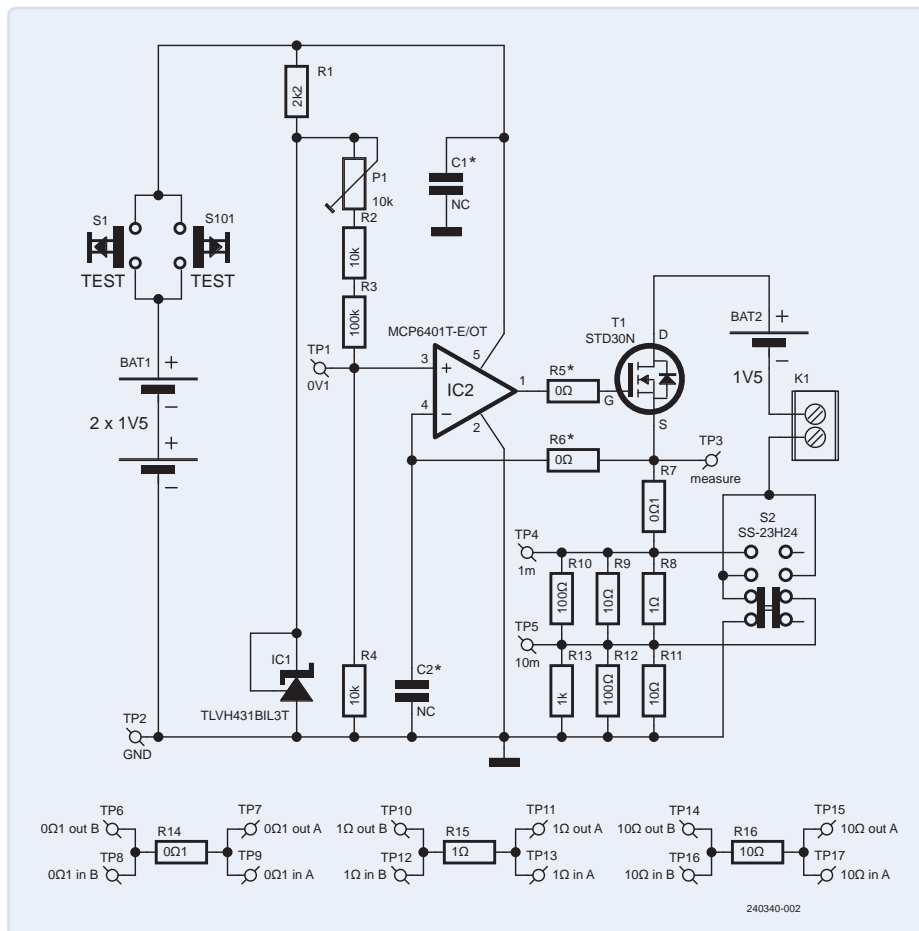


Figure 1: The milliohmmeter adapter circuit is a precise programmable current source.

pushbuttons is pressed, they will live a long and happy life. Life is a bit tougher on BAT2, especially in the 1 mΩ/mV position when it must deliver a current of 1 A. However, in most cases a test never takes longer than a few seconds, and so it will not suffer that much. I didn't change it once during my numerous experiments. A D-type cell will provide a longer lifespan, but it won't fit on the PCB.

For purely ergonomic reasons, there are two *Test* pushbuttons, S1 and S101, placed on either end of the board. It is up to you to decide which position you prefer.

Assembling the Adapter

As mentioned several times above, a printed circuit board (PCB) has been designed to hold all the components, including the batteries. It fits within one half of a low-cost plastic Hammond 1593N-type enclosure. This design protects the bottom-mounted components on the PCB (**Figure 3**) while keeping the batteries, range slide switch, and test buttons accessible.

The board includes multiple holes for securing the BAT1 holder with a nut and bolt. Properly

fastening it is highly recommended, as any movement can lead to a poor connection over time.

Calibration

The PCB has room for three calibration resistors, one for each range, R14, R15 and R16 in the schematic. These can be had as 0.1% types, allowing for precise calibration of the adapter. The first part of the calibration procedure is simple. Slide switch S2 to the 1 mΩ/mV position (1 A, closest to K1), connect short test leads to the 0.1-Ω calibration resistor R14 (contacts *in A* and *in B*), and connect a multimeter to measure the voltage drop over the resistor (contacts *out A* and *out B*). Press the *Test* button and adjust P1 to obtain a reading of 100 mV.

Calibration of the two other ranges is a bit more involved. A good way is to solder R8 and R11 first. Connect a resistor decade box or a trimmer in parallel to R8 and set S2 to the 100-mA position (middle). Connect the test leads to calibration resistor R15 and connect a multimeter to measure the voltage drop over it. Press the *Test* button and adjust the decade box or trimmer to obtain a reading of 100 mV.



Figure 2: The adapter is powered by three 1.5-V AA cells.

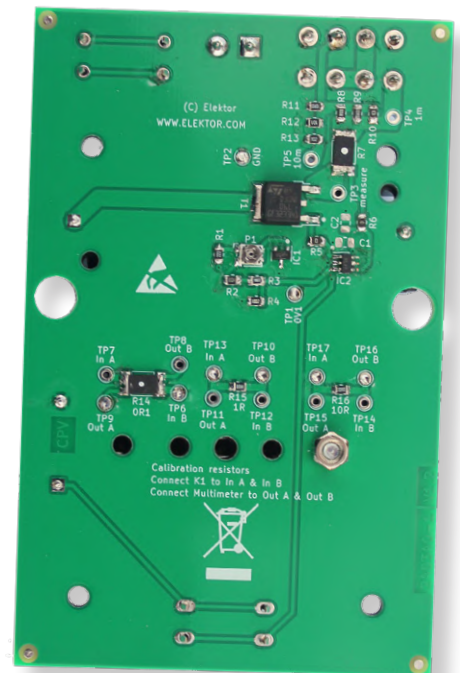


Figure 3: The components are mounted on the bottom of the PCB.

Measure the resistance of the trimmer or decade box and replace it by a parallel combination of two fixed resistors on positions R9 and R10. The 10-mA range is calibrated in the same way but now using R16, R12, and R13.

Another option is to only calibrate the 1-A range. Then, for the other two ranges, first measure a known resistor and divide the



Component List

Resistors (0805, 0.125 W, 5%)

R1 = 2.2 kΩ
 R2, R4 = 10 kΩ
 R3 = 100 kΩ
 R5*, R6* = 0 Ω
 R7, R14 = 0.1 Ω, 2512
 R8, R15 = 1 Ω
 R9, R11, R16 = 10 Ω
 R10, R12 = 100 Ω
 R13 = 1 kΩ
 P1 = 10 kΩ trimmer TC33X

Capacitors

C1*, C2* = 100 nF, 50 V, X7R, 0805

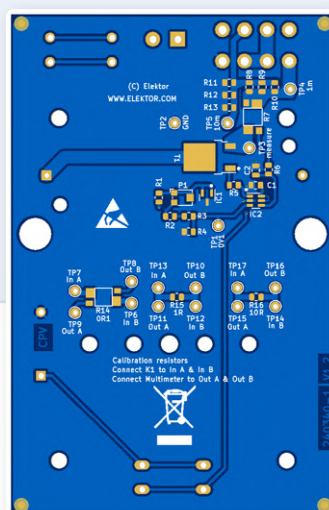
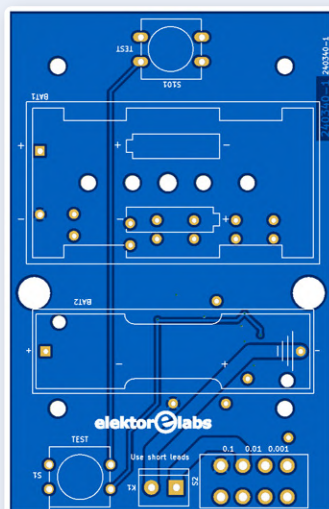
Semiconductors

IC1 = TLVH431BIL3T, SOT-23
 IC2 = MCP6401T-E/OT, SOT-23-5
 T1 = STD30NF06LT4, TO-252-2

Miscellaneous

BAT1 = Battery holder 2 x AA
 BAT2 = Battery holder 1 x AA
 K1 = Screw-terminal, 2 contacts, 5.08 mm pitch
 S1, S101* = Tactile switch, 12 mm
 S2 = DP3T slide switch (e.g., SS-23H24)

* = see text



real value by the measured value. This is the correction factor. Next, measure the unknown resistance and multiply it by the correction factor. Finally, apply the range multiplier.

Example: Suppose we have a reference resistor with a known value of 18 Ω. However, when measured in the 100 mΩ/mV position, we find 169.8 mV, which corresponds to 16.98 Ω. The correction factor therefore is $18/16.98 = 1.06$. Next, we measure an unknown resistance and find a value of 258.58 mV. Multiplied by the correction factor, we get 274.09 mV, and, after multiplying by the scale factor, we obtain 27.41 Ω.

The good news is that you only have to fully calibrate the adapter if you want to take precise resistance measurements. But then you must also use an accurate multimeter. For finding shortcuts or testing switch contacts,

and many other applications, the absolute value is less important than the relative value. When looking for a short, you just want to find the lowest value, no matter what it is. A tutorial on how to find short circuits on a PCB can be found at [3].

Using the Milliohmmeter Adapter

The milliohmmeter adapter uses a 4-wire (Kelvin wire) approach to measure resistance. The device injects the current into the resistance-under-test, and then the voltage drop over the resistance is measured using a multimeter. The advantage of this technique is that it eliminates the resistance of the multimeter leads, which results in more accurate readings.

The multimeter leads must be as close as possible to the contacts of the resistance-under-test to also eliminate component lead resistance. The placement of the adapter leads is not important (Figure 4). However, the leads from the adapter to the resistance-under-test should remain as short as possible. Long test leads with banana plugs will very likely make the adapter oscillate and the obtained readings will be useless. If you find unexpected values, check the leads.

Set the multimeter to the millivolt range to ensure that the resistance value will be displayed in millivolts. Multiply the readings by 1, 10 or 100 depending on the position of the slide switch to convert the values into milliohms. As an example, if a reading of 123 mV is obtained with the switch in the 10 mΩ/mV position, the resistance is 1230 mΩ, or 1.23 Ω. If the switch was in the 100 mΩ/mV position, the resistance is 12300 mΩ, or 12.3 Ω.

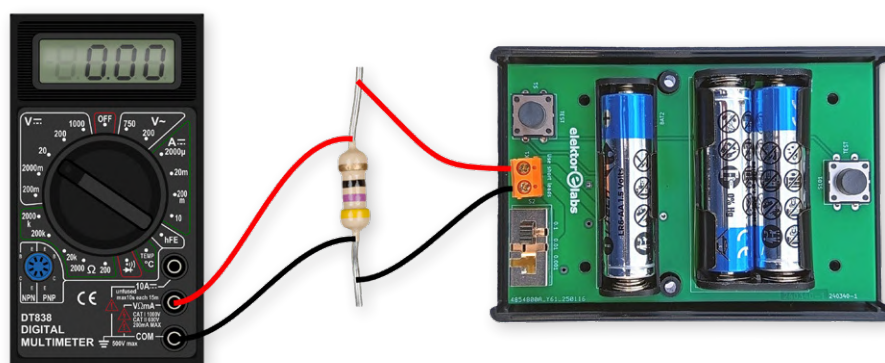


Figure 4: This is how the 4-wire Kelvin connection is used to measure small resistance values.



Open for Experimenting

For those who would like to experiment with the circuit, test points are provided on strategic positions. TP4 is for the 1-m Ω range (1m) and TP5 is for the 10-m Ω range (10m). Resistors R5 and R6 allow disconnecting the power stage from the op-amp output. R5 together with C2 let you create a low-pass noise filter in the (unlikely) case you need one. C1 is an optional decoupling capacitor. It is not needed when powered from batteries. Test points are available on the board on strategic positions, including ground. The KiCad project is available at [4]. ◀

240340-01

Questions or Comments?

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



Related Products

- > **Milliohmmeter Adapter**
www.elektor.com/21147
- > **OWON XDM1141 Multimeter**
www.elektor.com/20671

About the Author

After a career in marine and industrial electronics, Clemens Valens started working for Elektor in 2008 as editor-in-chief of Elektor France. He has held different positions since and currently is member of the product development department. His main interests include signal processing and sound generation.

WEB LINKS

- [1] Analog Devices, "Battery-Powered Circuit Measures Milli Ohms and Micro Ohms": <https://tinyurl.com/mw7usnfz>
- [2] Bernard C. Zschocke, "Milli-ohm measurement adaptor," Elektor 4/1992:
<https://www.elektormagazine.com/magazine/elektor-199204/32602>
- [3] Z. Peterson, "How to Test For a Short Circuit on a PCB," Altium, February 2018:
<https://resources.altium.com/p/how-test-short-circuit-pcb>
- [4] This project at Elektor Labs: <https://www.elektormagazine.com/labs/elektor-milliohm-meter>

JOIN OUR COMMUNITY

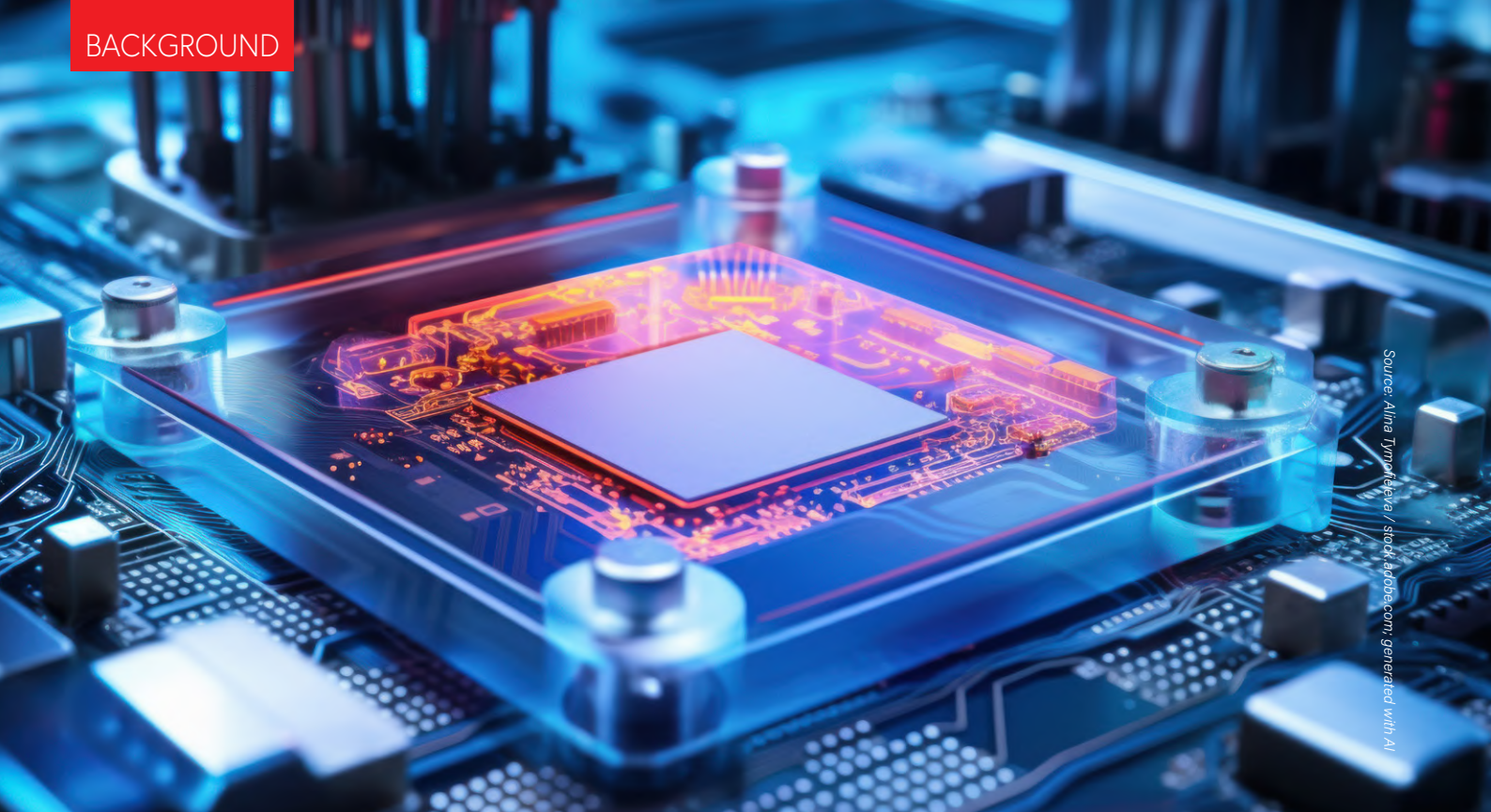


Subscribe today at
elektormagazine.com/ezine-24

GET FREE
DOWNLOAD



elektor
MEDIA & LEARNING



Source: Alina Tymkiv / Stock.adobe.com; generated with AI

The Next Leap in Semiconductors

Onward Toward 1.4 nm

By Mark Patrick (Mouser Electronics)

As 2025 races by and the 2 nm process node finally reaches production, the industry is already talking about what comes next: 1.4 nm.

That's what the semiconductor industry does — it prepares for the next revolution years in advance.

And so, designers straddle the need for miniaturization and efficiency, as always. But designers are not simply working to reduce the size of our electronics; designers are trying to surpass the limits of technology, and this requires smaller process nodes. It's not a trend as much as it is Moore's Law fulfilled. As we try

to make sense of these technological advancements, the semiconductor industry is on the brink of a significant milestone.

The 1.4 nm process node is expected to launch in 2027, promising a new age of semiconductor technology signified by excellent levels of transistor density, energy efficiency, and performance. The 1.4 nm node is the next phase in miniaturization that will redefine the constraints of computing power, opening the door to AI and quantum computing advancements.

Bringing the node to life requires advanced manufacturing technology, like the US\$400 million Extreme Ultraviolet (EUV) lithography machines developed by ASML [1]. These machines allow for the fine levels of precision needed to manufacture these process nodes and move us deeper into semiconductor evolution.

Let's take a look at how that evolution will play out and some of the implications of this leap in semiconductor technology.

The Role of EUV Technologies in Semiconductor Manufacturing

EUV lithography brings miniaturization to life. The processes conducted by these machines incorporate short wavelengths of light — much shorter than those used in traditional lithography — to etch fine patterns onto silicon wafers. Transitioning to EUV lithography has driven the miniaturization trend and allowed companies to pack billions more transistors into single chips. This was a more challenging task with previous Deep Ultraviolet (DUV) lithography technologies.

As mentioned earlier, though, these machines aren't cheap. Recently, Intel announced its heavy investment in High-NA EUV lithography machines from ASML [2], a move that shows off its pursuit of chips with cutting-edge 1.4 nm process nodes. Intel recognizes these machines' role in advancing chip manufacturing and is preparing to make them a big part of its strategy.

The Development of Transistor Density

When the integrated circuit was created in the 1950s, it housed just a handful of transistors. Today, chips contain billions of these units, forcing a transition from the microscale to the nanoscale, as well as some new manufacturing capabilities and theoretical understanding. The evolution continued from the 1950s to the 1970s with the 10 micrometer processes and continued up to the sub-10 nanometer processes in recent years. Each time designers reduce transistor size, we see a corresponding surge in computing power, efficiency, and complexity of electronic devices.

Transitioning to the 1.4 nm process node will be the most significant leap in transistor density and chip performance to date. It means an unheard-of number of transistors packed onto a chip, but it also presents some challenges in materials science and fabrication techniques. To overcome these challenges, the industry must innovate how semiconductors are manufactured, including lithography advancements and new materials that can manage heat and behavior at quantum levels.

Our current understanding of transistor density needs to be adjusted. It's not about the number of transistors on a unit, but perhaps the number of transistors per footprint when considering 3D volume. Applications such as virtual reality (VR), augmented reality (AR), and autonomous vehicles are examples of how

crucial specialized transistor applications are, as they rely on advanced semiconductor systems powered by a complex web of transistors.

This shift exemplifies a rebirth in transistor invention, where the challenge lies not just in making transistors smaller, lighter, and cheaper, but in co-designing hardware and software to model the world more effectively. As designers approach the anticipated 1.4 nm process node and look beyond, the semiconductor industry inevitably will transform, where system-centric chipset designs enabled by advanced transistors could redefine our capabilities.

Intel Leads the Charge

Currently, Intel uses EUV technology across three fabrication plants: Intel 4, Intel 3, and Intel 20A. These facilities contribute to approximately 15 percent of the company's total wafer output by volume. Despite the dominance of DUV lithography in its Intel 7 process, the company anticipates a shift towards EUV-based nodes soon [3]. Intel is taking a proactive approach to meeting the demand for semiconductors, especially with the growing importance of AI accelerators.

To adapt to this expected demand surge, Intel is not just focusing on transitioning to more advanced lithography techniques, but also expanding its manufacturing capacity. The company is placing a strong priority on advanced packaging techniques, too [4].

Beyond 1.4 nm

Yes, Intel's already thinking past 2027. Because, as the industry approaches the 1.4 nm process node, it also nears the physical and technical limits of silicon-based technology. While silicon is the backbone of



The journey beyond the 1.4 nm node is not about how much designers can shrink electronics, but a mission to redefine the makeup of computing technology.

electronics, it's at risk. It encounters challenges like quantum tunneling and heat dissipation that are problematic at smaller scales. This complicates things as the industry aims to miniaturize further, all while threatening the efficiency and reliability of chips. These hurdles for alternative materials to help carry on Moore's Law.


The good news is that the industry is already working on it, as it usually does. Emerging tech such as two-dimensional materials, like graphene and transition metal dichalcogenides, are ready to become potential successors to EUV lithography due to their electrical thermal, and mechanical properties at atomic thicknesses.

Also, quantum dots hold the potential for advances in photonics and quantum computing, offering new ways to process and store information. In addition, advanced packaging technologies, such as 3D integrated circuits (IC), offer solutions to combine multiple chiplets or dies into a single package to enhance performance and functionality without shrinking components.

What Do Advancements Beyond 1.4 nm Mean?

If, and when, we do move past 1.4 nm, we open up new possibilities to transform computing. AI systems become more powerful and capable of complex reasoning while consuming less energy. Quantum computing, a beneficiary of quantum dots and quantum materials advancements, has become more practical and can even solve problems that remain difficult for classical computers. Ultimately, if we move toward new materials and solutions, we can sustain the pace of miniaturization while improving energy efficiency.

Redefine the Makeup of Computing Technology

The journey beyond the 1.4 nm node is not about how much designers can shrink electronics, but a mission to redefine the makeup of computing technology. The promise of the 1.4 nm node by 2027 brings about much excitement, but not because of what it can do now, but rather because of the stepping stone it represents. With companies like Intel investing in cutting-edge EUV lithography and advanced packaging technologies, the industry is destined for a transformation. As we approach and look beyond 2027, the anticipation for the 1.4 nm node promises to bring in new realms of AI and quantum computing, powered by unprecedented efficiency and processing power. 

250220-01



About the Author

As Mouser Electronics' Director of Technical Content for EMEA, Mark Patrick is responsible for creating and circulating technical content within the region — content that is key to Mouser's strategy to support, inform, and inspire its engineering audience. Before leading Technical Content, Mark was part of Mouser's EMEA Supplier Marketing team and played a vital role in establishing and developing relationships with key manufacturing partners. Mark's previous experience encompasses hands-on engineering roles, technical support, semiconductor technical sales, and various marketing positions. A "hands-on" engineer at heart, Mark holds a first-class Honors Degree in Electronics Engineering from Coventry University. He is passionate about vintage synthesizers and British motorcycles, and thinks nothing of servicing or repairing either.

WEB LINKS

- [1] A. Friedman, "Intel video shows the \$400 million tool that will help it build more powerful chips," PhoneArena, March 2024: https://www.phonearena.com/news/intel-video-shows-installation-of-new-lithography-machine_id155941
- [2] C. Trueman, "Intel acquires ASML's entire 2024 stock of High NA EUV machines," Data Centre Dynamics, May 2024: <https://tinyurl.com/intel-acquires-high-na-euv>
- [3] "Intel reaches 3nm milestone," Global SMT & Packaging, June 2024: <https://www.globalsmt.net/advanced-packaging/intel-reaches-3nm-milestone>
- [4] Intel, "Intel Launches World's First Systems Foundry Designed for the AI Era," February 2024: <https://newsroom.intel.com/artificial-intelligence/foundry-news-roadmaps-updates#gs.e94taz>

Through-Hole Technology Connectors

The Best of Two Worlds: THR

By Fabian Altenbrunn and Markus Hildmann (Würth Elektronik eiSos)

Through-hole reflow (THR) combines the advantages of through-hole technology (THT) and surface mount technology (SMT), thereby saving both time and cost in the production process of electronic assemblies. THR is particularly attractive for connectors, but there are a few things to bear in mind.

Ongoing cost pressure is forcing electronics manufacturers to continuously adapt their production processes and make them more efficient. Although advancing miniaturization, coupled with constantly increasing packing density, is driving developments along the SMT production line relentlessly. So far, not all components are suitable for surface-mount placement technology. On the contrary, through-hole technol-

ogy, which was thought to be obsolete, will continue to have a right to exist in the future. The same applies to manual placement — even though odd-shape placement machines with customer-specific nozzles are becoming increasingly established in electronics production.

Manufacturers are working intensely on connectors that are SMT-compatible and

can withstand high mechanical loads. With multi-pin connector types in particular, there is a risk that the soldered connection will not be able to withstand the high insertion and withdrawal forces in the long term. Through-hole mounting ensures a robust and reliable connection to the PCB. However, THT assembly is usually still carried out by hand in order to prevent damage to components that have already been soldered onto the board. This applies in particular to PCBs assembled on both sides.

Taking THR Special Features Into Account

THR technology meets these challenges. It is able to process THT components together with SMT components in the same reflow soldering process, which saves manufacturing costs in two ways. On the one hand, additional soldering steps required for THT,

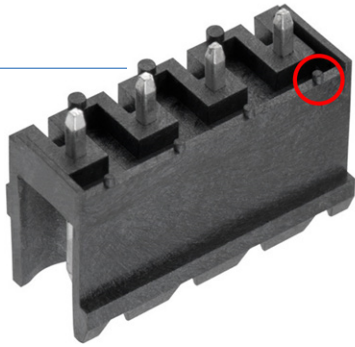


Figure 1: Standoffs on a THR-compatible terminal block ensure that the component does not come into contact with the solder paste.

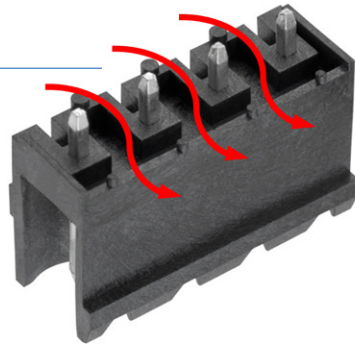


Figure 2: The standoff on the underside of the terminal block creates airflow channels during the reflow soldering process.

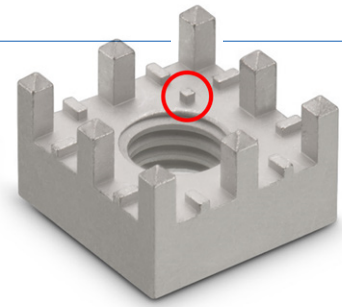


Figure 3: The metal Redcube THR components have standoffs. They ensure air circulation and ensure that the component is heated through.

such as wave, selective, or even manual soldering, are no longer necessary. On the other hand, completely automated assembly of the electronic module can be realized — without the previously necessary ejection to a manual workstation.

But not all connectors are the same. THR requires some adjustments that are often overlooked. So-called automated solder-in connectors must be able to withstand the high thermal stresses of the reflow soldering process. To make this possible, the connector materials and their geometry need to be adapted. On the PCB side, the THR technology must be taken into account in both the assembly and the print stencil layout. It is also important to select the correct solder paste and the correct application volume on the solder pads. Finally, the standard SMT reflow soldering profile needs to be revised. But first things first.

How Does a Component Become THR Capable?

The production of a THR-compatible connector requires changes to the design and material of both the housing and the contacts. The housing requires a plastic that can withstand the high reflow soldering temperatures of +245°C that are common. According to the IPC/JEDEC J-STD-020 standard, the plastic must even be suitable for a temperature range of up to +260°C. However, there are two key aspects to consider when it comes to temperature exposure. The first is related to the effects of soldering temperatures on the moisture content of the polymer. The absorbed moisture expands under the influence of the soldering temperature, creating bubbles and inclusions in the housing. The second effect is the melting and deformation of the housing due to the effect of temperature.

How THR Works

As THR components and SMT components are supplied in tapes on reels, for the feeders of the placement machines and applied to the pre-stenciled PCB solder pads using nozzles or pipettes, fully automatic placement is possible. A joint reflow soldering process is also guaranteed. The soldering process begins with the liquification of the solder according to the solder profile. While SMT components are held on the board by the surface tension of the solder, the solder paste of the THR components flows into the drill holes due to the capillary effect and thus forms a secure solder joint.

Würth Elektronik uses liquid crystal polymers (LCP) for the housing of its THR-compatible connectors (e.g., the WR-TBL terminal block family), which are particularly characterized by their high-temperature resistance and dimensional stability. The low coefficient of thermal expansion is particularly important when mechanical stresses caused by a mismatch in thermal expansion come into play. Moreover, the expansion behavior of LCP is similar to that of the PCB material FR4. Furthermore, LCP has low moisture absorption meaning it does not have to be tempered first and can be stored indefinitely.

In addition to the housing material, the housing design must also be adapted. Standoff pins are required on the underside of the housing (Figure 1) to create space for the solder paste and to promote the flow of air under the housing (Figure 2). This not only facilitates the melting of the solder paste in the reflow soldering process, it also ensures optimum heat transfer. It is important that the component does not come into contact with the solder paste. Contact between the standoffs and the solder paste should be avoided in order to minimize the risk of solder bead formation. In addition, this standoff is required for the ideal shaping of the solder meniscus on the top of the PCB. Even with a metal housing, spacers must be incorporated, as the amount of air flow is not sufficient to heat the component.

Würth Elektronik has also ensured THR compatibility for its Redcube THR family (WP-THRBU and WP-THRSH) designed for robust high-current connections (Figure 3) and for the coaxial connectors (WR-SMA and WR-SMP) by incorporating standoffs at dedicated points. The ideal distance between the underside of the housing and the solder paste deposit should be at least 0.5 mm.

Finally, in order to provide the required compatibility for the SMT production process, the housing shape of the THR connectors should be suitable for the SMT vacuum pipette and - if necessary - equipped with suction pads. It must also be taken into account that the placement process limits the maximum connector size that can be used.

THR-Compatible Contact Pins

In addition to the housing geometry, the connection ends of the contacts also play a major role. This mainly concerns the length. If the connection ends are too short, the stability of the contact pin in the through-hole may be impaired, which can lead to problems during the reflow soldering process. A pin that is too long, on the other hand, pushes the solder paste through the through-hole in the PCB, which means that there is not enough solder left in the hole to securely enclose the pin. In addition to the effect that solder beads form at the end of

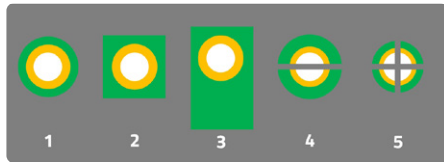


Figure 4: Apertures 1 and 2 are typical for THR printing. If additional solder paste is required, aperture 3 is useful, while apertures 4 and 5 are designed for large pads.

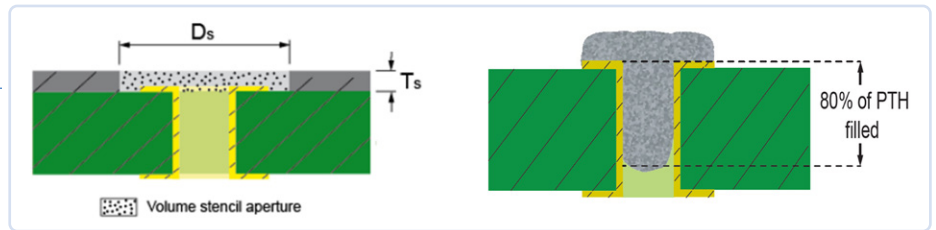


Figure 5: As a rule of thumb, in order to achieve a filling rate of at least 80%, the amount of solder paste must be twice the free volume. The reason for this is the solder paste composition of metal particles and flux. These are mixed together at a 50:50 volume ratio and the flux evaporates completely during the soldering process.

the pin, the formation of the typical solder eye is hindered or even prevented.

The end of the pin protruding from the PCB in the solder must therefore be recognizable and must not exceed 1.5 mm. According to the DIN EN 61760-3 standard, the pin should protrude at least 0.5 mm from the underside of the PCB. The THR contact pins from Würth Elektronik have a pin length of 2.6 mm to ensure compatibility with PCB thicknesses of between 1.6 and 2.3 mm. The centering of the pin in the plated-through hole is also very important for optimum solder flow.

The contact surface in combination with the PCB surface finish is also relevant. The solderability requirements of THR technology may make it necessary to switch from bright to matt tin plating. Insufficient control of the bright additive sometimes impairs the solderability of

bright tin-plated pins, as an oxidation layer can form when the connectors are stored. In today's electronic assembly production, tin is the first choice due to its wetting properties. However, tin can be bright or matt. Matt tin plating is unsuitable for high temperatures, while bright tin tends to darken under the influence of high temperatures. If immersion tin or an ENIG alloy is used as the PCB surface finish, the pin must first be nickel-plated to a thickness of at least 1.25 μm before tin or gold can be applied.

Layout Recommendations for THR

Because the contact pins can be round or any rectangular shape, the pin geometry plays an important role as it has an effect on the pad size and pad geometry on the PCB as well as in connection with the apertures on the print stencil. With sufficiently large solder pads on the top and bottom of the PCB, the plated-through holes should only be 0.2 mm larger in diameter than the pin diameter to be soldered. This increases the stability between the PCB and the connector. The size and geometry of the apertures of the printing stencil (aspect ratio) ultimately determine the volume of solder paste that is applied to the solder pads on the PCB (Figure 4). How well the solder paste fills the through-hole and the print stencil aperture depends on several factors; the choice of solder paste and the speed, pressure, and contact angle of the squeegee used to apply the solder paste to the PCB. In addition, the paste release behavior has a considerable influence on the reliability of the solder joint. Attention should therefore be paid to the surface quality of the print stencil in the apertures, but also to the correct area ratio (the ratio between the base area and the wall area of the stencil opening). With large pads, there is a risk of the squeegee dipping in and pulling the paste out again. To counteract this, squeegee bars are useful — either as a cross or in the direction of the squeegee.

The solder paste deposit must be calculated in advance to ensure that the through-hole is sufficiently filled and that a fillet or two fillets can form. Example: With a 150 μm thick printing stencil, around 80% of the hole is filled (Figure 5). This value varies depending on the thickness of the blank and stencil. The surface of the PCB also has an influence on the volume of the solder paste deposit. This can be achieved by overpressure of the solder paste on the top of the PCB, sufficient filling of the via hole, and a maximum through-print of 1 mm. From a via diameter of more than 0.8 mm, solder paste types 4 and 5 are recommended. This opening with the ridges can prevent dripping with a large via. ◀

250237-01



About Markus Hildmann

M. Eng. Markus Hildmann studied Computer Engineering and Systems Engineering at the Technical University of Nordhausen. Then he worked for 10 years as a Hardware Developer in the field of battery management systems. Since 2022, he has been employed as a Field Application Engineer at Würth Elektronik eiSos GmbH & Co. KG in the field of electromechanics.



About Fabian Altenbrunn

Fabian Altenbrunn (B.Sc.) completed Dual studies in electrical engineering at HS Schmalkalden. He worked as a Process Engineer, Product Manager, and Test Developer. Since 2019, he has been a Field Application Engineer for key account customers at Würth Elektronik eiSos GmbH & Co. KG.

Saving time and costs with THR technology

The demands on electronic assembly production are increasing. Ever higher quality at ever lower prices is driving the demand for process optimization. THR is becoming increasingly attractive because the technology combines the best of both worlds, THT and SMT. THR is a sensible alternative to THT, especially for plug-in connectors: As components in through-hole assembly, they can be placed on the PCB in the same way as SMT components using automatic placement machines and soldered in the same reflow soldering process. The machine-compatible process capability of the THR connector type means that a high level of reliability can be achieved in module production. However, THR technology places special demands on the component, layout and solder paste application. The cover story deals with the necessary design and material requirements.

Frequency Counter

Portable and Auto-Calibrating
Via GPS



By Willem den Hollander (Switzerland)

The major limitations of an inexpensive frequency counter are the poor medium- to long-term stability and the lack of an internal reference source for calibration. The project in this article overcomes these drawbacks by implementing a very stable TCXO and a highly accurate satellite reference source.

Affordable frequency counters often lack precision and stability. Here, a frequency counter is presented with a guaranteed eight-digit accuracy and of acceptable cost. An OLED display provides excellent readability. Calibration is automatic by use of the GPS signal. A built-in rechargeable battery makes the instrument portable.

Principle of Operation

The so-called reciprocal counter method is used. A number of complete cycles of the input signal is counted. During the period this takes place, the cycles of a reference signal with a precisely known frequency, the reference frequency (f_{ref}), are counted. With the formula

$$f_{in} = \frac{n_{in}}{n_{ref}} \times f_{ref}$$

the frequency of the input signal (f_{in}) is calculated. For acquiring the number of input cycles (n_{in}) and the number of cycles of our reference frequency signal, n_{ref} counts are required.

If we want a result with an accuracy of eight digits n_{ref} should be greater than 10^8 .

Thus, the value of the reference frequency defines the duration of the period during which the cycles of the input signal are counted. This time is the minimum time to measure the frequency of the input signal with 8-digit accuracy. Therefore, the higher the reference

frequency is, the faster we obtain a result. For calculations, a microprocessor is required. Modern microcontrollers run at high frequencies and often have various counters on-chip.

Design Considerations

Microchip's PIC18F26Q83 microcontroller has all the features required for this application. It operates at a maximum frequency of 64 MHz. Processors operating at a higher frequency are available at a considerably higher cost and are too complicated for the application described here.

A Temperature Compensated Crystal Oscillator (TCXO) provides the 64-MHz clock for the processor, since the internal oscillator would not be sufficiently stable. With this frequency, a counter with eight-digit accuracy can be achieved with an update time of less

Specifications

Sample Rate	1.6 s
Time Base	~64 MHz GPS calibrated
Frequency Range	5 Hz to 64 MHz
Resolution/Accuracy	8 digits
Input Impedance	1 MΩ/15 pF
Input Sensitivity	50 mVpp
Max. Input Voltage	30 V
Power Supply	Rechargeable Li-Ion battery
Battery Operation	~20 hours

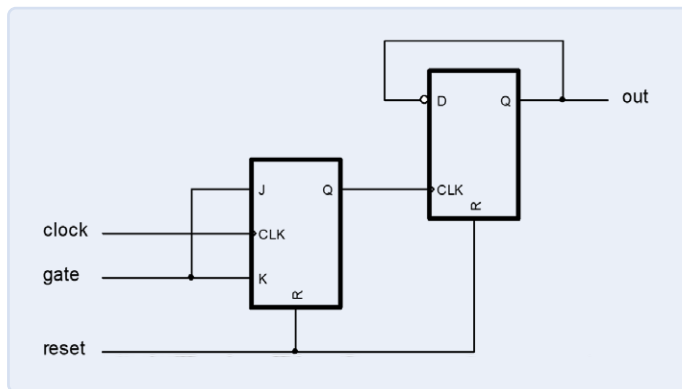


Figure 1: The 2-bit prescaler.

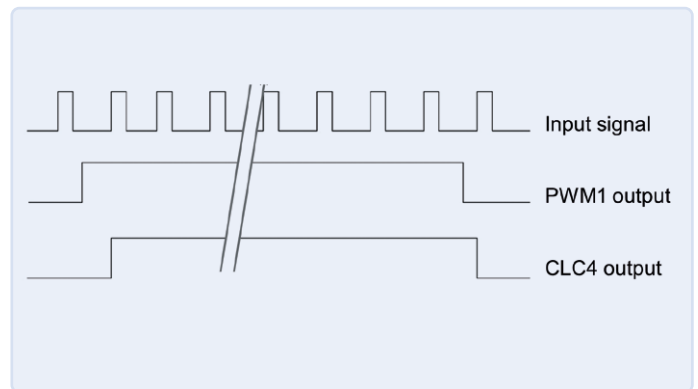


Figure 2: Waveforms of input signal, PWM1 output, and CLC4 flip-flop output (see text).

than 1.6 seconds. However, the timers of the processor can only handle signals with a maximum frequency of 20 MHz.

Fortunately, there are eight Configurable Logic Cells (CLC) on the chip, and they are fast enough to cope with signals with a frequency of 64 MHz. This is then the maximum frequency we can measure with this processor. With two CLCs, we make a gated 2-bit prescaler (see **Figure 1**) of which the output has then a frequency of maximum 16 MHz. This is acceptable for all the counters inside the microprocessor.

As already mentioned, we have to count for at least 1.6 s. Counting 64 MHz cycles for such a period requires a counter with a minimum of 27 bits. Two bits are provided by the prescaler and an internal counter should be at least 25 bits wide. Since the counters are all 8-bit wide or a multiple of this, we use 32-bit counters. The processor provides six 16-bit timers: TMR0, TMR1, TMR3, TMR5, UTMRA and UTMRB.

These timers can be chained to make 32-bit counters. Here we have chained the timers TMR1 and TMR3 and timers UTMRA and UTMRB. CLC5 and CLC6 are the prescaler for the 32-bit UTMR and CLC7 and CLC8 act as the prescaler for the TMR1/TMR3 combination. With the PWM1 module, a 1.6 s pulse is generated. As soon as a cycle of the input signal arrives, the flip-flop of CLC4 is set (see **Figure 2**).

This flip-flop provides the gate signal for the first stage of the prescalers. It is reset by the first cycle of the input signal after the PWM1 output goes low. The output of CLC4 going

low causes an interrupt. In the interrupt service routine, the counters are read, the calculations made, and the results displayed. After this, the counters are reset, and the procedure described starts again.

Precision is achieved by using a very stable 64-MHz TCXO of which we measure the frequency very accurately through the GPS signal. The output of the GPS receiver module used is a signal with an extremely stable (better than 10^{-15}) 1 Hz frequency. With 8-bit counter TMR2, we make a 10 s pulse. During this time period a prescaler made by CLC1 and CLC2 counts the TCXO output pulses. The output signal of this prescaler serves as the clock signal for the 32-bit counter made up by chained counters TMR0 and TMR5.

The output signal of TMR2 will cause an interrupt as soon as it goes high. The interrupt

service routine will read the value of TMR0/TMR5 counter, which represents exactly ten times the frequency of the TCXO. This value is used in the calculations during the frequency measurements. A possible variation of ± 1 count has no influence on the 8-digit accuracy of the counter, since the measuring period is 10 seconds. After measuring the TCXO frequency a few times, the result is stored in the EEPROM of the processor. This assures — when the frequency counter is turned on — that this value is available. Practice shows that only aging changes the frequency of the TCXO.

The Circuit

As illustrated in **Figure 3**, the microcontroller takes care of all logic functions and calculations. But a frequency counter needs more. The signals — whose frequency has to be measured — are not necessarily digital.

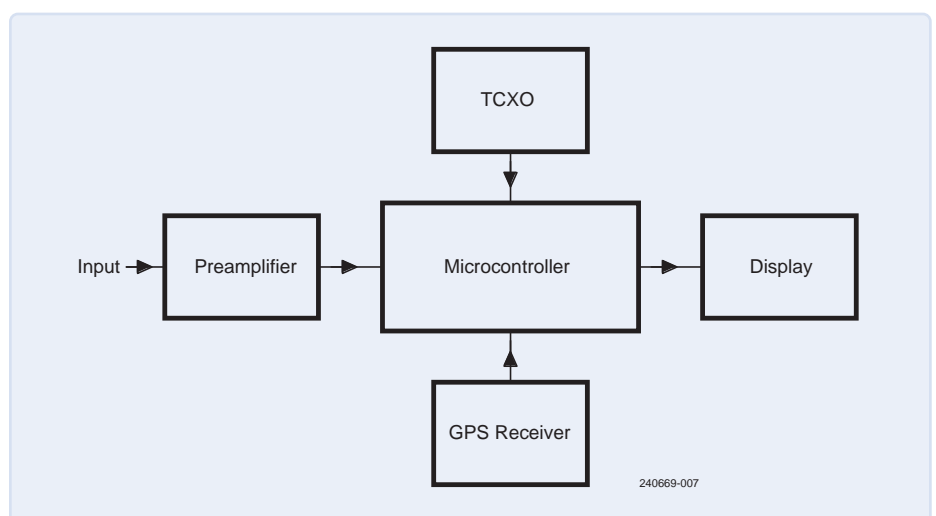


Figure 3: Block diagram of the frequency counter.

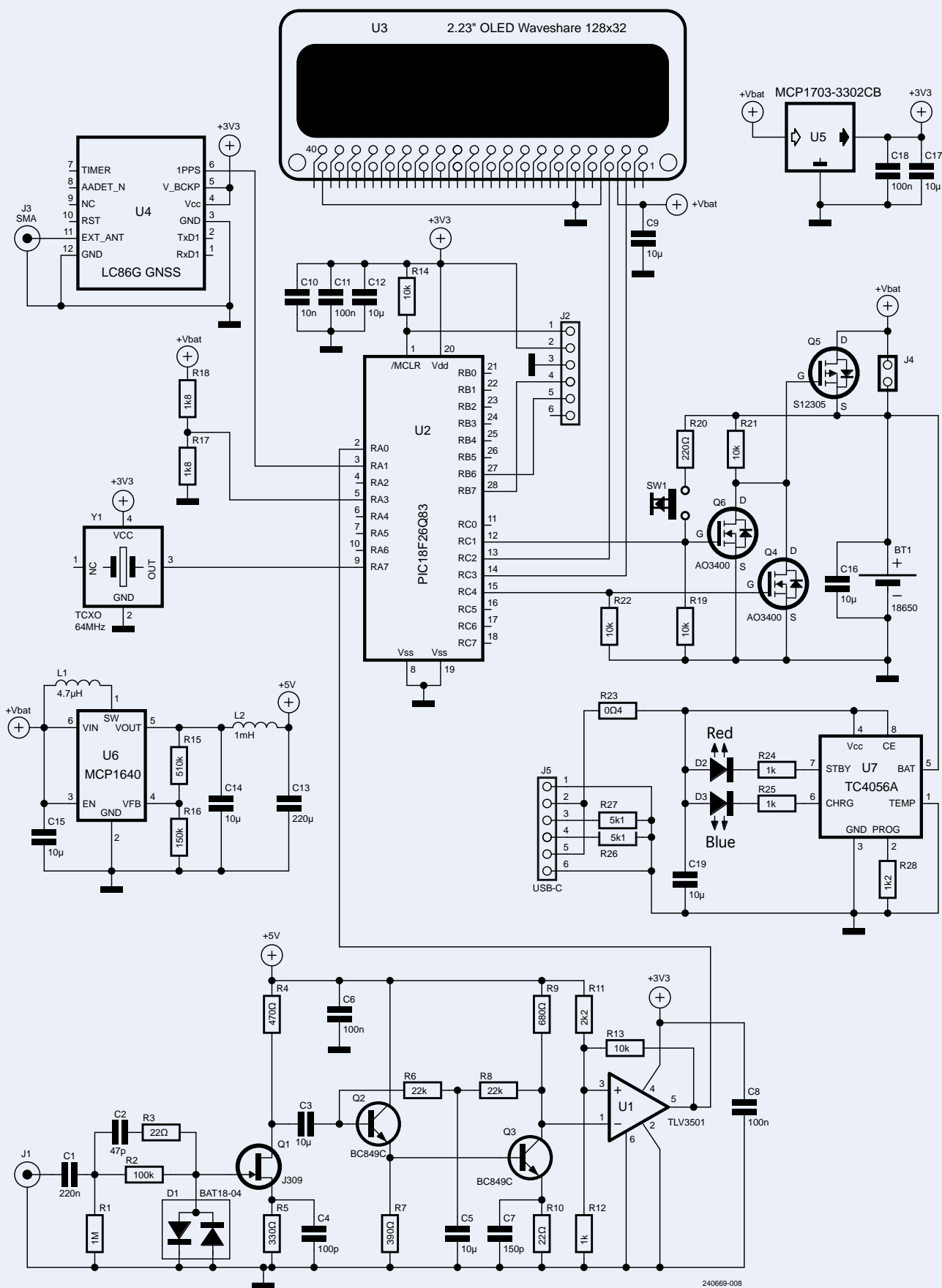


Figure 4: Schematic diagram of the frequency counter.

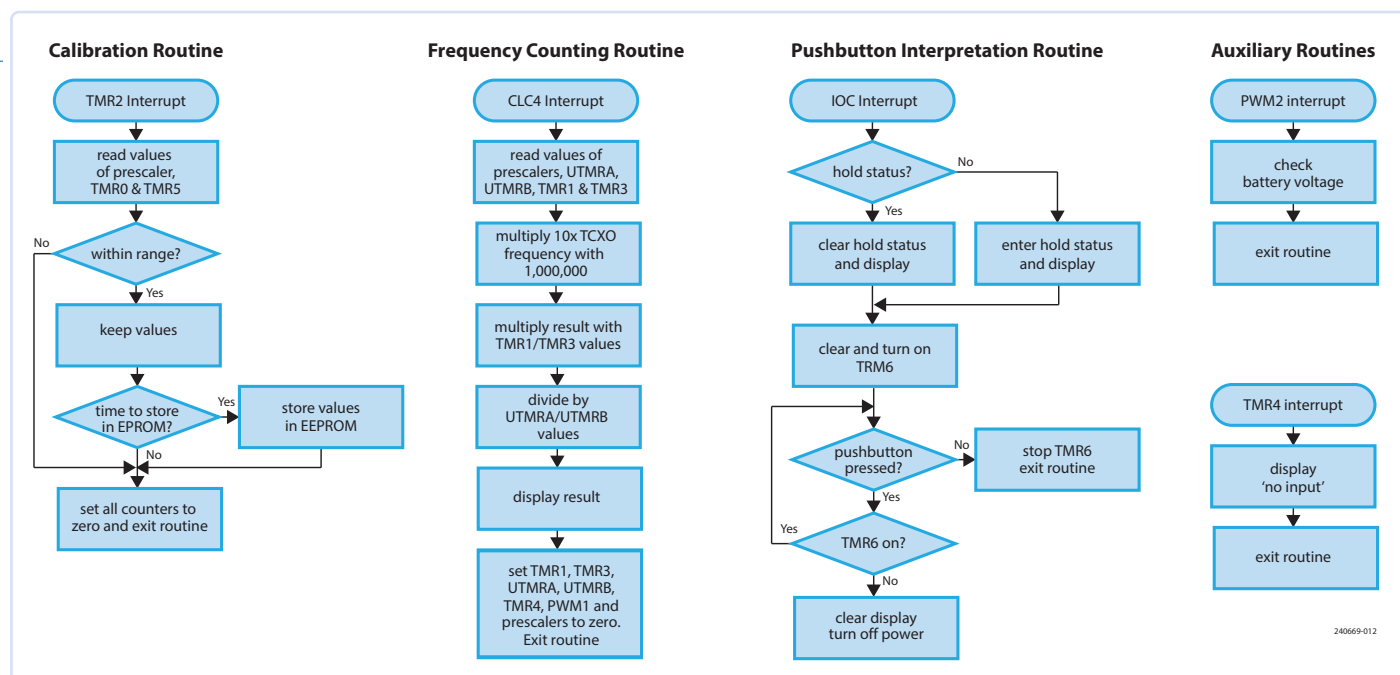


Figure 5: Flowcharts of the five interrupt routines handled by the software.

We need to amplify, or eventually attenuate, the signal. This is all taken care of by a preamplifier. A display is connected to the microcontroller to indicate the measurement results. And, as mentioned above, a TCXO and a GPS receiver are connected as well to provide the reference frequency.

The complete circuit diagram is shown in **Figure 4**. Processor U2 together with the TCXO, GPS receiver U4 and display U3 are the digital parts of the counter.

The preamplifier has an input impedance of 1 MΩ (R1). Junction FET Q1 provides a very high input impedance and some amplification (D1 is an AC voltage clamp protection for the sensitive gate). Emitter follower Q2 is DC-coupled to output amplifier transistor Q3. U1 is a high-speed comparator. It acts as a signal squarer that generates a nicely-shaped pulsed signal at the sampling input of the microcontroller.

This circuit arrangement has sufficient bandwidth for the counter, which has been confirmed by a SPICE simulation. The peak-to-peak voltage of the input signal might be as low as 50 mV for providing a good output signal, which is input to a high-speed comparator. Some hysteresis is provided by feedback resistor R13.

Even if the input voltage at pin 1 of the comparator exceeds its supply voltage, the device will not be damaged, since the input

current will not be higher than allowed. The output signal of the comparator connects to the microcontroller for processing. Because of its sensitive input, the first part of the preamplifier is shielded.

Power Supply

BT1, an 18650-type Li-ion rechargeable battery, provides power to the circuit. The battery voltage is nominally 3.7 V. The GPS receiver needs 3.3 V. Since the latter directly connects to the processor, we run the processor and the TCXO also on 3.3 V. A simple regulator (U5) takes care of this. The input amplifier requires a 5 V supply voltage for proper operation. Boost regulator U6 converts the battery voltage to slightly higher than 5 V.

L2 and C13 will remove the ripple on the output of the converter. The voltage drop on the inductor makes the supply voltage of the preamplifier exactly 5 V. Since the display has its own regulator on-board, it is powered directly from the battery. Operating pushbutton SW1 causes MOSFET Q6 to go on and with it MOSFET Q5, which serves to turn on the power. The processor starts operating and will then turn on MOSFET Q4, keeping the battery connected after the pushbutton is released.

With a fully charged battery, the counter will deliver up to 20 hours of continuous operation. For recharging, a USB power source will be used. The charger IC U7 takes care of this. Pins 3 and 4 of the USB connector each need a resistor connected to ground.

Without these, the so-called fast chargers would fail to deliver power. The red LED is on during charging, and the blue one lights up when the battery is fully charged. Connector J2 serves for programming the processor, whereas jumper J4 may connect the battery permanently, which is useful during debugging and programming.

Environmental conditions for good GPS signal reception may, at times, be insufficient. An optional connector for an external antenna has been provided for such eventualities (J3).

The Firmware

The firmware is written in assembler without using any libraries and can be downloaded here [1]. Programming in assembler results in firmware that is more efficient in using memory resources and is faster than produced in a higher-level language. Therefore, it takes only about 5% of the processor's program memory.

After the mandatory configuration of IOs, CLCs, counters and other internal modules, the main program is entered. This part takes only care of displaying the wave symbol during the time the GPS receiver outputs a pulse and of deleting the update dot (see details on these symbols later in the article). All other functions are taken care of by interrupt service routines, five in total.

The flow charts of these routines are shown in **Figure 5**. These are self-explanatory. Interrupt routines in legacy microcontrollers start with

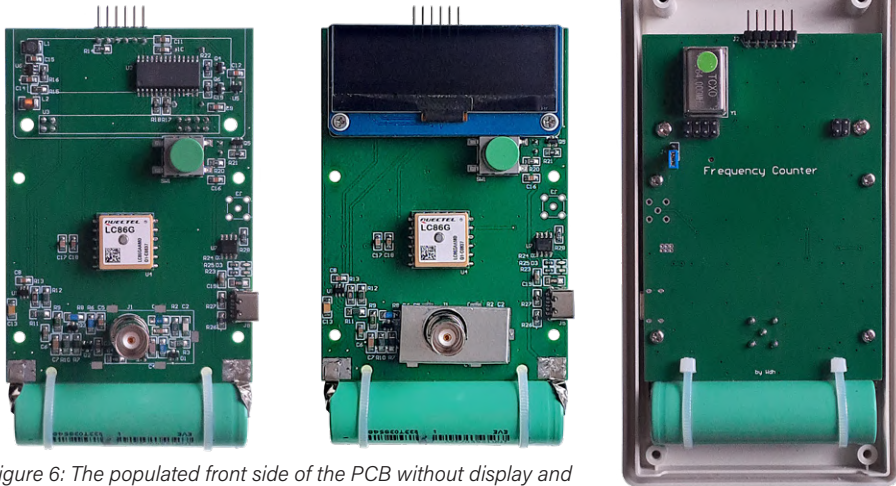


Figure 6: The populated front side of the PCB without display and shield (left), completed (mid) and the rear side of it, mounted in the top of the enclosure (right).

finding out which peripheral caused the interrupt. After finding the cause, a jump to the handling routine is performed. The micro-controller here uses so-called vectored interrupts. This means that each peripheral has its own unique vector that points to its interrupt service routine, so that inside the interrupt routine it is not required to find out which peripheral triggered it. Handling interrupts is very fast in this manner.

Construction

A single PCB of size $100 \times 70 \text{ mm}^2$ hosts the complete circuit, as shown in **Figure 6**. Care has been taken to keep the input amplifier, the processor, the GPS receiver and the power circuits separate from each other.

Except for the connectors, the display and the TCXO all parts are SMD's. This saves space, makes layout easy and results in a compact circuit of small dimensions. The resistors of the input amplifier are of size 1206. These have a lower parasitic capacitance than 0805 size parts which are used for the rest of the circuit.

By leaving a certain area on the board free of copper, the input capacitance of the amplifier could be kept at about 15 pF. The circuit board has been housed in an OKW Datec-Mobil-Box S. Because of the OLED display's height, the following procedure was used to place it. First, the display with 5 mm standoff spacers was mounted.

Then the headers were pushed in from the back side of the board. To make optimal contact, the pins of the header were then pushed in, one by one, as far as possible. The maximum height of the complete board

became so 19 mm, which fitted well in the available 22 mm of the OKW enclosure. Only a few connections to the display module



Component List

Resistors

(1206, unless otherwise noted)
R1 = 1 M Ω
R2 = 100 k Ω
R3, R10 = 22 Ω
R4 = 470 Ω
R5 = 330 Ω
R6, R8 = 22 k Ω
R7 = 390 Ω
R9 = 680 Ω
R11 = 2.2 k Ω , 1%
R12 = 1 k Ω , 1%
R13, R14, R19, R21, R22 = 10 k Ω , 0805
R15 = 510 k Ω , 1%, 0805
R16 = 150 k Ω , 1%, 0805
R17, R18 = 1.8 k Ω , 1%, 0805
R20 = 220 Ω , 0805
R23 = 0.4 Ω
R24, R25 = 1 k Ω , 0603
R26, R27 = 5.1 k Ω , 0805
R28 = 1.2 k Ω , 0805

Capacitors

(1206, unless otherwise noted)
C1 = 220 nF, 0805
C2 = 47 pF
C3, C5, C9, C12, C14...C17, C19 = 10 μ F, 0805
C4 = 100 pF
C6, C8, C11, C18 = 100 nF, 0805
C7 = 150 pF
C10 = 10 nF, 0805
C13 = 220 μ F

Semiconductors

D1 = BAT18-04
D2 = LED, red
D3 = LED, blue
Q1 = MMBFJ309
Q2, Q3 = BC849C
Q4, Q6 = AO3400
Q5 = SI2305
U1 = TLV3501AIDB
U2 = PIC18F24Q71 SO
U3 = 2.23" OLED Waveshare
U4 = LC86G Quectel
U5 = MCP1703-3302CB
U6 = MCP1640B
U7 = TC4056A

Inductors

L1 = 4.7 μ H, 1210
L2 = 1 mH, 1210

Miscellaneous

SW1 = Schurter 1241 SMD switch
Y1 = 64 MHz TXCO
BT1 = Li-ion 18650 cell
2x2 Header
4x2 Header
2x 5 mm, M3 spacer
4x 4 mm M2.7 spacer
Enclosure Datec-Mobil-Box S

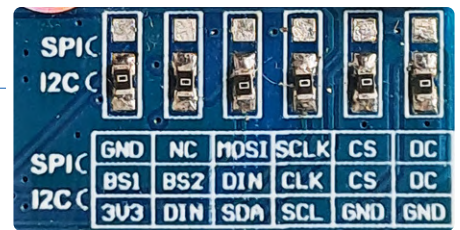


Figure 7: Detail of the display board, showing the correct position of the 0 Ω jumper resistors for selecting the I²C bus.

were required; therefore, only headers were mounted for those needed and some additional ones to hold the module in place.

The Waveshare OLED display is by default configured for being driven by an SPI bus. Here we use the I²C bus instead. Therefore, six 0 Ω resistors had to be moved on the module. **Figure 7** shows how it looks after the modification. Care has to be taken, since these parts are quite small.



About the Author

Willem den Hollander has been passionate about electronics since the age of 12. He has a Master's degree in electronics engineering and worked for 37 years in R&D in the field of consumer electronics. His favorite subjects are power supplies, digital circuits, programmable logic, and microprocessors. Several of his projects have been published in Elektor.



Figure 8: Auxiliary indicators, with display value updated (left), GPS output pulse received (mid), and 10 subsequent successful GPS pulses received (right).

Using the Counter

Pressing pushbutton SW1 turns on the instrument. A press during measuring frequency will hold the displayed value, and the *hold* message is visualized. Pressing again will release the display. The frequency counter will turn off spontaneously in the absence of an input signal for more than 60 s.

Alternatively, the counter may be turned off by pressing the pushbutton longer than 2 s, acknowledged by the display going dark.

Figure 8 shows everything that can be indicated in the display's corner. Each time the display is updated, a little dot will light up in the upper-left part (Figure 8, left). During the time the GPS receiver outputs a pulse, the wave icon in the middle picture of Figure 8 is displayed.

When the processor successfully received a series of ten successive pulses from the GPS receiver, the tick character on the right picture of Figure 8 will be present on the display for one second. **Figure 9** shows all the possible views of the main readout, with (from the top) the regular display, the *low battery* status indication, the frequency *hold* and the *no input* (dashed display) statuses. The battery voltage check is performed about every 30 s; if it is below 3.2 V, the *low battery* message will be displayed, indicating that battery charging is required. ◀

240669-01

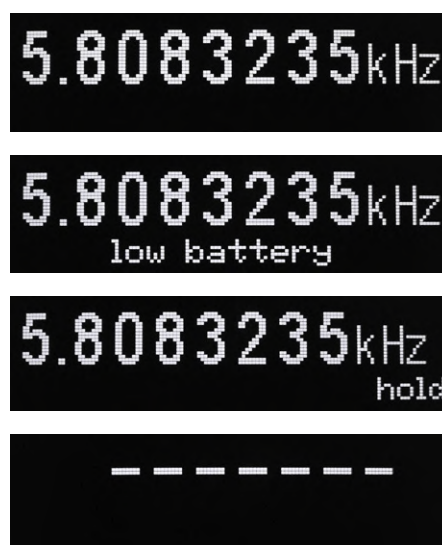


Figure 9: The frequency counter's display with all the possible options, i.e. (from the top) the regular reading, the *low battery* status, the frequency *hold* status, and the *no input* (dashed display) condition.

Questions or Comments?

Do you have technical questions or comments about this article? You may contact the author at wdenhollander@solnet.ch or the Elektor editorial team at editor@elektor.com.



Related Products

- Microchip MPLAB PICKit 5 in-circuit debugger/programmer www.elektor.com/20665
- Tam Hanna, *Microcontroller Basics with PIC* (Elektor, 2020) www.elektor.com/19188



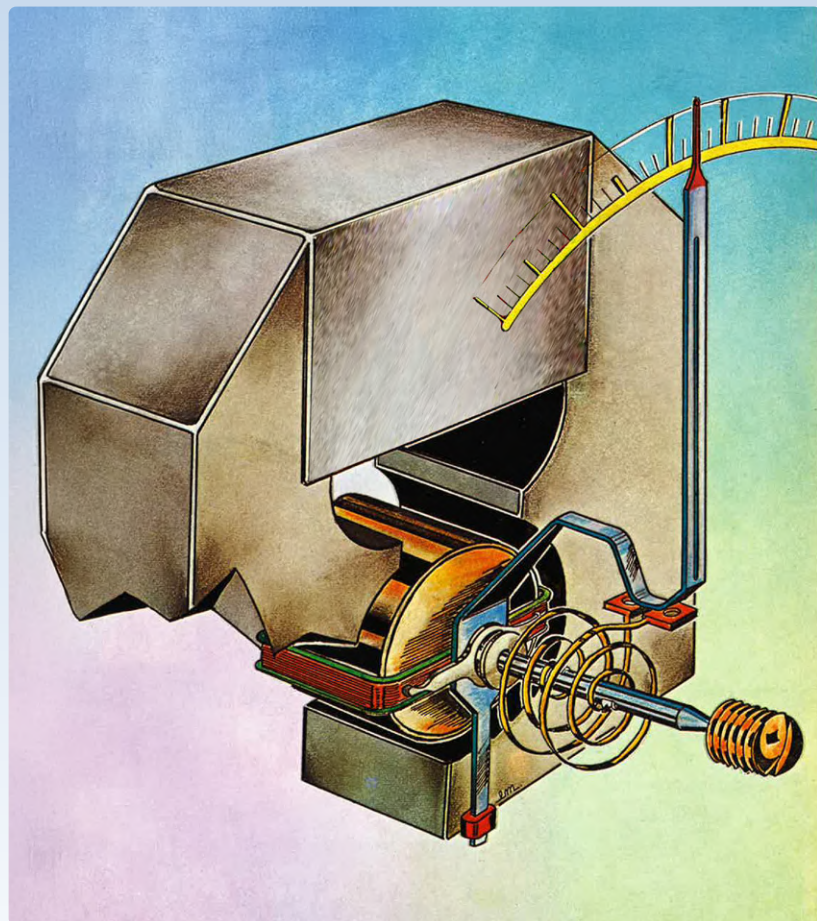
WEB LINK

[1] Firmware download link: <https://elektormagazine.com/labs/elektor-articles-software-downloads>

Analog Meters

By David Ashton (Australia)

Figure 1:
From a glorious Elektor
cover of 1979: cutaway
of the movement of a
moving coil instrument.



Analog meters can be easier to read. A needle a bit more than halfway across a 20-V scale is around 12 V, and if you need more precision, look closer. If you quickly read 1234 on a digital display, is that 1.234 V or 12.34 V? Where's the decimal point? This can be especially confusing on an auto-ranging display. And analog meters don't need a power supply. No fiddling with ± 5 V supplies, sometimes isolated...

Moving Coil and Moving Iron

The main part of an analog meter is called the *Movement*. As this is the bit that actually moves, this is fair enough. A movement consists of a needle, attached to something that will move it across a scale against the force of a tiny spring, to indicate a reading. Magnetism is usually — but not always — involved. The maximum current that a meter can measure is called *Full-Scale Deflection* (FSD).

Moving Coil meters are the most common type; the current to be measured generates a magnetic field in a coil, which moves against the field from a permanent magnet. They can be made very sensitive (50 μ A is not unusual), they are usually very linear, and you can achieve an accuracy of 1% on a good quality meter, with a "mirror scale" to ensure you're reading it exactly head-on.

But they are delicate, the cost is proportional to the quality, and good meters are expensive. The structure of a moving-coil meter is shown in **Figure 1**.

The other common type is the *Moving Iron* meter. An iron vane is attracted by the measured current in a fixed coil. Moving Iron meters are cheap, non-linear and very rugged, and usually not very sensitive; 1 A is a typical minimum. They're great for things like battery chargers or the like, where high precision is not required. **Figure 2** shows a 5 A moving iron meter. Note the non-linear scale at the ends.

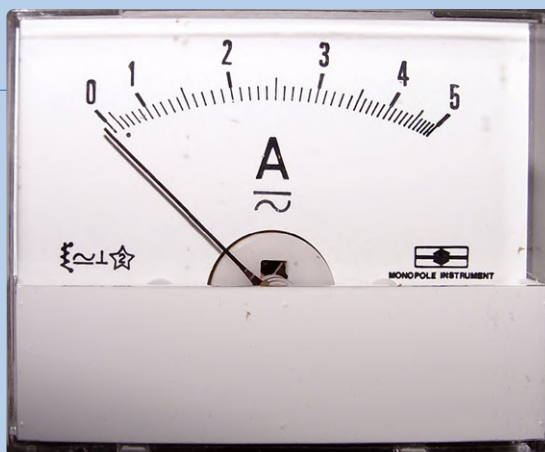


Figure 2:
Moving Iron 5 A meter face. Note the nonlinear scale, particularly at the ends, with the start of reading at 500 mA. (Source: David Ashton)


Shunts and Multipliers

Meters can be adapted to read higher currents than their rated current, or to read voltage. To measure a higher current, a shunt is used — a low-value resistor that shunts or “steals” some of the current in the circuit. To measure voltages, a multiplier resistor is used, i.e. a resistor that limits the current to the FSD value at the voltage to be measured. Calculating shunts and multipliers is a bit of an art, and beyond a short article like this.

Figure 3 shows a *VU meter* (calibrated to read decibels for audio measurement), while **Figure 4** shows an older type of round meter, beloved of retro and steampunk enthusiasts. **Figure 5** shows an edge-reading meter, where the movement is horizontal, and the needle is bent at 90 degrees to save space.



Figure 3: VU meter, calibrated in decibels, this one also has a peak LED. (Source: Wikipedia – Iainf – Creative Commons License, https://it.wikipedia.org/wiki/VU_meter#/media/File:VU_Meter.jpg)

A knowledge of analog meters — and how and when to use them — can simplify your designs, sometimes save you money, and result in some very good-looking equipment! 

250117-01

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

David Ashton was born in London, grew up in Rhodesia (now Zimbabwe), lived and worked in Zimbabwe, and now lives in Australia. He has been interested in electronics since he was “knee-high to a grasshopper.” Rhodesia was not the center of the electronics universe, so adapting, substituting, and scrounging components were skills he acquired early (and still prides himself on). He has run an electronics lab, but has worked mainly in telecommunications.



Figure 4: Old type of round Bakelite, AC/DC 300 V FSD meter from the 1940s. (Source: Roberto Armani)

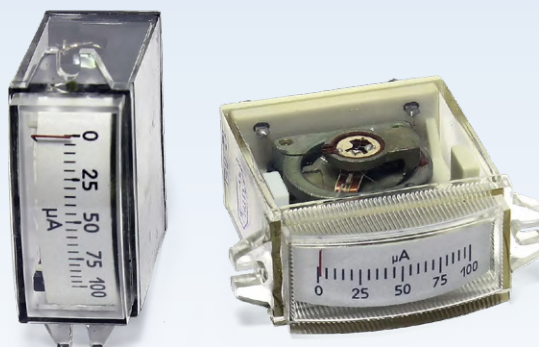


Figure 5:
Edge-reading meter with 100-µA FSD. (Source: Oleg Rozencvaig - eBay - alexer1)

Stand-Alone Crystal Tester

How Accurate Is Your Clock Source?



By Philippe Le Guen (France)

When it comes to searching for, or even sorting, the various quartz crystals at one's disposal, it sometimes turns out that their markings are partially erased or even non-existent. The compact instrument described in this article can't compete with more complex and expensive professional instruments, but it can be useful to check the correct functionality of an unknown quartz, displaying its frequency onto an eight-digit readout.

By this design, I wanted to create a simple yet sufficiently accurate device for my lab, mostly using standard components available in the drawers. There are of course other projects of this kind, but not with an 8-digit frequency display, as my prototype features in **Figure 1**.

Operating Principle

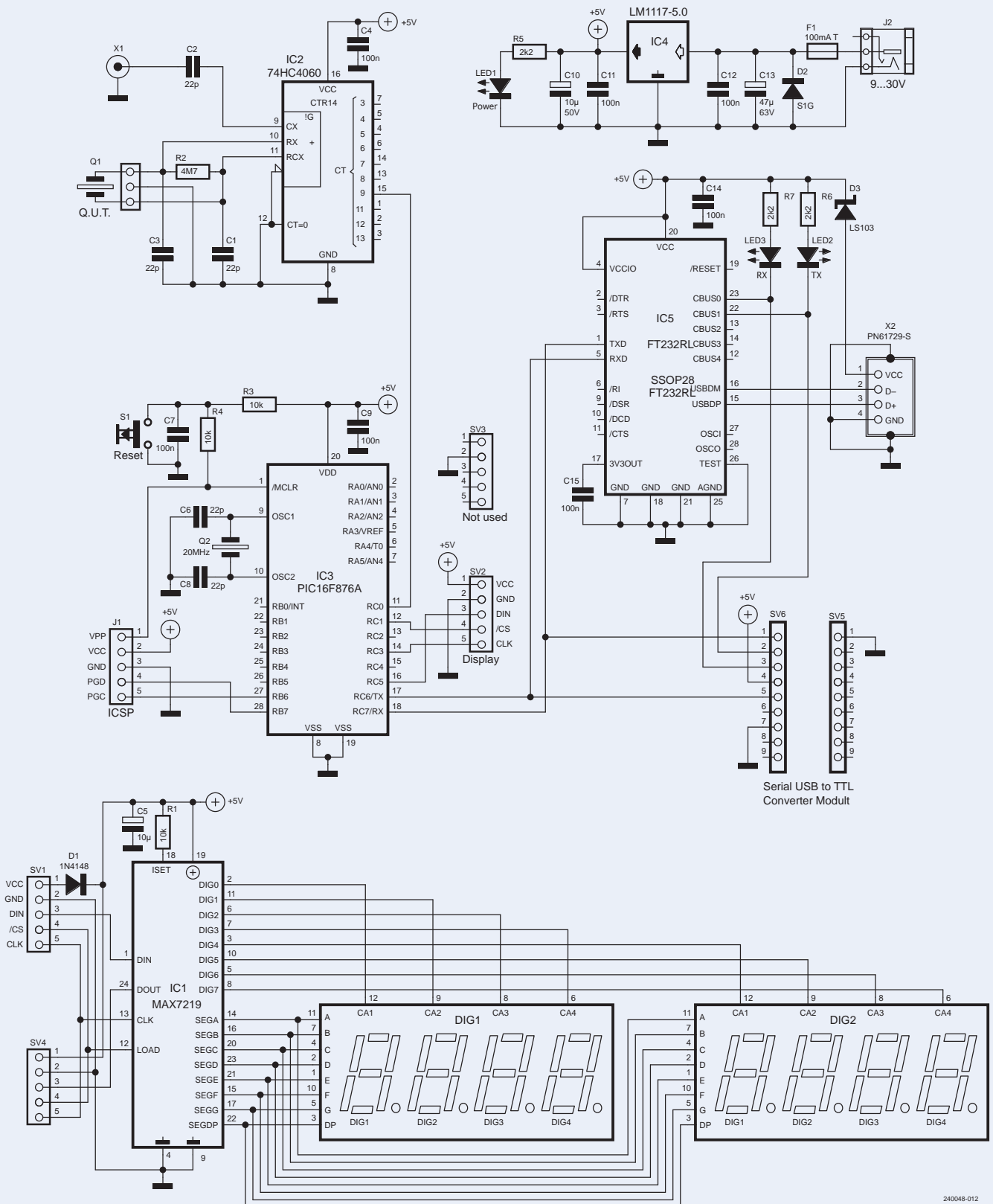
As visible in the schematic of **Figure 2**, all the electronics revolve around a PIC16F876A microcontroller by Microchip [1], which counts the pulses

Figure 1: The eight-digit display prototype of the Crystal Tester at work, while testing a 30 MHz quartz.

from a 14-stage analog counter circuit (74HC4060) [2] after dividing the quartz frequency to be analyzed by 1024. Why these choices?

- I wanted to use salvaged components that were available in my lab.
- I had already used a pre-divider for my stand-alone inductance meter [3], in that case a 12-stage counter HEF4040 from Philips Semiconductors, that worked perfectly.

The 74HC4060 integrates a Pierce oscillator [4] (shown in its typical configuration on **Figure 3**, with the RC network made by R2-C1-C3 and the quartz crystal to be tested). I've partly taken up the idea of Karl-Heinz Lorenz, whose article appeared in *Elektor* December 2002 [5]. In this architecture, R2 tends to keep equal the voltages at both input and output of the inverting gate 1 of IC2; therefore, this gate will operate in the transition region, giving a phase shift of 180° — when toggling — in the loop with Q1, C1 and C3. These three components will then provide an additional 180° shift, closing the loop and satisfying Barkhausen phase-shift main criteria (360°) for oscillating circuits. In this “parallel crystal” configuration, the loop stabilizes at the resonating frequency of the quartz.



240048-012

Figure 2: Schematic diagram of the Crystal Tester.

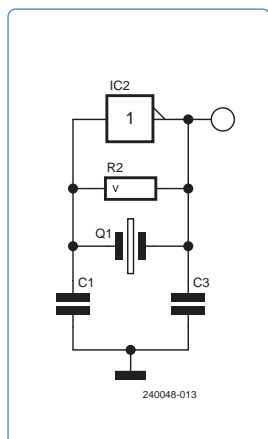


Figure 3: A typical Pierce oscillator. For convenience, the nomenclature of the components is the same as that used in the main schematic diagram.

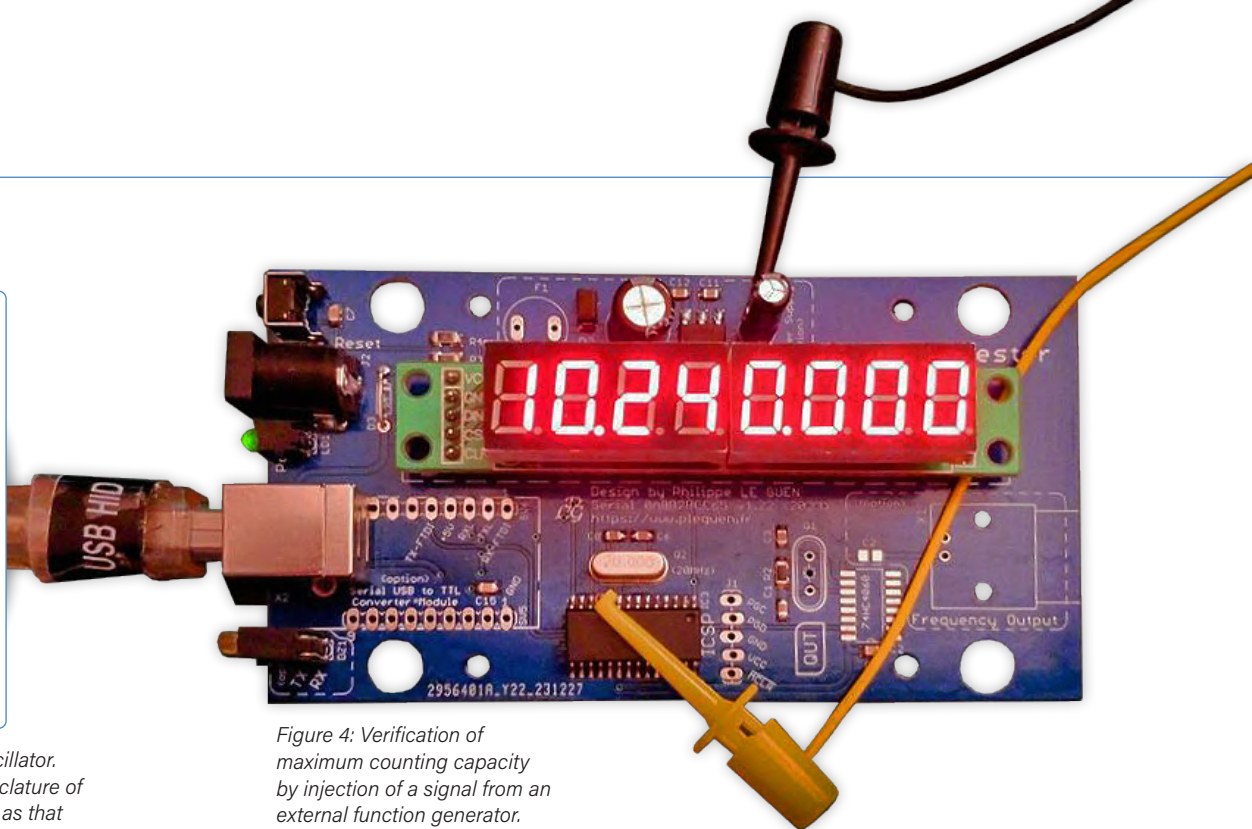


Figure 4: Verification of maximum counting capacity by injection of a signal from an external function generator.

In my design, the oscillator could have been based on a Colpitts architecture as well. In that case, however, I would have needed some more electronics before sending the resulting signal to the microcontroller, which would have complicated the schematic and the PCB routing; therefore, I decided to stay on a Pierce architecture.

The chosen microcontroller already uses its 1:256 pre-divider and cannot count beyond 65,535 in direct measurement (RC0/TICK1). To check quartz crystals up to 30 MHz, it becomes necessary to use an additional divider, and this is achieved with the 74HC4060. Using the counter with its Q9 output, which counts up to 1024, it then becomes theoretically possible to measure frequencies of 67 MHz ($65,535 \times 1024 = 67,107,840$). This works, as verified with my function generator (see Figure 4).

In fact, given the tiny error generated by the system, I was unable to exceed an input frequency of 65.530 kHz (on pin 11 = RC0 of IC3). The display then showed 67,107,840, which divided by 1024 gives 65,535, the maximum that Timer1 can count in the microcontroller's 16-bit mode. Given the multiplication coefficient (1024), the instrument's resolution is 1024 Hz.

Microcontroller

The PIC16F876A microcontroller that I'm using (salvaged from a scrapped machine) is in SMD version, and is clocked by a 20 MHz quartz. The network that includes S1 push-button and R3/R4 resistors forms the reset circuit, with C7 acting as a debouncing capacitor for the switch.

The SPI serial bus uses pins 12, 14 and 16, which reach the display module via SV2 connector, while SV3 just works as a mechanical support. Counting pulses delivered by the 74HC4060 reach the microcontroller via pin 11, while pins 17 (TX) and 18 (RX) are connected to the serial communication lines.

Display

The display section uses a small eight-digit, seven-segment module widely available on the web [6] driven via SPI bus. This is assembled on the PCB using the "double-decker" mounting technique (see Figure 5).

To reduce the overall height of the module, it is soldered directly to the main board. This choice keeps PCB routing to a minimum, and given its very low cost, it would be a shame not to use it.

USB-UART Interface

This is implemented using a specialized circuit, the popular FT232RL by FTDI [7] and its associated components or with a ready-made module, depending on your personal choice. It receives signals from the USB type B connector on pins 15 and 16 (USBDP/USBDM). Two LEDs (LED2 and LED3, marked as DZ1-1 and DZ1-2 on the PCB silkscreen) indicate the data flow on the communication lines.

Power Supply

Power can be supplied via the USB interface mentioned above, or via an external 12 V DC source with an on-board (optional) 5 V regulating IC (IC4). A Schottky diode (D3) is installed between pin 1 (VCC) of the X2 connector and pin 4 and pin 20 of FT232 (IC5) to protect the +5 V line coming from the PC USB in case the 5 V generated by the external power supply is also present.

IC4 is an LM1117-5.0 LDO regulator by Texas Instruments [8] capable of delivering up to 800 mA. There's no need to add a heat sink here, given the low current drawn by the project. A protection diode (D2) is inserted immediately after fuse F1, with filtering provided by capacitors C10...C13.

The board's power consumption is around 40 mA with all eight digits on at +12 V, which is still reasonable (considering the reduction in

brightness...), even during test mode with the entire display on (with multiplexed displays). If you want to set the brightness to its maximum, you'll need to adjust the current rating of fuse F1 accordingly, but with a 100 mA type there's still some margin. (I've tested this on my prototype, of course.)

Connections

An optional BNC connector can be used to display the measured signal on an oscilloscope, or to connect an external frequency meter.

Firmware

I've written the firmware in mikroC language, the code is available for download on the Elektor Labs webpage for this project [9]. Programming can be done with the programmer of your choice (in my case, it's a PICkit4 by Microchip [10]) via the ICSP connector, as shown in **Figure 6**. I've included a bootloader which — thanks to the USB-UART stage — makes it easy to modify and/or update the code without having to open the device or extract the microcontroller.

A micro-switch accessible through the case allows resetting during the programming process. This is optional, however, as it is of course possible to dispense with the bootloader and program the microcontroller directly with the firmware, eliminating the need for the USB-UART interface.

The firmware is extremely resource-efficient:

- Used RAM (bytes): 37 (11%), Free RAM (bytes): 315 (89%)
- Used ROM (program words): 1166 (14%), Free ROM (program words): 7026 (86%)

The principle is basic: the measurement frequency of the DUT associated with the oscillator consisting of IC2 (74HC4060) and the RC network (R2-C1-C3) is divided by 1024, the resulting signal being available on pin 15 (Q9) of IC2. This signal is routed directly to IC3 microcontroller (PIC16F876A) on pin 11 (RC0/TICK1). A pulse counter using Timer1 in 16-bit mode counts these pulses over a period of 1 s, and by multiplying the number of pulses by 1024; we then obtain the frequency of the quartz being analyzed. The result is displayed on the eight-digit, seven-segment display via an SPI serial bus.

Min/max detection is performed for each new measurement, generating an appropriate display:

- Eight dashes "-----" mean that the frequency read is too low, or may indicate that the component to be tested is missing, or even faulty (see **Figure 7**).
- "0F. Error" (Overflow error) means that the frequency read is too high, the maximum being 67.10784e6 or approximately 67 MHz (see **Figure 8**).

Depending on the value of the frequency to be displayed, the dots of the thousands are automatically positioned to facilitate reading.

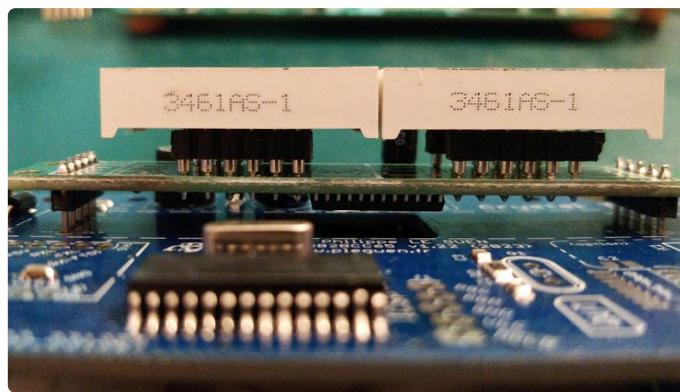


Figure 5: The display board, stacked on top of the main board of the crystal tester.

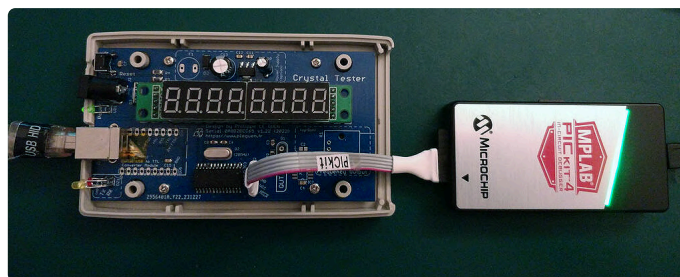


Figure 6: Programming of the PIC microcontroller via the ICSP connector through a PICkit 4 by Microchip.

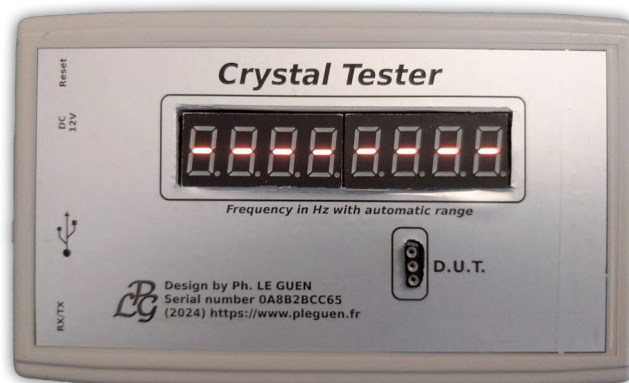


Figure 7: Dashes on the display may indicate that the quartz under test is missing, defective, or oscillating at a frequency that's too low for the measurement.



Figure 8: When the display reports "0F. Error" (Overflow error), the quartz's frequency is too high to being measured correctly.



the Lab, but I preferred to use discrete components, recovered from old boards that I store just for this purpose.

Therefore, the PCB has areas equipped for both modular and SMT assembly. Not all the components on the schematic are absolutely necessary (they are marked as "option"). So you can either leave the USB-UART module area unpopulated and use an external power supply to operate the instrument, or leave the 5 V power regulation area unused and use the USB-UART part with either the components mounted on the PCB, or with a small Arduino-compatible module as a USB-to-UART converter, available just about anywhere on the Web [11]. Always check the compatibility of the module you're going to buy, as some pin-outs may be different from the one I have in the lab.

One of the few difficulties lies in soldering IC5 (FT232RL), as the pitch is minimal (0.65 mm) and you need a temperature-controlled soldering iron with a sufficiently fine tip or, better, a hot-air station. This IC is optional and only useful in the case of need of a USB connection for any software updates. So, if you don't program a Bootloader into the microcontroller, it's not necessary at all.

I've designed a PCB (see both sides in **Figure 9**) specifically for integration into a Multicomp MCRH3135 or Hammond MH3135 enclosure, which I often use in my small devices.

Finishing

To design the front panel, I simply used the excellent GIMP image editor. Laser printing on a matte silver aluminum-style polyester adhesive [12] produces a beautiful silkscreen. ◀

240048-01

Questions or Comments?

Do you have technical questions or comments about this article? You may write to the author at pleguen@gmail.com or contact the editorial team of Elektor at editor@elektor.com

About the Author

Now retired, Philippe Le Guen worked as an Audio-Visual Maintenance Technician, repairing camcorders and video recorders, having started out in the field of audio and Hi-Fi. The measuring equipment in his small personal lab comes mainly from acquisitions made during his work as an engineer. Philippe adds to it or replaces it over time, according to his needs. He works on small designs, mainly in the field of measurement for his lab, and also programs in C language, in entirely self-taught fashion. He also sometimes has to repair certain devices, such as his drinking water softener and many others.



Related Products

- **Microchip MPLAB PICKIT 5 In-Circuit Debugger/Programmer**
www.elektor.com/20665
- **Bert van Dam, PIC Microcontrollers (Elektor E-book, 2021)**
www.elektor.com/18093

WEB LINKS

- [1] Microchip PIC16F876A: <https://www.microchip.com/en-us/product/pic16f876a>
- [2] 74HC4060 Datasheet: <https://tinyurl.com/mu6vw4kj>
- [3] Author's inductance meter project: <https://tinyurl.com/48pakwaa>
- [4] Pierce oscillator on Wikipedia: https://en.wikipedia.org/wiki/Pierce_oscillator
- [5] K.-H. Lorenz, "Crystal Tester," Elektor 12/2002: <https://www.elektormagazine.com/magazine/elektor-200212/17361>
- [6] Display module: <https://tinyurl.com/3t8emufc>
- [7] FTDI FT232RL Datasheet: <https://tinyurl.com/58nbehe4>
- [8] TI LM1117 Datasheet: <https://www.ti.com/lit/ds/symlink/lm1117.pdf>
- [9] Elektor Labs webpage for this project: <https://tinyurl.com/3sejrpc4>
- [10] Microchip PICKIT4 In-Circuit Programmer: <https://tinyurl.com/4ey2z9pm>
- [12] USB-UART module: <https://fr.aliexpress.com/item/1005005956307565.html>
- [13] Polyester adhesive support: <https://tinyurl.com/ycscscbh>

Low-Cost I²C Tester

Connect I²C Devices Directly to Your PC

By Iván Cabrera (México)

The Inter-Integrated Circuit (I²C) protocol is an addressable serial bus that was innovated by Philips in 1982. It's designed for the seamless intercommunication between various integrated circuits (ICs). Today's market boasts a plethora of I²C-compatible devices, including — but not limited to — memory chips, expanders, sensors, real-time clocks, and LCD controllers. Given this diversity, a straightforward testing tool can become an indispensable asset in your electronics laboratory.

While the conventional method of implementation involves connecting I²C devices to a microcontroller, certain scenarios call for a more streamlined approach. For testing purposes, employing a microcontroller can introduce unnecessary complexity. This complexity stems from the need to develop valid firmware, program the MCU, establish a connection to a secondary device (often a PC), and create corresponding testing software. However, during the testing phase, the goal is to simplify this task and to eliminate some variables outside of the testing process.

This article presents an interface that facilitates the direct testing of I²C devices with a PC, thereby obviating the need for an additional microcontroller. The interface leverages an IC to communicate with the I²C device and features a graphical user interface for the visualization of the data acquired. The central element of this setup is a CH341 bridge controller, engineered by Nanjing Qinheng Microelectronics, also recognized as WCH. Retailing at an economical price point of

\$2 to \$3, the CH341 chip is remarkably cost-effective. It performs serial conversions from USB to UART, SPI, parallel (EPP, MEM), and, most notably, I²C. Despite its longstanding presence in the market, the resources for development with the CH341, particularly for the Windows operating system, are somewhat scant.

Installation Process

The initial step involves downloading and installing the appropriate drivers. The file *CH341SER.zip* [1] will only facilitate USB-to-UART conversion, which is not useful for I²C connections. Therefore, it is advisable to get the *CH341PAR.zip* [1] file. Although it may seem counterintuitive that a parallel driver version is required, this package contains the suitable driver and the essential libraries for I²C operations.

The subsequent sections describe the utilization of CH341 in conjunction with C and Python programming languages for the PC software. For C-based software development, the *CH341DLLA64.LIB* library (64-bit) and the *CH341DLL_EN.H* header file are imperative. For Python, *CH341DLLA64.DLL* (64-bit) suffices. My personal GitHub repository [2] hosts all the examples cited herein. Note that the examples are absent the libraries and/or header files, which must be integrated manually.

Fundamentals of I²C

The workings of I²C should be generally well-understood. Beyond the requisite power supply for the chips and a shared ground, communication between I²C devices typically involves two wires: SDA (data) and SCL (clock). To initiate communication, it is essential to configure an address for your device. I²C devices often feature terminals to set the lower bits of the address through connections to VCC or GND. The device's datasheet provides detailed instructions on configuring the I²C address.

Exploring the CH341 Library

CH341DLL_EN.H enumerates the supported library functions. Given the sparse commentary within the file, the most fundamental functions for initiating development are elucidated below. Once acquainted with

these basics, you are encouraged to investigate the remainder of the functions independently. **Table 1** presents the four basic functions. The `i` parameter indexes the device you intend to address, with an index of `0` signifying the first device. Parameter `m` determines the operation mode, with the two least-significant bits being pivotal for I²C (see **Table 2**).

Table 3 itemizes the functions pertinent to data transmission. As per the documentation provided by WCH, these functions can be utilized in the following manner:

Writing to an I²C Device

The `CH341WriteI2C()` function is used to write data to an I²C device. The syntax is as follows:

```
BOOL WINAPI CH341WriteI2C(int iIndex, int iDevice,
int iAddr, int iByte)
```

With the parameters:

- > `iIndex` (ULONG): ID of the CH341 device.
- > `iDevice` (UCHAR): Address of the I²C device (lower 7 bits).
- > `iAddr` (UCHAR): Register address of the I²C device.
- > `iByte` (UCHAR): Data byte to be written to the I²C device.

For example, `CH341WriteI2C(0, 0x40, 7, 0x75)` sends the address of the device indicated by `iDevice` (0x40) via the bus first. Before writing the data byte, this function writes an additional byte, `iAddr`, (in that case 7), and then the data in `iByte` (0x75). This will lead to the data sequence in **Figure 1**.

Reading from an I²C Device

Similarly, `CH341ReadI2C()` reads data from an I²C device:

```
BOOL WINAPI CH341ReadI2C(int iIndex, int iDevice,
int iAddr, int* oByte)
```

With the parameters:

- > `iIndex` (ULONG): ID of the CH341 device.
- > `iDevice` (UCHAR): Address of the I²C device (lower 7 bits).
- > `iAddr` (UCHAR): Register address of the I²C device.
- > `oByte` (UCHAR): Pointer to the data byte read from the I²C device.

For instance, `CH341ReadI2C(0, 0x40, 1, &data)` sends first the address of the device indicated by `iDevice` (0x40) via the bus. Before

Table 1: Basic Functions.

Name	Description
<code>CH341OpenDevice(i)</code>	Open a CH341 device and return the handle. A return of <0 indicates an error.
<code>CH341ResetDevice(i)</code>	Reset a CH341 device. Return is true if succeeded.
<code>CH341SetStream(i, m)</code>	Set the mode of operation, see Table 2.
<code>CH341CloseDevice(i)</code>	Close a CH341 device.

Table 2: Modes.

Mode	I ² C Speed
00	20 kHz
01	100 kHz, default
10	400 kHz
11	750 kHz

Table 3: Data Functions.

Name	Description
<code>CH341WriteI2C()</code>	Write a data byte to I ² C.
<code>CH341ReadI2C()</code>	Read a data byte from I ² C.
<code>CH341StreamI2C()</code>	Perform a write and a read operation over I ² C.

reading, this function writes an additional byte, `iAddr` (1), and then reads the data into the address indicated by `oByte` (&data). This produces the I²C sequence in **Figure 2**.

Streaming I²C Data

The `CH341StreamI2C()` function facilitates both writing and reading operations in a single call:

```
BOOL WINAPI CH341StreamI2C(int iIndex, int iWriteLength, const int* iWriteBuffer, int iReadLength, int* oReadBuffer)
```

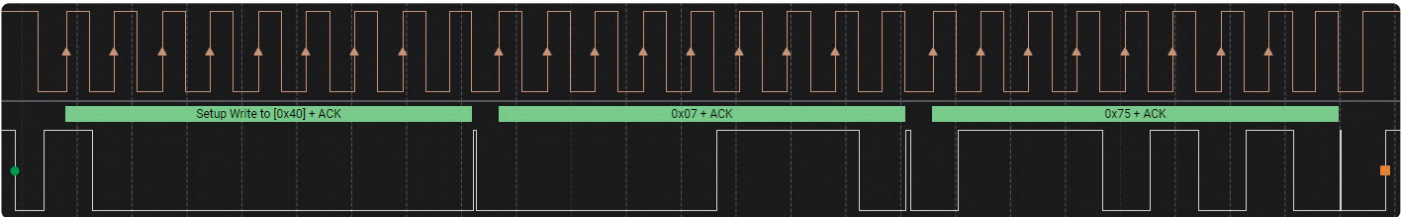


Figure 1: A typical data sequence resulting from `CH341WriteI2C()`.

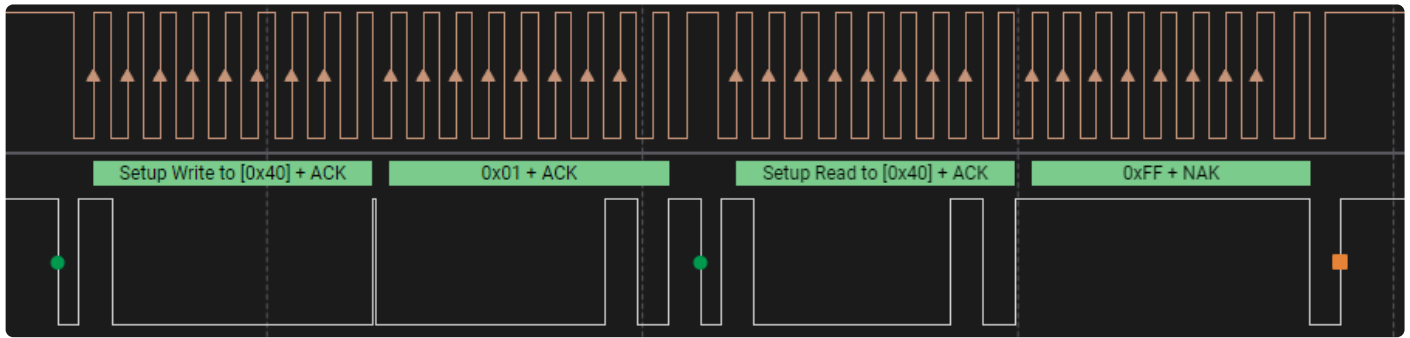


Figure 2: Sequence of `CH341ReadI2C()`.

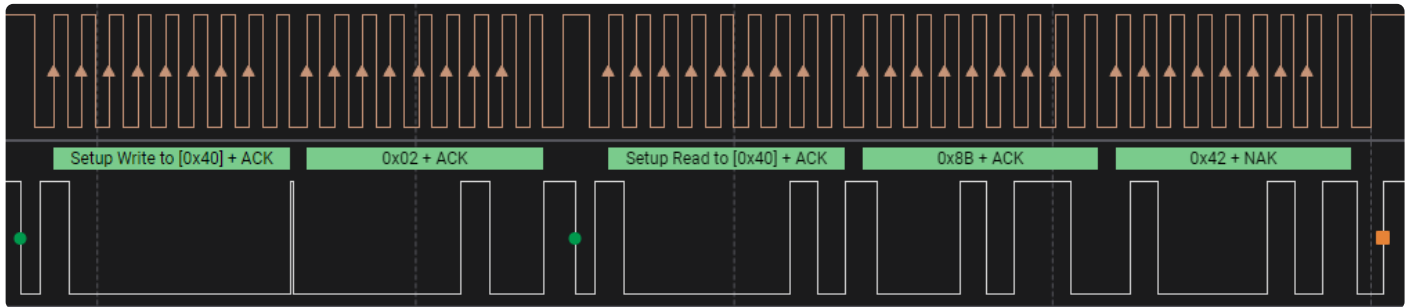


Figure 3: Example of `CH341StreamI2C()`.

With the parameters:

- > `iIndex` (ULONG): ID of the CH341 device.
- > `iWriteLength` (ULONG): Number of bytes to write.
- > `iWriteBuffer` (PVOID): Pointer to data to write buffer, including device address.
- > `iReadLength` (ULONG): Number of bytes to read.
- > `oReadBuffer` (PVOID): Pointer to data to read buffer.

For example, `CH341StreamI2C(0, 2, [0x80, 0x02], 2, &data)` sends the address of the I2C device in the first byte of `iWriteBuffer` (0x80), then writes the number of bytes indicated by `iWriteLength` (in that case 2), and then reads the number of bytes in `iReadLength` (again 2) to the address indicated by `oReadBuffer` (&data). This generates the I2C sequence in **Figure 3**. The `CH341StreamI2C()` function will do the job most of the time but depending on your I2C device. However, `CH341WriteI2C()` and/or `CH341ReadI2C()` can be useful too.

First C Program

My initial C program is an I2C scanner designed to identify the address of a connected I2C device. Any C compiler can be used for development. The example is made with the *Code::Blocks* IDE [3] but you can use what you prefer.

The setup requires specifying the path to `CH341DLL_EN.H` and informing the compiler of the library's location for linking during compilation (see **Figure 4**).

The scanner operates by attempting to read a byte from each possible I2C address. A successful read, indicated by an acknowledgment (ACK), confirms the presence of a device at that address. The complete code for the I2C scanner in **Listing 1** is provided as a starting point for your own C development.

Accelerating Development with Python

Having mastered the installation and basic C programming, it's time to expedite development, particularly for testing purposes. Python emerges as the ideal choice for crafting a graphical interface, such as displaying outputs from a current sensor. Let's explore the use of the INA219 sensor by Texas Instruments [4] as an example.

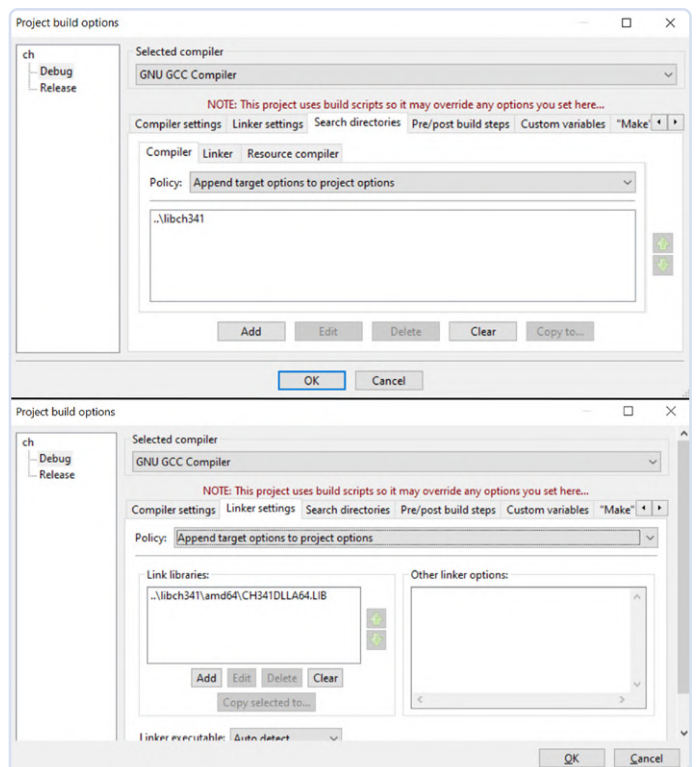


Figure 4: Project build options.



Listing 1: I²C Scanner implementation in C.

```
int i2c_scanner(void)
{
    int address, address_i2c;
    int nDevices = 0;
    byte obyte;
    printf("CH341 I2C Scanner\n");
    HANDLE result = CH341OpenDevice(0);
    if (!FAILED(result))
    {
        printf("Scanning...\n");
        for(address = 1; address < 256; address++ )
        {
            address_i2c = address >> 1;
            CH341ReadI2C (0, address_i2c, 0, &obyte);
            if (obyte != 0xFF && address % 2 == 0)
            {
                printf("I2C device found at address 0x%02X 0x%02X\n", address, address_i2c);
                nDevices++;
            }
        }
    }
    CH341CloseDevice(0);
    return nDevices;
}
```



Figure 6: Physical implementation of Figure 5.

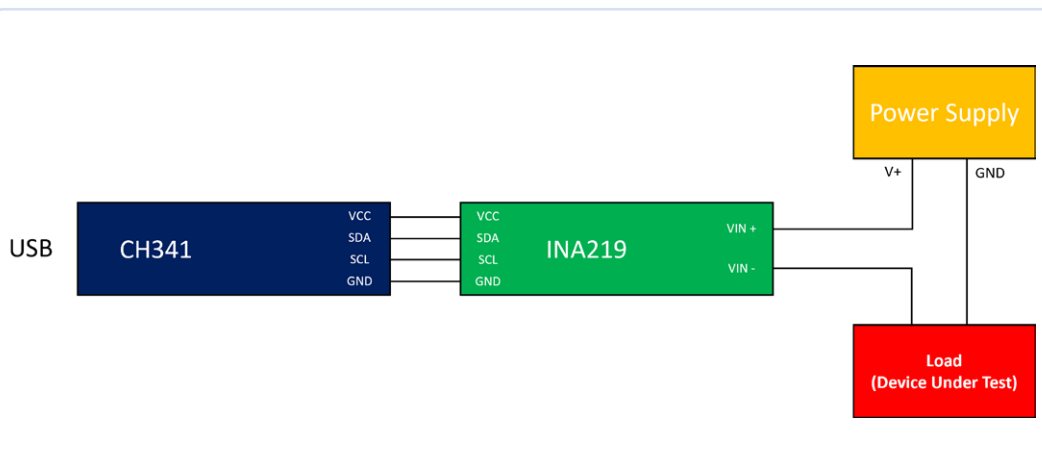


Figure 5: Wiring diagram of a CH341 connected to an INA219.

The INA219 chip is a comprehensive current monitor equipped with an I²C interface, delivering real-time current, voltage, and power readings. It operates within a 3.0 to 5.5 V range and has a current draw of approximately 1 mA, making it feasible to power directly via USB. The wiring diagram and a photograph of the setup are depicted in **Figures 5** and **6**, respectively.

The primary aim is to read bus voltage, current, and power metrics from the INA219 and visualize them in a graphical user interface. This process underscores the simplicity of extracting data from I²C devices.

The underlying code is adaptable for other I²C devices by crafting a dedicated class file encapsulating the requisite logic.

In Python, integrating the DLL is straightforward with a command such as `cdll.LoadLibrary("CH341DLLA64.DLL")`. Ensure that the DLL is accessible to the Python interpreter. Function names and arguments remain consistent with their C counterparts. The most challenging aspect might involve formatting arguments to comply with *ctypes* requirements, but the provided examples offer sufficient guidance.



Listing 2: Example of reading implementation using Python.

```
1 def get_power(self):
2     data_power = [c_uint8(self.i2c_address), c_uint8(INA219_REG_POWER)]
3     output = [c_uint8(0), c_uint8(0)]
4     Uint8Array = ctypes.c_uint8 * len(data_power)
5     parameter_array = Uint8Array(*data_power)
6     power_output = ctypes.c_uint8 * 2
7     output = power_output(*output)
8     self.ch341.CH341StreamI2C(0, len(data_power), parameter_array, len(output), output)
9     return self.convert_bytes2int(output) * self.power_multiplier
```

Table 4: INA219 Functions

Name	Description
<code>get_bus_voltage()</code>	Get the bus voltage from INA219.
<code>get_current()</code>	Get the current from INA219.
<code>get_power()</code>	Get the power from INA219.

The application is bifurcated into two scripts: *sensor_graph.py* and *INA219.py*. The former encompasses all necessary code for displaying sensor data, while the latter is tasked with sensor configuration, initialization, and data retrieval. Refer to the calibration and configuration functions for detailed insights. **Table 4** lists three simple functions to acquire data. An example of power value readings is presented in **Listing 2**.

To do that, follow this recipe:

- Build a list with these two elements: `self.i2c_address` and the index of the register to read `INA219_REG_POWER` (line 2).
- Create the array `data_power` of the length of the request (line 4).
- Expand the array with the content of `data_power` and save it as a new variable, `parameter_array` (line 5).
- Prepare a data buffer to read the output (lines 6 and 7).
- Call the `CH341StreamI2C` DLL function with the prepared arguments (line 8).
- Convert the result and return (line 9).

By substituting *INA219.py* with a class file tailored to your specific I²C device, you can seamlessly transition between different sensors and the corresponding data visualization. **Figure 7** illustrates the GUI showcasing voltage, current, and power plots. Additionally, the GUI features combo boxes that allow for the adjustment of various INA219 parameters. User inputs are directed to the configuration and calibration functions, which in turn modify the INA219 settings.

The graphical interface is constructed using *PyQt5* [5] coupled with *pyqtgraph*'s *PlotWidget*. Adjusting a setting via the combo boxes triggers a callback to a function that recalculates the configuration and

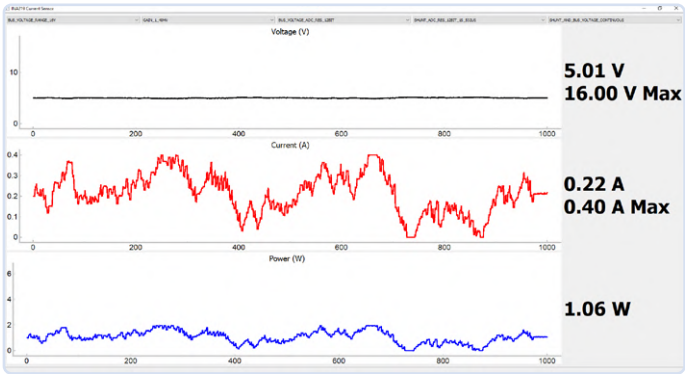


Figure 7: Graphical interface of sensor readings.

calibration, followed by re-initialization of the CH341. A periodic timer invokes the `process_samples()` function, which reads the voltage, current, and power values from the INA219 through the CH341. These readings are then updated in a list and displayed on a label within the main window. The `update_graph()` function is responsible for purging outdated data and presenting the most recent information. ◀

230364-01



About the Author

Iván Cabrera is an Electronic Systems Engineer with a Master's degree in Computer Science. He develops firmware for a manufacturer of electrical devices. He likes to explore new technologies and devices, automation of boring things, and writing development tools. Embedded systems and communications are his favorite topics.

Questions or Comments?

If you have any technical questions, you can contact the Elektor editorial team by email at editor@elektor.com.



Related Products

- > **ScanaQuad SQ200 Logic Analyzer & Signal Generator**
www.elektor.com/18103
- > **LabNation SmartScope USB Oscilloscope**
www.elektor.com/17169



WEB LINKS

- [1] Files "CH341SER.zip" and "CH341PAR.zip" from WCH: <https://wch-ic.com/search?q=CH341&t=downloads>
- [2] Github repository of Iván Cabrera: https://github.com/icabrera/sensor_graph
- [3] Code::Blocks: <https://codeblocks.org>
- [4] INA219 datasheet: <https://ti.com/lit/ds/symlink/ina219.pdf>
- [5] PyQt5: <https://pythonguis.com/pyqt5-tutorial>

Join the Elektor Community

- ✓ 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 5,000 Gerber files



Take out a
membership!



www.elektormagazine.com/member





Figure 1: A beautifully situated tiny house...
(Source: Adobe Stock / Fokke Baarssen)

From Life's Experience

Who Doesn't Honor the Small Things?

By Ilse Joostens (Belgium)

Dreaming of a low-maintenance, off-grid tiny house?

Legal barriers in your location can make it tough. This article explores the challenges, energy solutions, and how electronics can power self-sufficient living.

Can tiny homes be the future, or will red tape keep them out?

Having a house is a hassle, and nowadays, there is also more and more strange interference from the government, which — of course and as always — has our best interests at heart. I, too, have not been spared the many annoyances that house owners have to deal with, such as (endless) maintenance work. I am getting on in years now — in a few years, I may even officially be allowed on

the Seniors Network [1] — and would finally like some more peace and quiet in my life. I also loathe cleaning, scrubbing, dusting, and other household chores. After a few days, you won't see anything of it anymore, and you can start over again. I have actually been dreaming of a tiny house for a while now, with much less maintenance and lower energy costs — in short, more freedom.

Apartmentisation

At the beginning of 2023, I was in the neighborhood to view a tiny house that was temporarily set up in the car park of a home furnishings store as a demo home. I even had contact with the people behind the company that builds these tiny houses. Unfortunately, despite the fact that there is more interest in alternative forms of housing these days, it didn't work out. The situation is as follows: in Flanders, a minimum living area of 18 m² is required for an independent home, with a minimum ceiling height of 2.20 m. Local authorities often extend this minimum living area considerably, which, unfortunately, means that it is not always possible to live in a tiny house as a primary residence [2]. In practice, placing a tiny house in the garden of an existing home is only permitted in the context of a so-called care home. Fortunately, I am not yet in such need of care that I would qualify for this. It continues to bother me, and maybe I should pretend to be disabled like Philippe Abrams in the French film "Welcome to the Sticks" [3] or

move to the French-speaking part of the country, Wallonia, where a legal framework for alternative housing already exists, as evidenced by the brochure “L’habitation légère en Wallonie” [4] that highlights the specific regulations.

I don’t quite understand it. Governments want to limit CO₂ emissions as much as possible with all kinds of Draconian measures, but tiny houses that use much less energy and help us to downsize are not allowed. Tiny houses surrounded by greenery could also be a nice alternative to the ever-increasing apartmentisation and densification of residential areas (**Figure 1**). Personally, I can’t see myself living in one of those 15-storey concrete consumer boxes. I’d much rather have a tiny house, and the nice thing is that I can do something with electronics again.

To Grid or Not to Grid

The living space of a tiny house is much smaller, which means less energy is needed for heating and lighting. Such a mini-home is also easy to insulate and ventilate, especially compared to older existing homes. The fact that you can build such a house yourself and that the low energy consumption also allows you to go off-grid gives the enthusiastic do-it-yourselfer plenty of opportunity to experiment to their heart’s content.

The most obvious energy source for an off-grid electricity supply is a combination of solar panels and battery storage [5].



Figure 2: The sun provides warm water.
(Source: Adobe Stock / Michal)

To ensure you have enough electricity available in winter, it is important to oversize the number of solar panels and the battery capacity. The excess energy in summer can then be used to heat sanitary water in a boiler, whether or not in combination with a heat pipe solar collector (**Figure 2**). In theory, heating with a heat pump is possible in winter and there are small mini-split systems for colder climates with an inverter-controlled compressor on the market. In practice, however, this is a bit of a balancing act with the generally drizzly climate here in the lower countries, even in summer. It is for good reason that the “Drache Nationale” [6] is a household name in Belgium, referring to the heavy rainfall that sometimes falls on the national holiday of 21 July. In winter, when the ground freezes solid, a small pellet stove might be a safer option, as you can also use it to heat the sanitary water in the same boiler.

Direct or Alternating Voltage?

I have no intention of rehashing the feud between Thomas Edison and George Westinghouse, from around 1890, known as “The war of the currents.” With the large number of HVDC connections currently in use worldwide, Thomas Edison

did eventually win a small victory. It is a no-brainer to provide sockets with alternating current in a tiny house, just like in a regular house, but if you go off-grid you are already dealing with batteries, and it may be a smart idea to provide an additional 12 V or 24 V grid [7]. Many appliances such as LED spotlights and strips, smartphones, and laptops operate on their own on a very low DC voltage, which means that mains adapters can be dispensed with, and efficiency increases. The copper losses are not too bad because in a tiny house, there are simply no dozens of meters of cable running. A ring circuit [8], as is also used in the United Kingdom for mains voltage, with branches for consumers, will get you a long way and limit copper losses even more.

Very low voltage networks and devices are also easy to combine with popular boards, from Arduino, Raspberry Pi, and ESP32 to even industrial IoT gateways. With open-source software such as Node-RED, Domoticz, Home Assistant, and other packages, the possibilities for experimentation are endless. If the Flemish government would now be willing to cooperate a little... ◀

Translated by Hans Adams — 250136-01

WEB LINKS

- [1] SeniorenNet.be, the website for the active over-57s in Flanders [Dutch]: <https://www.seniorennet.be/>
- [2] Jubel – Eva Sterkens: The legal regime of tiny houses in Flanders [Dutch]: <https://www.jubel.be/het-wettelijk-regime-van-tiny-houses-in-vlaanderen/>
- [3] Wikipedia: Welcome to the Sticks: https://en.wikipedia.org/wiki/Welcome_to_the_Sticks
- [4] Housing in Wallonia: Light Housing in Wallonia [French]: <https://logement.wallonie.be/fr/actualite/habitation-legere-en-wallonie>
- [5] YouTube – Victron Energy: Off Grid Tiny House deep inside a Belgian forest: <https://www.youtube.com/watch?v=LtNRxxGiLwo>
- [6] VRT nws: The myth of the “drache nationale” on 21 July: what has actually happened? [Dutch]: <https://www.vrt.be/vrtnws/nl/kijk/2024/07/19/drache-nationale-regen-21-juli-jacotte-brokken-nationale-feestda/>
- [7] Strohplatz: House on 12 Volt DC [Dutch]: <https://www.strohplatz.com/zelfvoorzienend-off-grid/huis-op-12-volt>
- [8] Wikipedia: Ring circuit: https://en.wikipedia.org/wiki/Ring_circuit



2025: An AI Odyssey

The Transformative Impact on Software Development

By Brian Tristam Williams (Elektor)

As AI begins writing more of the world's software, developers find themselves shifting roles — from coders to collaborators.

With techniques such as “vibe coding” and AI-driven embedded systems on the rise, the future of software engineering is being rewritten in real time.

As we navigate through 2025, the landscape of software development is undergoing a seismic shift, primarily driven by advancements in artificial intelligence (AI). The integration of AI into coding practices is not merely an enhancement but a fundamental transformation that redefines roles, processes, and the very essence of software engineering.

The Rise of AI in Code Generation

In recent months, AI's capability to generate code has reached unprecedented levels. Dario Amodei, CEO of Anthropic, predicts that within three to six months, AI will be responsible for writing 90% of software

code, potentially leading to AI generating all code within a year [1]. This projection underscores a rapid evolution where AI transitions from an assistive tool to a primary coder.

Supporting this trajectory, OpenAI's Chief Product Officer, Kevin Weil, suggests that AI could surpass human coding skills by the end of 2025, reshaping the software development landscape [2]. Such advancements indicate a future where AI not only supports, but potentially leads, in software creation.

The trend is not isolated to trendy startups or AI research companies. Major corporations such as Google, Microsoft, and Amazon have been heavily investing in AI-driven development environments that integrate AI-generated code into their software production pipelines. With AI's growing ability to write, debug, and optimize code, software engineers are shifting from traditional programming to overseeing and guiding AI models.

“Vibe Coding:” A New Paradigm

The concept of “vibe coding,” introduced by AI expert Andrej Karpathy in February 2025, epitomizes this shift. Vibe coding allows individuals to describe desired functionalities in natural language, with AI models generating the corresponding code. This approach democratizes coding, enabling even those

without formal programming training to develop software. However, it raises concerns about code quality, security, and the depth of understanding required to manage complex systems [3].

Benefits of Vibe Coding

- **Accelerated Development:** AI-generated code speeds up software development, allowing smaller teams to accomplish more with fewer resources.
- **Lowered Entry Barriers:** Enables non-programmers to create functional applications.
- **Enhanced Creativity:** Developers can focus on innovation rather than routine coding tasks.
- **Prototyping Efficiency:** AI can rapidly generate multiple iterations of code, allowing engineers to test different approaches quickly.
- **Faster Debugging and Optimization:** AI tools can automatically identify inefficiencies and refactor code on the fly.

Risks of Vibe Coding

- **Code Quality and Maintainability:** AI-generated code may lack structure, leading to unmaintainable codebases.
- **Security Vulnerabilities:** AI can introduce exploitable flaws into software.
- **Lack of Deep Understanding:** Over-reliance on AI may result in developers struggling to comprehend complex code.
- **Ethical Concerns:** AI models trained on public datasets may inadvertently generate code that violates licensing agreements or introduces bias.
- **Inconsistent Code Styles:** AI-generated code may lack uniformity, making long-term maintenance difficult.

While some companies have embraced vibe coding, others remain skeptical. Platforms such as Reddit feature heated debates on the practicality and reliability of this approach, with some developers outright refusing to adopt it [4]. Companies handling mission-critical or security-sensitive software tend to be the most resistant, preferring traditional coding methods that offer greater transparency and accountability.

Always ready with amusing analogies, Reddit is a kindle with commentary from both sides. One Reddit's take is:

"Vibe coding is like buying a kit to build a race car, paying your drunk uncle who 'knows a thing or two about racing' to build the kit for you, then telling all your friends that you built it." [5]

AI at Embedded World 2025

In March, Elektor was at embedded world 2025 in Nuremberg, Germany, and AI-driven code generation and AI in embedded systems emerged as dominant trends. AI is now making its way into low-power, resource-constrained devices, optimizing performance without human intervention.

AI Code Generation in Embedded Systems

AI-assisted code generation is helping engineers develop firmware faster, reducing debugging time, and improving overall efficiency. Numerous exhibitors showcased AI-powered development tools that automate coding processes, making embedded development more accessible [8].

Companies such as Arm and Connect Tech demonstrated how AI models could write and optimize code for embedded processors, ensuring maximum efficiency while minimizing power consumption. The impact on IoT (Internet of Things) devices, automotive software, and industrial automation is profound, as these sectors rely heavily on high-efficiency embedded systems.

Edge AI: Intelligence at the Source

Edge AI, which involves deploying AI capabilities directly on devices, reduces latency, enhances privacy, and enables real-time decision-making. This technology is particularly valuable for industrial automation, healthcare, and consumer electronics, where real-time data processing is crucial [9].

Also at embedded world 2025, multiple vendors showcased AI-powered microcontrollers capable of running complex machine learning models on tiny, energy-efficient chips. This represents a shift from cloud-based AI processing to localized decision-making, reducing the need for continuous cloud connectivity.



AI Tools Enhancing Developer Productivity

The proliferation of AI coding assistants has been instrumental in boosting developer productivity. Tools (Figure 1) such as GitHub Copilot, the Tabnine AI code assistant, and Amazon CodeWhisperer offer real-time code suggestions, automate repetitive tasks, and assist in debugging, allowing developers to focus on more strategic aspects of software development [6].



Watch Elektor
Lab Talk #31:
Live from Embedded
World 2025 on video!





Figure 1:
Tools such as Amazon CodeWhisperer, GitHub Copilot, and Tabnine speed up programmer drudge work.
Source: Prima91, Photo For Everything, Koshiro K @ Adobe Stock

Focusing on strategy rather than tactics has led some to report rapid upticks in efficiency: JPMorgan Chase reported a 20% increase in software engineer productivity after implementing an AI coding assistant. This enhancement enables engineers to allocate more time to high-value projects, particularly in AI and data domains [7].

The Future of Software Engineering

As AI continues to permeate software development, the role of software engineers is evolving. The focus is shifting from routine coding to tasks that require human judgment, creativity, and ethical considerations. Developers are now expected to collaborate with AI tools, guiding them to achieve desired outcomes and ensuring the reliability and security of AI-generated code.

This transition necessitates a reevaluation of educational curricula and professional training programs to equip future software engineers with the skills required to thrive in an AI-augmented environment. Emphasis on critical thinking, adaptability, and continuous learning will be paramount.

We Still Have Questions

The integration of AI into software development is not a distant prospect, but a present reality that is rapidly transforming the industry. While challenges such as job displacement, ethical considerations, and economic uncertainties persist, the potential benefits of AI-enhanced productivity, innovation, and the democratization of coding are substantial. As we progress through 2025, embracing this AI odyssey with a balanced perspective will be crucial for harnessing its full potential while mitigating associated risks. ◀

230181-M-01

Questions or Comments?

If you have questions or comments, brian.williams@elektor.com is my email address. You can also catch me on Elektor Engineering Insights each month on YouTube, and you can find me @briantw on X.



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 AI and single-board computers such as Raspberry Pi.



Related Products

- **D. Situnayake & J. Plunkett: *AI at the Edge* (O'Reilly Media, 2023)**
www.elektor.com/20465
- **G. Spanner: *Machine Learning with Python for PC, Raspberry Pi, and Maixduino* (E-book, Elektor Digital, 2022)**
www.elektor.com/20150

WEB LINKS

- [1] Anthropic CEO: AI to Write 90% of Code within 6 Months, businessinsider.com: <https://tinyurl.com/anthropicbi>
- [2] OpenAI: AI to Surpass Human Coding Skills in 2025, opentools.ai: <https://tinyurl.com/openaicodeing>
- [3] Vibe Coding, Wikipedia: https://en.wikipedia.org/wiki/Vibe_coding
- [4] Why "Vibe Coding" Makes Me Want to Throw Up?, reddit.com: <https://tinyurl.com/redditvibecoding>
- [5] Reddit vibe coding discussion, reddit.com: <https://tinyurl.com/redditvibeuncle>
- [6] 15 Best AI Coding Assistant Tools in 2025, Qodo: <https://qodo.ai/blog/best-ai-coding-assistant-tools>
- [7] JPMorgan: AI coding assistant makes software engineers 20% more efficient, New York Post: <https://tinyurl.com/nypostjpmorgansw>
- [8] Embedded's AI Predictions for 2025: <https://embedded.com/ai-predictions-for-2025>
- [9] embedded world 2025: Trains, Planes, Automobiles...and Edge AI, EE|Times: <https://tinyurl.com/eetimesew25>

Err-electronics

Corrections, Updates, and Readers' Letters

Compiled by Jean-François Simon (Elektor)



DIY Solar Energy Storage

Elektor 1-2/2025, p. 6 (220233)

Although I don't actually need a solar storage system, I'm still very interested in this topic. I found some related YouTube videos that I thought were fascinating and that might be of interest to other readers.

Claus Offermanns, on his channel, *Offys Workshop*, has described the build of a 14.4 kWh battery, among many other things, in a two-part video. The first part is here: <https://youtube.com/watch?v=RzKX-TqFj30>. Of course, there are many other videos on the subject, but here Claus is building at the highest level of craftsmanship. In one of his latest videos, he also demonstrates how hot the connection cables can get if they are undersized. That might deter those who think they can just throw something like this together, and make them more aware of the risks. One simply has to work carefully.

Frank Klee (Germany)

Thank you for the suggestions and the links. I appreciate your concerns. Indeed, readers tackling a DIY solar battery should know how to calculate cable cross-sections based on current without needing another explanation of Ohm's law. In the article, I didn't provide a step-by-step guide but rather presented an example. Anyone attempting such a project needs to do extensive research online and will inevitably come across the issue of adequate cable sizing more than once.

Thomas Scherer (article author)



New Precise Nixie Clock

Elektor 5-6/2016, p. 56 (150189)

We have been informed by a number of our readers that the clock stopped working and started showing zeroes at the beginning of January 2025. This is due to a software issue concerning automatic Daylight Saving Time (DST) in the original firmware. In the code, the DST database ended in 2025. Several fixes have been proposed in the comments of the corresponding Elektor Labs project (www.elektormagazine.com/labs/150189-6-digit-nixie-clock). To hopefully solve this issue for once and for all, I wrote new software which doesn't try to be clever. DST (+1 hour) is not automatic; you can activate or deactivate it by pressing S1. Furthermore, the code is open-source, plain C without funky libraries, so it should be easy to correct and adapt. It is available in the download section on the Elektor Labs project page, under the *Software* tab. If needed, this code should also be easy to adapt for the 4-digit version of this clock.

Clemens Valens (Elektor)



Err-electronics

Elektor 1-2/2025, p. 110 (240681)

Just a quick comment about the issue raised by Hans-Joachim Petzky in the January issue, about a combined project in KiCad with multiple different schematics. He mentions that all components appear on the PCB at the same time. However, there is a way to prevent this. If you only want a specific subcircuit on the PCB, go to the schematics that you don't want on the board, select all components there, right-click them, and then, under *Attributes*, choose to exclude them from the PCB. This way, only the desired components will remain.

Dick Goossens (The Netherlands)

Thank you very much for your comment, we have forwarded it. I'm sure it can be useful to other readers too!

Jean-François Simon (Elektor)



Crescendo Millennium Edition

Elektor 4/2001, p. 36 (010001)

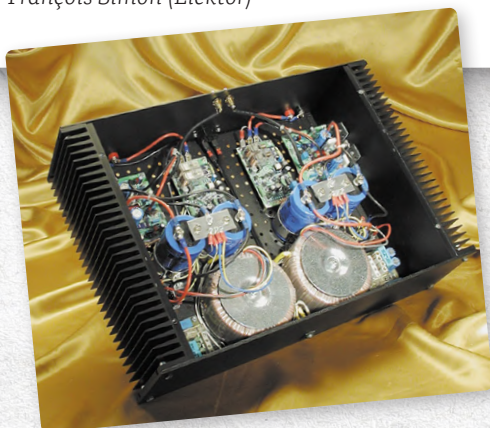
After a long time, we decided to build this MOSFET amplifier. It was really difficult because many of the components are no longer produced. But even during the first start-up, we encountered problems. At test point TP3, the DC voltage is about 6 V. Are you able to send us more data for other test points? After so much hard work and waiting for components, we don't want to give up.

Zoran Gjurovski (North Macedonia)

Unfortunately, we're not in a position to provide detailed construction assistance to our readers for a project published over twenty years ago. Nevertheless, here are some "quick tips" which can be relevant for this kind of vintage Elektor projects:

- Given that the circuit's main transistors are obsolete, I imagine you've used platforms such as eBay or Aliexpress to get hold of them? Beware, there is a high probability that what you received are not genuine components. Some sellers on these platforms sell counterfeit components, which don't necessarily have the same voltage and current ratings as the original part; sometimes they are even a completely different type of component.
- When using eBay or your local classifieds, look for "New Old Stock" (NOS) items sold locally by small businesses that are clearing their stock or by private individuals. This may increase your chances of success. I recommend testing what you receive using a transistor tester such as the LCR-T4 or the DCA75 pro, depending on your budget.
- Of course, also check all solder joints with a magnifying glass after assembly and look for possible mistakes. If the circuit still doesn't work as described in the article, you can turn to one of the audio enthusiasts' forums on the internet. Good luck!

Jean-François Simon (Elektor)



40 Years — A Lifelong Journey with Elektor

It was the last days of winter 1983 in Isfahan, Iran. I remember the very nice, sunny weather as I entered a bookstore on the boulevard. I had heard about Elektor Magazine from a friend with an electronics shop and was eager to have my own copy. After I asked, the shopkeeper pulled one from a drawer — the March 1983 issue. The glossy cover caught the light, and an 8-pin IC seemed to glow on the front. Before I could reach for it, he shook his head. "It's reserved," he said. Disappointed, I asked if he could order another copy. He couldn't, because they were out of stock. But he advised me to subscribe. At the time, getting foreign magazines was difficult due to strict regulations. It took nearly a year, but by April 1984, I had secured an annual subscription. From then on, Elektor arrived in my post office box almost every month.

Back then, I was a university student studying electrical engineering with a growing passion for electronics. Elektor became my companion, bringing projects, schematics, and the unmistakable PCB layouts that defined its style. The magazine was so practical that every issue had something interesting — at least one circuit or project that anyone could try and build. The Summer Circuits issues particularly amazed me, packed with simple, practical circuits, many contributed by readers. I often adapted them into my own designs!

Through the years, Elektor evolved. New features appeared: color printing, an annual CD/DVD, a website, downloadable PDFs, professional projects, and books. The magazine adapted to the rapid advances in electronics, shifting toward microcontrollers and digital designs. While some traditions changed and the staples disappeared in favor of glue, the core of Elektor remained: a source of inspiration and learning.

Now, in 2025, I'm still subscribed to Elektor. Looking back, I realize how much I owe to it. It shaped my journey in electronics, providing knowledge, ideas, and joy. For that, I am deeply grateful — to the magazine, to its founder, and to all those who have contributed to its legacy.

Sam Abadani (Iran)

Thank you very much for this very touching story, and for your loyalty to our magazine!

Jean-François Simon (Elektor)

Jens Nickel (Elektor)



Project Update #4: ESP32-Based Energy Meter

Elektor 11/2024, p. 20 (240349)

We have received quite a few questions about the ESP32-based energy meter developed by Elektor engineer Saad Imtiaz. Thank you all for your emails! Here is a summary of the main points discussed:

- *Can this meter also be used to detect when energy is being fed back into the grid and generate a signal for this? Possibly with an Arduino interface to switch on a power consumer in such a case, so that less energy is fed back into the grid? This could help avoid feed-in fees charged by the utility company.*

Yes, the firmware provides real-time measurements of both positive and negative power, including active, reactive, and apparent power. Almost all the functionality of the M90E32AS metering IC is available. I highly encourage readers to fork the GitHub repository (<https://github.com/ElektorLabs/esp32-energymeter>) and contribute by enhancing the existing MQTT-based firmware with additional features and optimizations. Your contributions will help improve functionality and expand the project's capabilities.

- *I look forward to your research about the AI processing of the captured data. When and where will you publish the results?*

At present, I am unable to provide a timeline for sharing the results or progress on the AI aspect of this project. This will be featured in both the magazine and the [elektormagazine.com](https://www.elektormagazine.com) website.

- *I will operate the board in three-phase mode. Can I use my existing SCT013-020 current transformer (20 A / 1 V) for this?*

Yes, you can utilize your existing SCT013-020 current transformers. However, you will need to calibrate in software appropriately and also modify the burden resistors present on the PCB. To simplify the process, I recommend using the SCT013-000 (100 A / 50 mA) version of this current sensor instead. The current input channels for this sensor

are calibrated accordingly, but you will still require slight software calibration to achieve the most accurate current readings.

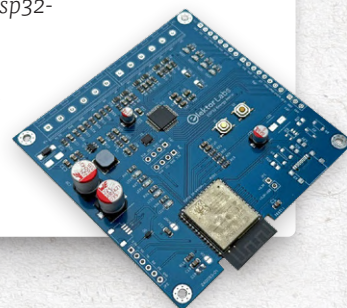
- *There is no USB-C connector on the board. Is this intentional? What's the part number for it, so I can add one? How can I update the firmware, or enable Over-The-Air updates without USB? Are additional components available as a kit from Elektor?*

We decided not to include the USB-C port for two main reasons, safety considerations and avoiding additional complexity and cost. Fortunately, programming the meter is still straightforward using an external ESP32 programmer, such as the *Makerfabs ESP32 Programmer USB2UART*, or the *Connect* programmer from Soldered (<https://soldered.com/product/connect-programmer>, available at Reichelt with part number 333073 for example). Simply connect the pins from the ESP32 programmer to the corresponding pins at the JP2 header on the Energy Meter (on the lower-left side of the PCB). Once connected, you can program it just as you would any standard ESP32 development board.

After you've programmed the board for the first time, you can enable OTA (over-the-air) updates for easier firmware uploads without needing a physical connection. I recommend that you use it with Home Assistant and use the ESP Home firmware to keep it simple and easy, as it also has OTA features.

If you wish to add a USB-C port in the future, you'd need to source the required SMD components yourself. Presently, Elektor does not offer a kit specifically for adding the USB-C functionality. For the complete bill of materials, please check out the GitHub repository of the project (<https://github.com/ElektorLabs/esp32-energymeter>). I hope this helps!

Saad Imtiaz (Elektor, article author)



—
Ideas or Feedback?
Got a bright idea or valuable feedback
for Elektor? Reach out to us at
editor@elektor.com. We're eager to
hear from you!
—

250104-01

Elektor Mini-Wheelie Self-Balancing Robot

The Elektor Mini-Wheelie is an experimental autonomous self-balancing robot platform. Based on an ESP32-S3 microcontroller, the self-balancing robot is fully programmable using the Arduino environment and open-source libraries. Its wireless capabilities allow it to be controlled remotely over Wi-Fi, Bluetooth or ESP-NOW or to communicate with a user or even another robot.



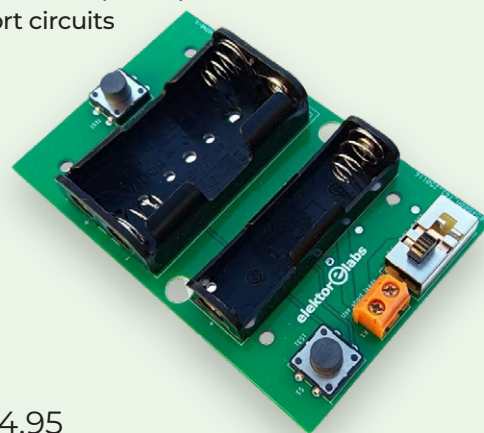
Price: €99.95

Member Price: €89.96

www.elektor.com/21087

Elektor Milliohm-meter Adapter

The Elektor Milliohm-meter Adapter uses the precision of a multimeter to measure very low resistance values. It is an adapter that converts a resistance into a voltage that can be measured with a standard multimeter. The Elektor Milliohm-meter Adapter can measure resistances below 1 mΩ using a 4-wire (Kelvin) method. It is useful for locating short circuits on PCBs.



Price: €34.95

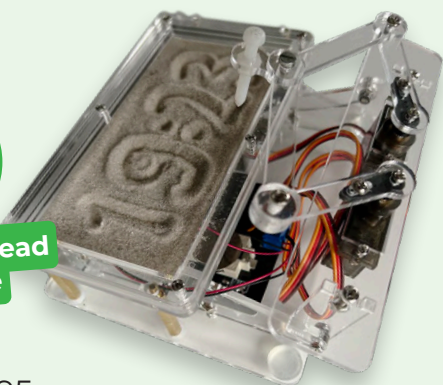
Member Price: €31.46

www.elektor.com/21147

Elektor Sand Clock for Raspberry Pi Pico



incl. Laser Head Upgrade



Price: €84.95

Member Price: €76.46

www.elektor.com/21169

FNIRSI DSO-TC4 (3-in-1) Oscilloscope + Transistor Tester + Signal Generator



Price: €89.95

Member Price: €80.96

www.elektor.com/21146



Raspberry Pi 45 W USB-C Power Supply



Price: €16.95

www.elektor.com/21165

Velleman Multifunctional Precision Drill Set (162 pcs)



Price: €54.95

Member Price: €49.46

www.elektor.com/21148

Geekworm KVM-A3 Kit for Raspberry Pi 4

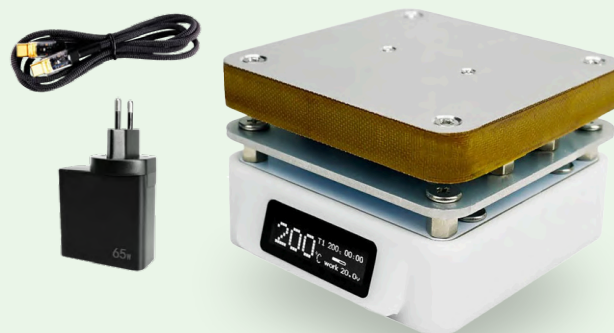


Price: €119.95

Member Price: €107.96

www.elektor.com/21139

SEQUIRE T55 Smart Mini Soldering Hot Plate (55x55 mm)



Price: €49.95

Member Price: €44.96

www.elektor.com/21149

Raspberry Pi Standalone MIDI Synthesizer (2)

Enhancing Our Setup with Intelligence

By Brian Tristram Williams (Elektor)

In Part 1, we explored how to connect a MIDI keyboard to a Raspberry Pi and use FluidSynth to generate sound. Now, we'll take the project to the next level by adding an audio DAC, automating the setup, improving latency for live performance, and integrating AI-driven enhancements.

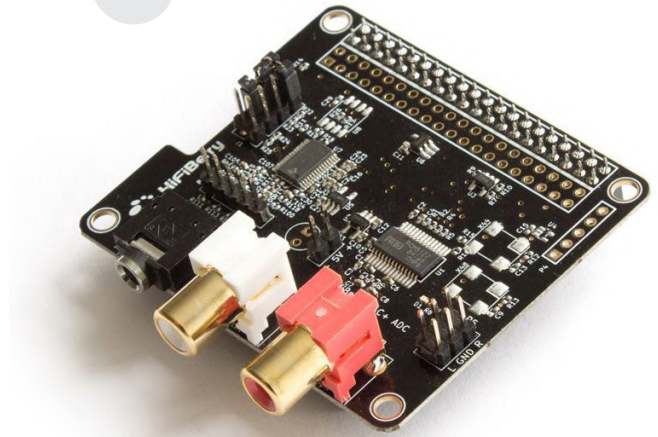


Figure 1: The HiFiBerry DAC + ADC has a lot of nice audio features, but no display.

In the last part of this series [1], we achieved our first goal of getting the Raspberry Pi 5 to output music when sent notes from a standalone MIDI keyboard. This time, I'd like to streamline the setup so that we don't need to have an unnecessary monitor just to hear the audio. We want to run it headlessly.

Solving the Monitor "Problem"

The Raspberry Pi 5 and Zero series by default output audio only via the HDMI connection to the monitor, which in turn strips it out and sends it to speakers or a headphone jack. One option would have been an HDMI audio extractor, but the ones I saw online were clunky, expensive, unwieldy, required cables, and so on, so I sought out other options.

I looked around and found several digital-to-analog converters (DACs), from USB sound "cards" and USB pigtail adapters, to HATs such as the HiFiBerry DAC + ADC (**Figure 1**), and even an Elektor DIY HAT project by Ton Giesberts in 2017 [2].

There was one creative solution that caught my eye: The Pimoroni Pirate Audio Headphone Amp for the Raspberry Pi [3] (**Figure 2**). What I found impressively creative about it was its inclusion of a

240×240-pixel display, intended to display album art, but you could use it for many other display purposes.

It also has four buttons, which could be used to control your MP3 music playback, but again, they are GPIO-attached, so we can use our own creativity to come up with other ways to make use of those.



Figure 2: The Pimoroni Pirate Audio Headphone Amp for Raspberry Pi has a convenient display and four GPIO buttons.



Finally, what impressed me was the form factor. It fits snugly over a Raspberry Pi Zero or variants. It will of course work well with the Raspberry Pi 5 I already started this project out on, but I couldn't resist trying to compact this solution a little more.

The important stuff for me is underneath (**Figure 3**): The DAC, the headphone amplifier, and the headphone socket.

For a first test, I did need to do the usual Raspberry Pi OS installation from [4], and then I set up the Pirate Audio application for MP3/Spotify/SoundCloud, etc., playback just to try it out, which you can see in my video at [5]. But, the real goal here is not MP3, but MIDI.

The Software Side

Our goal is to make the Raspberry Pi a smarter, more expressive MIDI synth that doesn't require manual configuration each time and can dynamically respond to how you play. We'll implement dynamic velocity adjustments, AI-powered chord generation, and even real-time accompaniment using machine learning models like Google's Magenta.

Automating the MIDI Setup

Making FluidSynth Auto-Start

Manually starting FluidSynth every time is inefficient. Let's set it to launch on boot:

1. Create a *systemd* service file:

```
sudo nano /etc/systemd/system/fluidsynth.service
```

2. Add the following content:

```
[Unit]
Description=FluidSynth MIDI Synthesizer
After=sound.target

[Service]
ExecStart=/usr/bin/fluidsynth -a alsa -g 1
-o synth.polyphony=64 /usr/share/sounds/sf2/
FluidR3_GM.sf2
Restart=always

[Install]
WantedBy=default.target
```

3. Save and exit (*CTRL+X*, then *Y*).
4. Enable and start the service:

```
sudo systemctl enable fluidsynth
sudo systemctl start fluidsynth
```

Auto-Connecting the MIDI Keyboard

Next, we need to automatically route MIDI input to FluidSynth:

1. Create a startup script:

```
nano ~/auto_midi.sh
```

2. Add:

```
#!/bin/bash
sleep 5 # Allow time for MIDI devices to initialize
aconnect 24:0 128:0
# Adjust these numbers to match your hardware
```

3. Make it executable:

```
chmod +x ~/auto_midi.sh
```

4. Open *crontab* editor by typing this in the terminal:

```
crontab -e
```

then add this line:

```
@reboot /home/pi/auto_midi.sh &
```

Now, when you power on the Raspberry Pi, it will automatically recognize and connect your MIDI keyboard to FluidSynth without any manual intervention.

Optimizing for Real-Time Performance

Latency is critical for live performance. Here's how we can reduce it:

Use a Low-Latency Kernel:

```
sudo apt install linux-lowlatency
```

Adjust Buffer Settings in FluidSynth:

```
fluidsynth -a alsa -g 1 -o audio.period-size=128 -o
synth.polyphony=128 /usr/share/sounds/sf2/FluidR3_GM.sf2
```

Lower buffer sizes reduce latency but may cause glitches if they're too low.

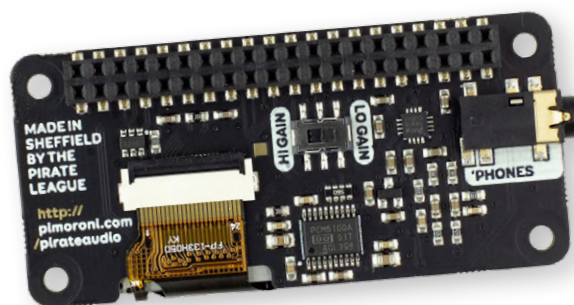


Figure 3: On the underside of the Pirate Audio board is a high-quality DAC based on a Texas Instruments PCM5100A DAC and a PAM8908 headphone amplifier.



Listing 1: velocity.py.

```
import rtmidi
import numpy as np

midi_in = rtmidi.MidiIn()
midi_out = rtmidi.MidiOut()
midi_in.open_port(0)
midi_out.open_port(1)

velocity_history = []

def smooth_velocity(velocity, factor=0.2):
    velocity_history.append(velocity)
    if len(velocity_history) > 10:
        velocity_history.pop(0)
    return int((1 - factor) * velocity + factor * np.mean(velocity_history))

while True:
    msg = midi_in.get_message()
    if msg:
        note, velocity = msg[0][1], msg[0][2]
        new_velocity = smooth_velocity(velocity)
        midi_out.send_message([msg[0][0], note, new_velocity])
```



Listing 2: chords.py.

```
import rtmidi
import random

chords = {
    'C': [0, 4, 7],
    'G': [0, 4, 7],
    'D': [0, 3, 7],
    'A': [0, 4, 7]
}

midi_in = rtmidi.MidiIn()
midi_out = rtmidi.MidiOut()
midi_in.open_port(0)
midi_out.open_port(1)

while True:
    msg = midi_in.get_message()
    if msg:
        note, velocity = msg[0][1], msg[0][2]
        key = random.choice(list(chords.keys()))
        for interval in chords[key]:
            midi_out.send_message([msg[0][0], note + interval, velocity])
```



Trying Out Intelligent Enhancements

Let's try two experiments. You can download the Python scripts for them at [6].

1. Dynamic Velocity Adjustment

A rigid velocity response makes MIDI playing feel robotic. Let's dynamically adjust velocity based on past inputs, creating a more human feel.

The Python script is shown in **Listing 1**. It uses the *rtmidi* library to process MIDI messages in real time, applying velocity smoothing before sending them to an output device. It begins by initializing a MIDI input and output interface, each connected to a specific port.

To reduce abrupt changes in velocity, the script maintains a history of the last ten velocity values. When a new MIDI message is received, the velocity is blended with the average of recent values using a smoothing factor. This creates a more natural transition between notes rather than sharp spikes or drops in intensity.

A continuous loop listens for incoming MIDI messages. When a note is played, the script extracts its velocity, applies the smoothing function, and sends the modified message to the MIDI output.

2. Scripted Chord Generator

Instead of playing a static harmony, the script in **Listing 2** transforms single MIDI notes into randomly selected chords. It defines a set of chords as note intervals and listens for incoming MIDI messages. When a note is played, the script picks a random chord and sends the original note along with additional notes based on the chosen chord's intervals. This creates a dynamic, harmonized effect where each key press generates a different chord.

By automating MIDI setup and optimizing latency, we've transformed our Raspberry Pi MIDI synth into an intelligent, dynamic instrument. It's not AI up to now, but it creates a more musically "intelligent" accompaniment system.

With AI tools at our disposal, including training our own models, we can now explore real-time melody prediction, AI-assisted improvisation, and interactive generative music systems — we have only reached the tip of the iceberg. Let's start digging in the next installment!

240714-B-01

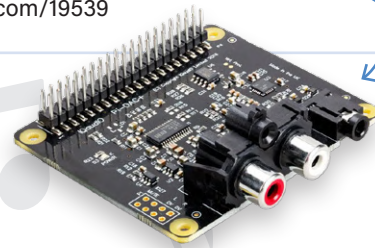
Questions or Comments?

We'd love to hear about how you are using AI and how it has affected you over the past two years. If you have questions or comments, email me at brian.williams@elektor.com. You can also catch me on Elektor Engineering Insights each month on YouTube, and you can find me @briantw on X.



Related Products

- > **Pimoroni Pirate Audio Headphone Amp for Raspberry Pi**
www.elektor.com/20168
- > **HiFiBerry DAC+ ADC**
www.elektor.com/18948
- > **IQaudio Pi-DAC+ (for Raspberry Pi)**
www.elektor.com/19539



WEB LINKS

- [1] Brian Tristam Williams, "Raspberry Pi Standalone MIDI Synthesizer (1)," Elektor 3-4/2025: <https://elektormagazine.com/240714-01>
- [2] Ton Giesberts, "Audio DAC for Raspberry Pi," Elektor 7-8/2017: <https://elektormagazine.com/magazine/elektor-201707/40496>
- [3] Pirate Audio: Headphone Amp for Raspberry Pi: <https://shop.pimoroni.com/products/pirate-audio-headphone-amp>
- [4] Raspberry Pi OS — Download and Installation: <https://raspberrypi.com/software>
- [5] Unboxing the Pimoroni Pirate Audio Headphone Amp for the Raspberry Pi, Elektor TV: <https://youtu.be/5rnQldmviRg>
- [6] GitHub repository for these experiments: <https://github.com/briantw/pi-midi>

Nortonized Wien Bridge Oscillator

Small Changes Yield Significant Improvements

By Alain Delahodde (France)

Analog enthusiasts are likely familiar with the term Wien bridge, a fundamental electronic circuit also known as an RC generator or Wien-Robinson oscillator. Even such a standard circuit can be optimized, as this article demonstrates.

The classic Wien bridge oscillator is based on the half bridge in **Figure 1**, which combines series and parallel resistors and capacitors.

With two identical resistors and capacitors, the transfer function of the half bridge is

$$\frac{s}{e} = \frac{RZ}{R^2 + Z^2 + 3RZ}$$

where $Z = 1 / j C \omega$.

If $R C \omega = 1$, then we get $R^2 + Z^2 = 0$ and

$$\frac{s}{e} = \frac{1}{3}$$

Adding an opamp results in a complete, but idealized, Wien bridge oscillator (**Figure 2**). An amplifier with a gain of 3, as in Figure 2, will yield a transfer function value of 1, causing the circuit to oscillate.

Nortonizing

A drawback of the standard configuration is that the upper capacitor lacks a connection to ground, which is inconvenient for a variable oscillator because the dual variable capacitor's frame can't be grounded. Good insulators and plastic or nylon gears are needed to manipulate the variable capacitor, and it's necessary to compensate for the parasitic capacitance between the frame and the bulk.

Using the Norton transform, series components become parallel, currents become voltages, and vice versa. This transformation is applied in **Figure 3**. The result is

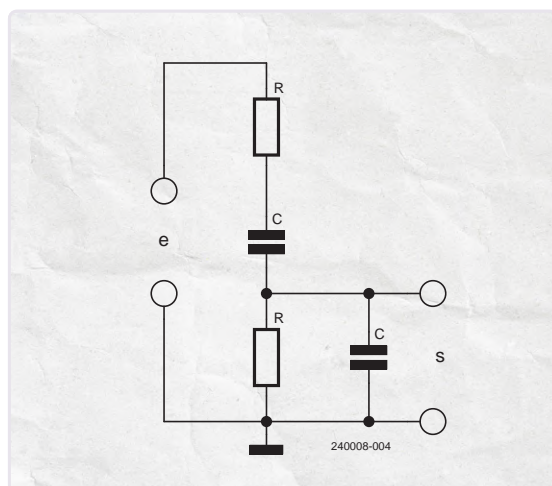


Figure 1: Half Bridge.

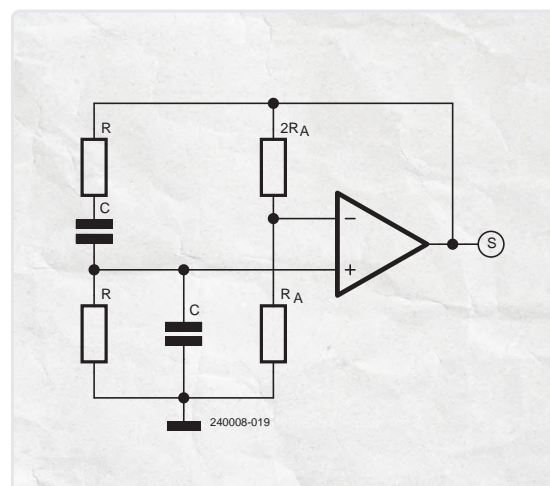


Figure 2: Adding an opamp to a Wien bridge results in an (idealized) Wien bridge oscillator.

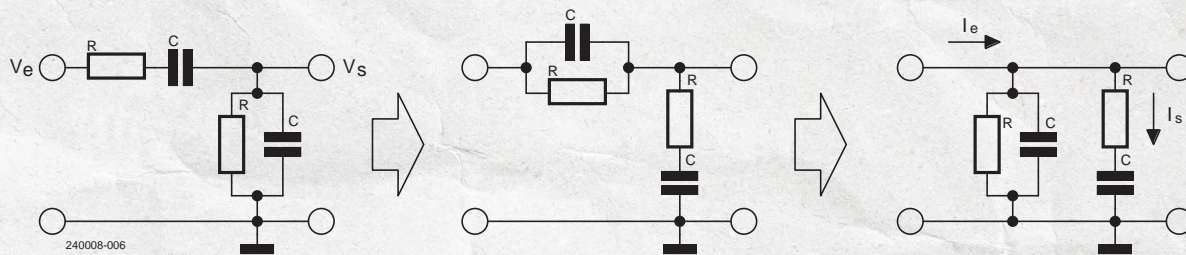


Figure 3: Applying Norton transformation to a Wien bridge allows both capacitors to be grounded.

$$\frac{I_s}{I_e} = \frac{RZ}{R^2 + Z^2 + 3RZ}$$

and if $R C \omega = 1$ it gets

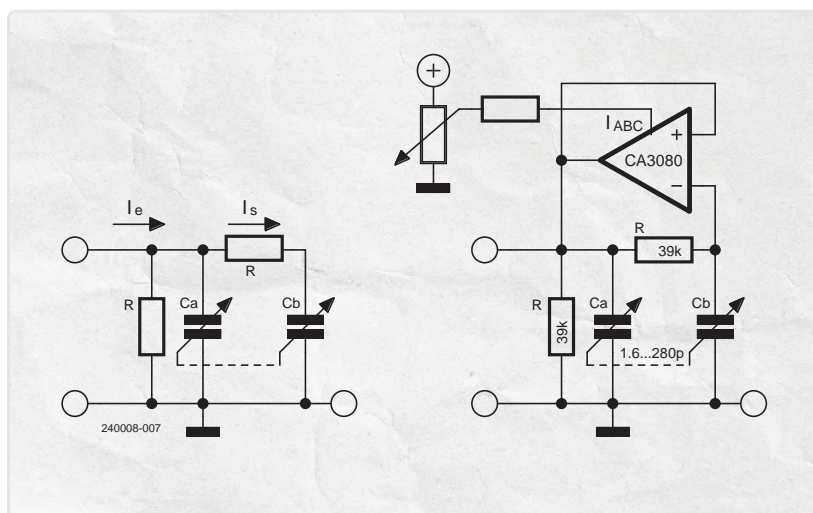
$$\frac{I_s}{I_e} = \frac{1}{3}$$

The main advantage of this configuration is that both capacitors are now grounded, allowing the dual variable capacitor's frame to be connected to the bulk.

A first and simple setup to confirm this concept is shown in **Figure 4**. The amplifier IC in this circuit is an OTA (operational transconductance amplifier) of type CA3080. It "reads" the output current I_s through the horizontal resistor and injects I_e at the entrance. This setup operates from 15 to 250 kHz, but adjusting I_{ABC} requires the amplitude to be in the range of a few tenths of millivolts to avoid distortion.

Improvements and Variants

The following setup in **Figure 5** uses only a part of the voltage over the horizontal resistor, reducing distortion and allowing a higher output amplitude of several volts. With a ± 5 V power supply, the circuit delivers a constant 4 V_{pp} output signal with low distortions in the range of 15 to 250 kHz. For IC1, the same OTA as in Figure 4 is used. IC2 is a buffer to achieve a low-impedance output, and



IC3 controls I_{ABC} via the function $I_e = 3 \times I_s$, serving as feedback control for the output amplitude.

Figure 6 shows that the OTA can be replaced by a current generator using a simple opamp. Applied to the previous setup in Figure 5, this leads to a very similar design, but with better performance. With

$$I = \frac{v - e}{\rho}$$

Figure 4: A Nortonized adjustable Wien bridge with the help of an OTA.

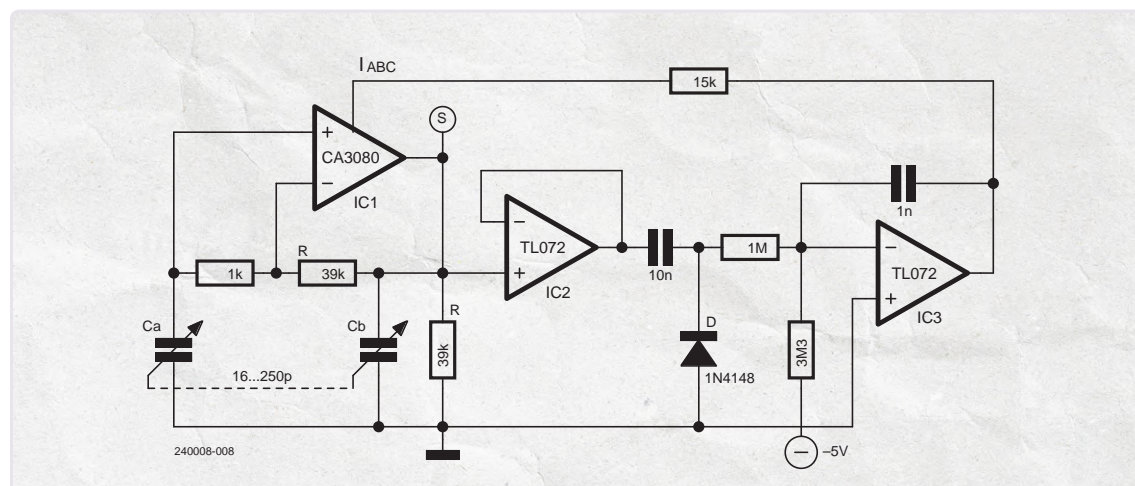


Figure 5: Improved Nortonized Wien bridge with amplitude stabilization.

Figure 6: This circuit around an opamp can replace an OTA.

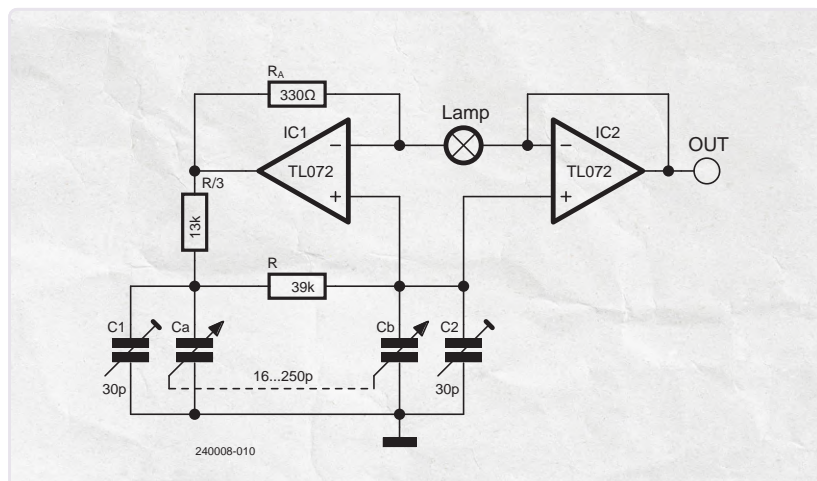
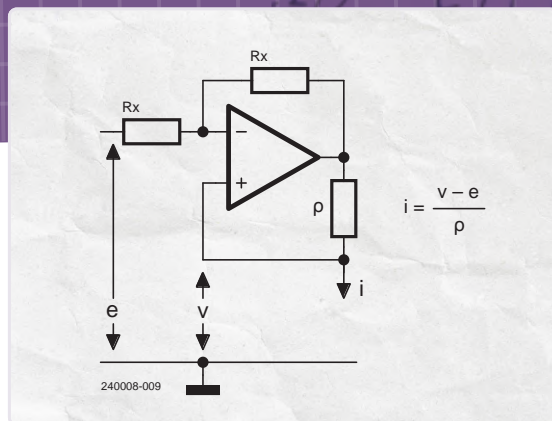



Figure 7: Improved nortonized Wien bridge without OTA, with trimming capacitors.

it's clear that I_e will be equal to $3 \times I_s$ if $\rho = R/3$. This further improvement results in the circuit of **Figure 7**. With $R = 39 \text{ k}\Omega$ ($R/3 = 13 \text{ k}\Omega$) and the same variable capacitor, the circuit covers a 1:10 range from 15 to 150 kHz. Powered by $\pm 12 \text{ V}$, the amplitude is 15 V_{pp} ($\approx 5 \text{ V}_{eff}$). Two trimming capacitors of 30 pF are added to each half of the dual variable capacitor to limit the frequency range and equalize the output amplitude over frequency. The output signal amplitude is stabilized by a small tungsten lamp (12 V / 20 mA), which acts as temperature and therefore voltage dependent resistor. In a balanced state, its resistance is equal to $R_A = 330 \Omega$.

The frequency range depends on the three resistors, $2 \times R$

and $R/3$. **Figure 8** shows that lowering the range by a factor of 10 is possible by modifying only two resistors: In the middle circuit the 0Ω resistor from the left circuit is changed to $9 \times R/4$, and the left R then changes to $10R$. The right circuit shows that the middle one is functionally identical to the standard way of using $10R$ to reduce the frequency by a factor of 10. This variety of Wien bridge oscillator allows for the use of a dual variable capacitor with its frame connected to common ground, making it more convenient to create a multi-decade sine generator.

The final circuit in **Figure 9** shows how to create quadrature outputs by adding two more opamps. The numbers in front of some R_x resistors indicate a ratio. A value of $10 \text{ k}\Omega$ is a good choice for R_x . A quad opamp such as the TL074 is sufficient for this circuit. 

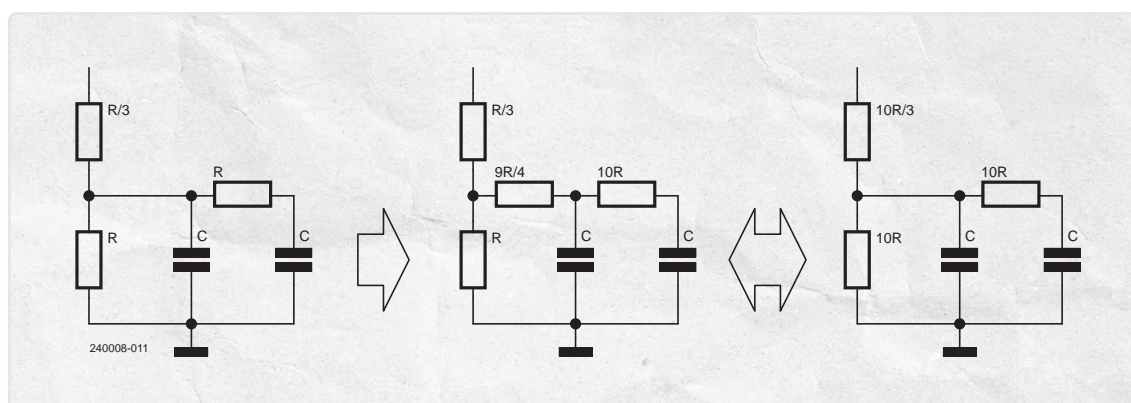
240008-01



About the Author

Alain Delahodde is an electronic engineer (Ecole de l'air and Supélec) who served in the French Air Force, specializing in radar Air Defense Systems and Electronic Warfare. After retiring, he joined Air France for French AWACS maintenance. A lifelong fan of electronic design, he enjoys finding innovative solutions in metrology and measuring apparatus. Influenced by publications such as *Radio Plans*, *Le Haut-Parleur*, *Electronique Pratique*, and *Elektor*, he values both digital and analog electronics.

Figure 8: There are (at least) two equivalent ways to reduce the frequency by factor 10.



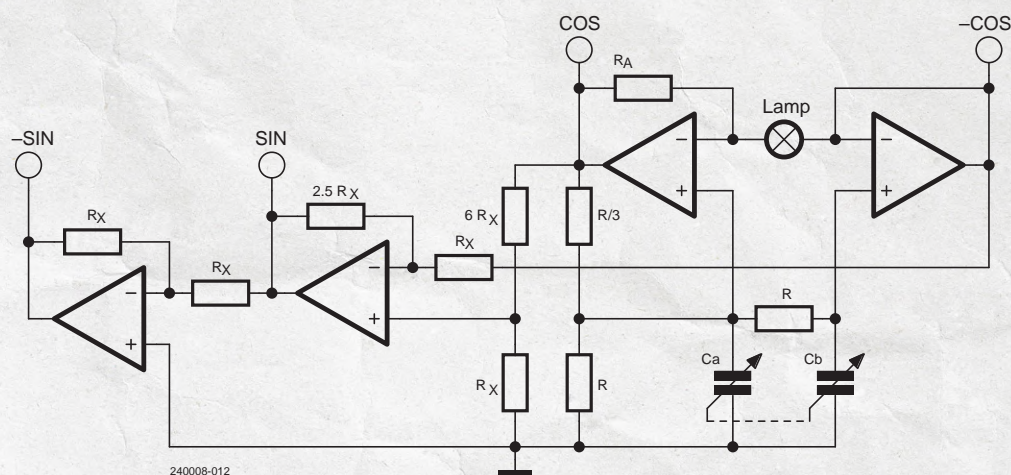


Figure 9: A Nortonized Wien bridge with quadrature outputs.

Questions or Comments?

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



Related Products

➤ **OWON HDS307S (3-in-1) 2-ch Oscilloscope (70 MHz) + Multimeter + Signal Generator**
www.elektor.com/20880



They trust us, do you?

Great service.
Nice store, fast shipping, good packing and documentation.
Jun 27, 2024

Excellent product at a good price
Excellent product at a good price with fast trouble free delivery.
Date of experience: November 25, 2023

Easy ordering process(online)
Easy ordering process(online) , good communication of delivery steps.
Overall smooth process
Date of experience: June 01, 2024

Great service
Great service!
Easy to purchase and delivered just in time as promised.
Date of experience: February 12, 2024

We love electronics and projects, and we do our utmost to fulfill the needs of our customers.
 The Elektor Store: **'Never Expensive, Always Surprising!'**

Elektor Store

Reviews 365 • Excellent



VERIFIED COMPANY



Check out more reviews on our Trustpilot page: www.elektor.com/TP/en

Or make up your own mind by visiting our Elektor Store, www.elektor.com



Putting a \$0.10 Controller to the Test

The CH32V003 RISC-V Microcontroller and MounRiver Studio in Practice

By Tam Hanna (Hungary)

The open-source RISC-V ISA and ever-dropping manufacturing costs make a lot of innovation possible. WCH offers a well-equipped 32-bit controller at a unit price of \$0.10. But what is it like to work with the controller and the associated MounRiver Studio IDE? We put it to the test.

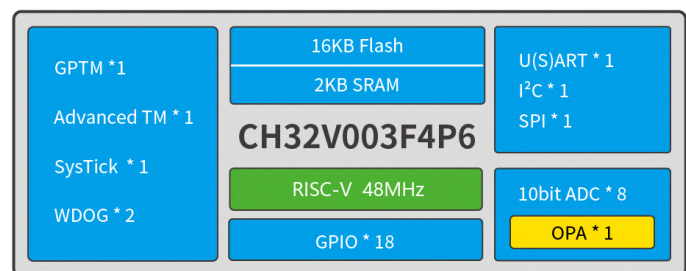


Figure 1: The block diagram provides information about the included peripherals. (Source: WCH)

The demands of users are growing constantly, while inflation and the economic crisis are causing disposable income to shrink. If you can save costs on the bill of materials by using cheaper parts, you give your product a valuable competitive advantage.

There is no question that you can always find microcontrollers based on the once popular 8051 architecture, for example, for well under a euro/dollar on notorious mail-order platforms. The disadvantage of these offers, however, is that the chips are often not available in unlimited quantities — once a redesign is required, all the cost benefits will go out the window. Furthermore, it has always been said that really cheap chips are not particularly easy to be programmed.

With the availability of the open-source RISC-V architecture, more powerful controllers can now be realized at low cost because the license fees for ARM are no longer applicable. In addition, the structures in the high-end sector are becoming smaller and smaller, which creates space in fabs.

With the CH32V003, WCH offers a very powerful 32-bit microcontroller, which nevertheless costs only 10 US cents per unit. Since the chip can also be programmed with a — very convenient — IDE and a more or less fully-fledged C dialect and comes with a lot of peripherals (**Figure 1**), this should be enough to motivate us — and you — to experiment with it.

Buying an Evaluation Board, Setting Up the IDE

It has long been known that the Chinese semiconductor industry is relatively indifferent to Western markets. As a result, the in-house development tools are all too often either not available to Western developers at all or only available very late. In the case of WCH, the situation is better in that the company is open to working with Western customers, at least in principle.

The first step is to order hardware — the author recommends using the American hardware provider Tindie. In addition to the evaluation board provided under [1], make sure you also purchase the debugger device provided under [2]. Once it has been delivered by

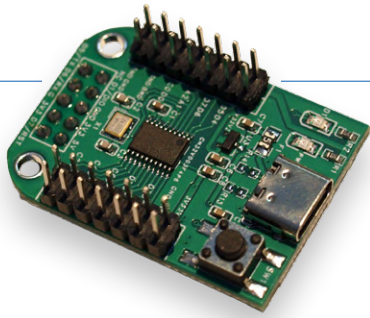


Figure 2: The evaluation board with pins soldered on...

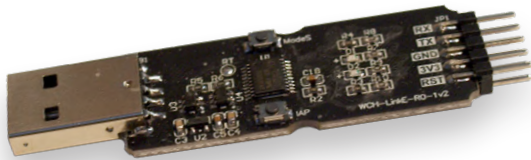


Figure 3: ...and the associated debugging device.

mail (which took a good three weeks to Hungary), you will have the hardware shown in **Figure 2** and **Figure 3** in place.

Due to the well-known problems with the pick-and-place processability of the various connectors, some manual work is required in the case of this hardware. However, with a pin spacing of 2.54 mm, soldering the connector strips is a task that can be easily accomplished with simple tools. It is important to note that the plug on the programming device is not protected against polarity reversal. The author used a leftover socket strip to connect the two boards together (**Figure 4**).

In terms of the actual development environment, WCH supports Windows, Mac OS, and Linux equally; the actual tool is an IDE that WCH obtains from MounRiver.

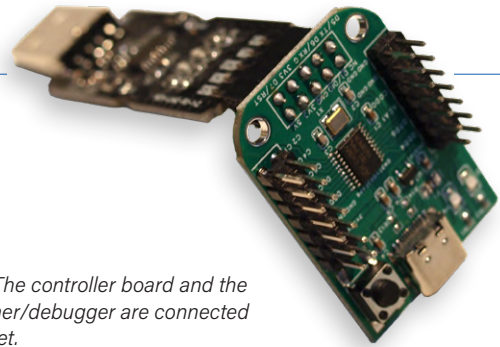


Figure 4: The controller board and the programmer/debugger are connected via a socket.

For reasons of convenience, the author will use Windows 10 in the following steps. Open [3] to download the file *MounRiver_Studio_Setup_V192.zip*. Then unzip the archive to a conveniently accessible location on the file system and start the installation wizard — the author chose the folder *C:\MounRiver* as the location for the following steps.

For some time now, MounRiver has also offered a variant based on Visual Studio Code (MRS2). For reasons of convenience though, the author uses the Eclipse-based version of the IDE in the following steps.

Problems? Downgrade!

In the author's tests, the older version V183 of MounRiver Studio proved to be more stable in some cases. If you have problems with the latest version, you can download the older version at <https://tinyurl.com/MounRiver-Setup-zip>.

The first start of MounRiver takes a little longer because the IDE registers the various drivers for the different command devices with the operating system. After the work is done, you will see the start screen shown in **Figure 5**.

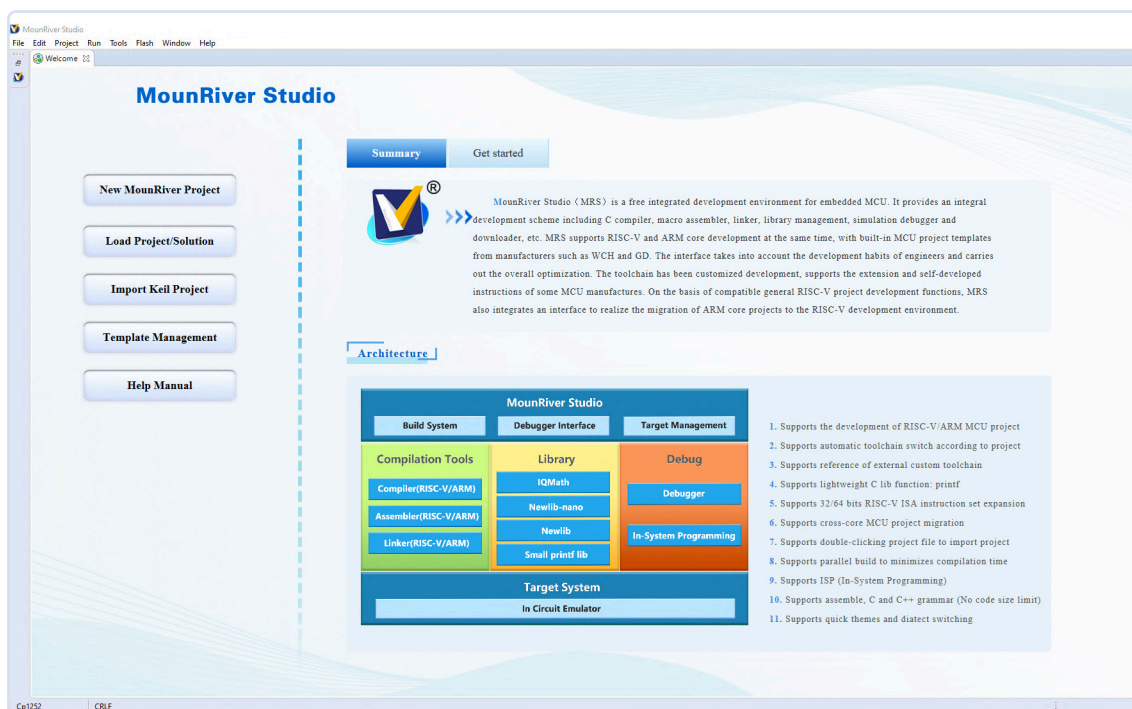


Figure 5: MounRiver is ready for action!

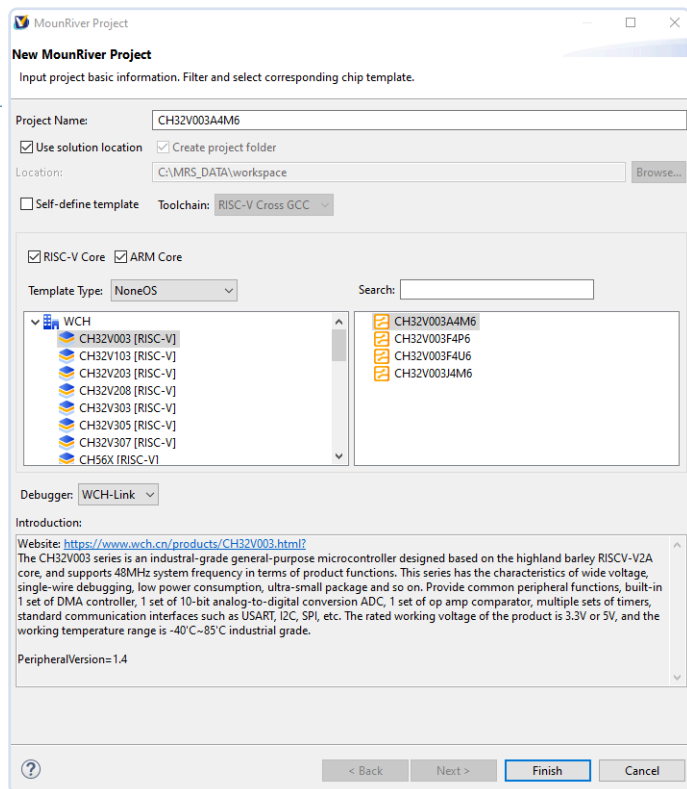


Figure 6: Selecting the controller type.

Creating a Project Skeleton

After the successful start of MounRiver, we click on the button *New MounRiver Project*. The IDE responds by displaying the project selection wizard shown in **Figure 6**.

Since we are using a controller of the type CH32V003F4P6 here, it is recommended that you select the appropriate template. We accept the remaining settings exactly as they are and then click *Finish* to prompt the system to create the solution. Your efforts will be rewarded by the appearance of a new project skeleton that you can explore as usual in Eclipse.

When providing the various libraries, WCH follows the ST and GigaDevice example: Each project skeleton is given its own complement of hardware drivers. These can therefore be conveniently analyzed using the tracking functions included in Eclipse, which, in conjunction with the datasheet, helps with familiarization.

By default, MounRiver generates a comparatively extensive example project skeleton that receives the incoming signals at pin PD5, inverts them and then outputs them to PD6. In both cases, the example uses the onboard hardware UART.

The next step is to perform an initial startup. Connect the control unit and the circuit board as shown in Figure 4. If you solder sockets onto the bottom side, you can avoid fiddling around with jumper wires, which can sometimes be quite a hassle, especially when debugger connections are set for fast operation.

USB Via Jumper Wires

The author is aware of attempts to route for example USB 2.0 via jumper wires. What may work “experimentally” can quickly become an additional (and avoidable) source of faults, especially in an EMI-prone laboratory.

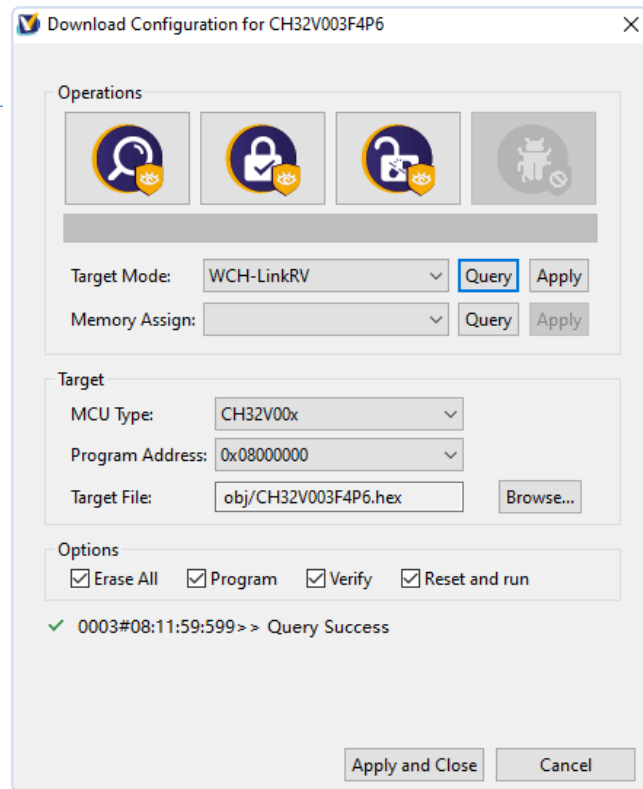


Figure 7: The target was successfully recognized.

In any case, the light labeled PWR should then light up — provided that the board is in a virgin state.

The first step is to set up a small debug test. To do this, click on the line to the left of the `RCC_APB2PeriphClockCmd` command to set a breakpoint. In the next step, click on the down arrow next to the configuration icon to open the configuration manager (**Figure 7**). Then click on the *Query* button in the *Target Mode* section to make sure that the correct debug tool (WCH-LinkRV) is being used. Finally, click on *Apply and Close*.

For the actual program execution, we click on the *Debug* option in the next step. We are then notified of the absence of a binary file. To fix this problem, it is sufficient to initiate a recompilation via *Project* → *Build Project*. As with many other embedded systems, the first run takes a little longer because the various hardware drivers also need to be passed through the compiler.

In the next step, we open the debugger configuration management and, as shown in **Figure 8**, decide on a variant that is suitable for our core. Then, click the *Debug* button to tell the program to start. After the start, you will initially find yourself in the function `handle_reset`. Note that a “convenience” breakpoint has been set before the first instruction, a practice you might be familiar with from STM32. After that, you will find yourself in the expected method, where you can manipulate the various values in the controller.

Experiments with Peripheral Devices

In the following steps, we will connect the WCH controller to a potentiometer and use its setting to control an LED. The first task is to get the ADC of the chip up and running.

To do this, we switch to the .c file, where we call a group of initialization methods:

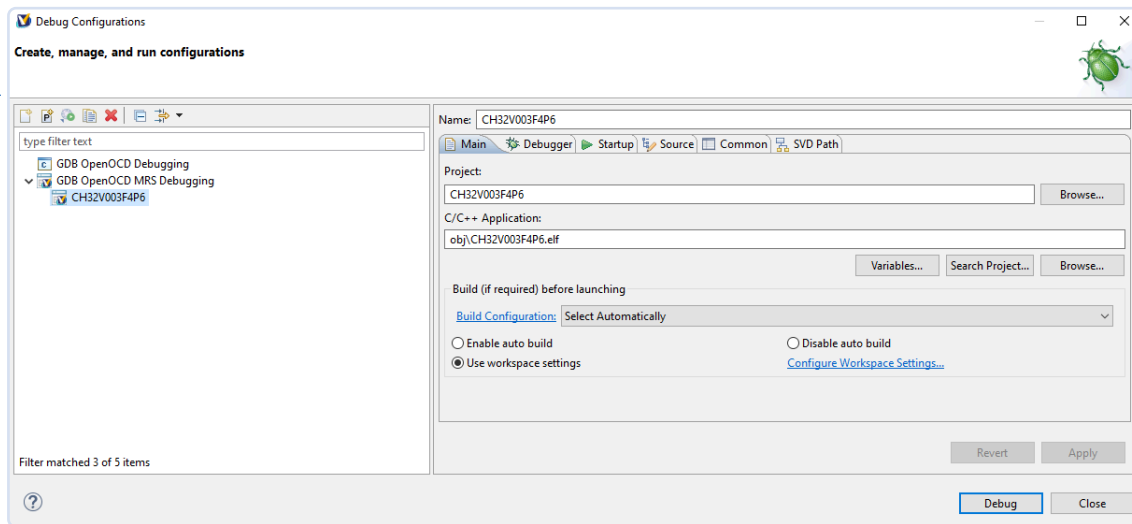


Figure 8:
This configuration leads
to the debugger.

```
int main(void)
{
    u16 adc_val;

    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
    Delay_Init();
    USART_Printf_Init(115200);
    printf("SystemClk:%d\r\n", SystemCoreClock);

    ADC_Function_Init();
}
```

Calling the `USART_Printf_Init` method ensures that the `printf` output is sent via the internal UART – this is important because it means you can capture it with a logic analyzer. The `ADC_Function_Init` function is responsible for the actual configuration of the ADC; we will take a closer look at this in a moment.

But before we do that, let's take a look at the actual application code, which — as is usually the case — is located in an endless loop:

```
while(1)
{
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
    while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
    adc_val = ADC_GetConversionValue(ADC1);
    Delay_Ms(500);
    printf("val:%04d\r\n", adc_val);
    Delay_Ms(2);
}
```

Our first task is to call the method `ADC_SoftwareStartConvCmd` to prompt the ADC to start a new conversion. As with almost all microcontrollers, the conversion does not happen instantaneously. For this reason, we check the value of the `ADC_FLAG_EOC` bit — the `while` loop only returns when the ADC indicates the end of the conversion by setting this flag.

After that, all that is still missing is a call to `ADC_GetConversionValue` to write the value to a variable. The specification given in the data sheet (see **Figure 9**) states that the value range goes from 0 to 2^{10} .

For an initial function test, it is then recommended to place a breakpoint at the second `Delay` command and to harvest the value in

the variable using the debugger that has just been introduced.

Next, we will turn to the actual configuration of the ADC. WCH has opted for an interesting approach here. The individual peripheral devices to be initialized are almost always described using an `InitTypeDef` class:

```
void ADC_Function_Init(void)
{
    ADC_InitTypeDef ADC_InitStructure = {0};
    GPIO_InitTypeDef GPIO_InitStructure = {0};
```

In the interest of lower energy consumption, the CH32V003 also features *clock gating*, which is why we call the `RCC_APB2PeriphClockCmd` method in the next step to ensure that the microcontroller's required peripherals have access to the clock signal:

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC |
                        RCC_APB2Periph_GPIOD, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
RCC_ADCLKConfig(RCC_PCLK2_Div8);
```

In the next step, we need to configure the GPIO pin as an input. This is done by parameterizing the `GPIO_InitTypeDef` object, which is then passed to the `GPIO_Init` method:

```
//PC4
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIOC, &GPIO_InitStructure);
```

1.4.11 Analog-to-digital converter (ADC)

The product is embedded with a 10-bit analog/digital converter (ADC) that shares up to eight external channels and two internal channel samples, with programmable channel sampling times for single, continuous, sweep or intermittent conversion. Provides analog watchdog function allows very accurate monitoring of one or more selected channels for monitoring channel signal voltages. Supports external event-triggered transitions with trigger sources including internal signals from the on-chip timer and external pins. Support for using DMA operation. Supports external trigger delay function. When this function is enabled, the controller delays the trigger signal according to the configured delay time when an external trigger edge is generated, and the ADC conversion is triggered as soon as the delay time is reached.

Figure 9: The AD converters have a resolution of 10 bits.

Initializing the ADC is a bit more complicated because modern microcontrollers usually offer dozens of different ways to control clock sources and conversion triggers. The configuration shown here represents the bare minimum:

```
ADC_DeInit(ADC1);
ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
ADC_InitStructure.ADC_ScanConvMode = DISABLE;
ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
ADC_InitStructure.ADC_ExternalTrigConv =
    ADC_ExternalTrigConv_None;
ADC_InitStructure.ADC_DataAlign =
    ADC_DataAlign_Right;
ADC_InitStructure.ADC_NbrOfChannel = 1;
ADC_Init(ADC1, &ADC_InitStructure);

ADC-RegularChannelConfig(ADC1, ADC_Channel_2,
    1, ADC_SampleTime_241Cycles);
ADC_Calibration_Vol(ADC1, ADC_CALVOL_50PERCENT);
ADC_Cmd(ADC1, ENABLE);
```

Last but not least, the methods `ADC_ResetCalibration(ADC1)` and `ADC_StartCalibration(ADC1)` are called to initiate a software calibration of the ADC. This is an algorithmic procedure that seeks to correct hardware imperfections through numerical processes:

```
ADC_ResetCalibration(ADC1);
while(ADC_GetResetCalibrationStatus(ADC1));
ADC_StartCalibration(ADC1);
while(ADC_GetCalibrationStatus(ADC1));
}
```

If you connect the input pin — perhaps via a small series resistor for safety reasons — to a 10 kΩ potentiometer connected to 0 and 3.3 V, you can execute the program at this point and enjoy the values appearing in the variable.

In the next step, we have to work out how to control the LED. To do this, we initialize another GPIO pin:

```
void GPIO_Function_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOD, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOD, &GPIO_InitStructure);
}
```

The most important difference to the above is that we now set the mode to `GPIO_Mode_Out_PP` and configure the pin as a digital output. Similarly to STM32, it is also possible to adjust the

“maximum edge steepness” using the `GPIO_Speed` attribute — a measure that is not absolutely necessary here due to the maximum clock rate.

The next step could be to instruct the LED to flash according to the following scheme:

```
while(1)
{
    . . .
    printf("val:%04d\r\n", adc_val);
    Delay_Ms(2);
    GPIO_WriteBit(GPIOD, GPIO_Pin_7,
        (i == 0) ? (i = Bit_SET) :
        (i = Bit_RESET));
}
```

What is particularly interesting here is that the `GPIO_WriteBit` method relies on a ternary operator to determine the actual value to be written to the pin transceiver. In the first step, a comparison is made to see whether the value of variable `i` is zero. If it is, an assignment is carried out depending on the respective value. When the program is executed, the LED flashes periodically.

As a final small demo, let's implement a primitive pulse width modulation. It is a fact that almost all modern microcontrollers come with extensive PWM hardware — but getting it up and running requires studying the datasheet. At this point, we will therefore simply use the software PWM:

```
while(1)
{
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
    while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
    adc_val = ADC_GetConversionValue(ADC1);
    //Delay_Ms(500);
    printf("val:%04d\r\n", adc_val);
    Delay_Ms(2);
    for (int outer=0;outer<10;outer++)
    {
        for(int inner = 0; inner<1024;inner++)
        {
            if(inner<adc_val)
                GPIO_WriteBit(GPIOD,
                    GPIO_Pin_7, Bit_SET);
            else
                GPIO_WriteBit(GPIOD,
                    GPIO_Pin_7, Bit_RESET);
        }
    }
}
```

As the final step, we execute the program again. Now the brightness of the red LED can be adjusted using the potentiometer.

Documentation

It is out of the question to describe a microcontroller in full in a technical article. Although WCH focuses its in-house resources on the domestic market, the documentation department is also willing to respond to Western developers — field engineers are happy to answer questions in English in the manufacturer forum.

Those who want to search for resources on their own initiative will find what they need in the first step in the GitHub repository provided at [4]. As shown in **Figure 10**, the subfolder [5] contains a collection of examples that developers can choose from as they please.

A nice fact that usually cannot be taken for granted with Chinese microcontrollers is that complete English-language documentation is provided. The datasheet can be found at [6].

Procuring the Controller

At the time of writing, a search for the string CH32V003F4P6 in a component search engine returns almost no results. On AliExpress, WCH now offers packages of 50 of the various controllers, which — after taking shipping costs and customs into account — cost around €0.30 each when they arrive at the author's laboratory in Budapest.

In practice, it is more advisable to leave the procurement of the controller in quantities to the manufacturer. If the manufacturer is in Asia, it is usually easy to procure the controllers from WCH, which are generally always available from stock.

If this is not feasible, WCH is very cooperative even with the smallest customers. Visit the URL of the respective chip family [7] and scroll down to contact the controller manufacturer's sales team, which is generally very responsive.

A Chip to Be Aware Of

There is no question that chips such as STM32 and ESP32 are very easy to program. However, if you take a look at the BOM, you will quickly realize that the chips are anything but inexpensive, probably also due to the large die size.

The controller presented here may not be able to keep up with its established competitors in terms of "ease of use," but it is many

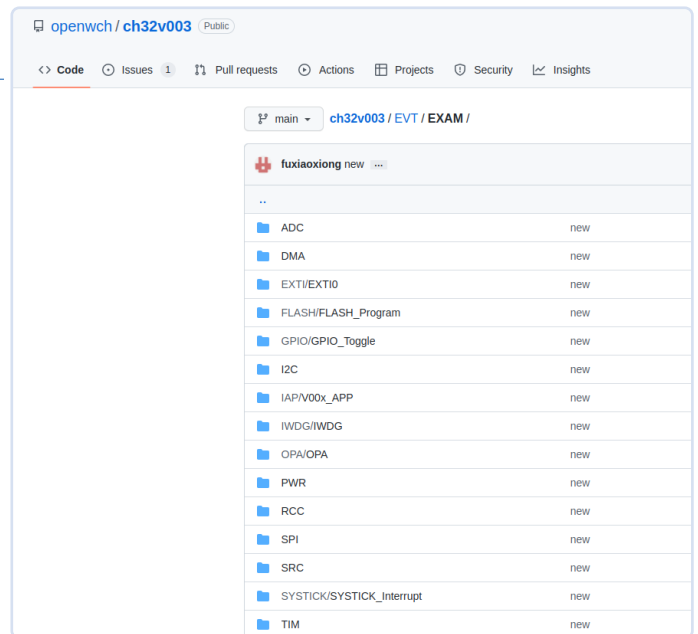


Figure 10: Comprehensive code examples accompany the core.

times cheaper. For this reason alone, it is a chip worth knowing about, especially if you are considering a larger series. ◀

Translated by Jörg Starkmuth — 250042-01

Questions or Comments?

Do you have questions or comments about this article? Email the author at tamhan@tamoggemon.com, or contact Elektor at editor@elektor.com.



About the Author

Engineer Tam Hanna has been working with electronics, computers, and software for more than 20 years; he is a freelance developer, book author, and journalist (www.instagram.com/tam.hanna). In his free time, Tam's interests include 3D printing and selling cigars.



Related Product

> **WCH CH32V307V-EVT-R1 RISC-V Development Board**
www.elektor.com/20448

WEB LINKS

- [1] CH32V003 RISC-V MCU development board: <https://www.tindie.com/products/adz1122/ch32v003-risc-v-mcu-development-board/>
- [2] WCH LinE debugger CH32V003: <https://www.tindie.com/products/adz1122/wch-lin-e-online-download-debugger-ch32v003/>
- [3] MounRiver Studio download: <http://www.mounriver.com/download>
- [4] CH32V003 GitHub repository: <https://github.com/openwch/ch32v003>
- [5] Code examples from WCH, GitHub: <https://github.com/openwch/ch32v003/tree/main/EVT/EXAM>
- [6] CH32V003 datasheet, GitHub: <https://github.com/openwch/ch32v003/tree/main/Datasheet>
- [7] CH32V003 controller, WCH: <https://www.wch-ic.com/products/CH32V003.html>

An FPGA-Based Audio Player with Equalizer (2)

Adding Volume Control, Advanced Mixing, and a Web Interface

By Dr. Christian Nöding (Germany)

We continue our series on creating an FPGA-based audio player with equalizer functionality. Here, we'll implement advanced audio processing features like equalization, dynamic compression, and crossover filtering, as well as a web interface to control it all.

In the first part [1] we covered the basics: preparing the toolchain, setting up the hardware, and introducing audio signal processing concepts. Now it's time to tackle the actual signal processing part and finish the project!

The most basic audio processing that needs to be implemented now is a volume control of the individual channels. This can be done using one or more of the 9×9 bit DSP multipliers of the FPGA, to perform a signed integer multiplication with the signal. To reduce the number of used DSP multipliers, I used an unsigned 8-bit volume signal between 0 and 255. After the multiplication, we can divide by 128 simply by shifting the result 7 bits to the right (without using any kind of DSP slices within the FPGA) to achieve a volume control between 0 % and 200 %, which allows both to attenuate strong audio signals and to amplify low-level audio signals.

From Volume Control to Mixing

There's only a small step between a simple volume control and a mixer. To mix different audio sources together, the process works just like on an analog mixing console, where each input channel has its own volume potentiometer. The user selects the desired volume level for each input signal, and then all the signals are added together to produce an output signal. This means the volume control block described in the previous paragraph can be reused. In our case, each audio input

signal is actually controlled by two volume signals: one for the left channel and one for the right. This allows for precise adjustment not only of the mix but also of the stereo balance.

With our reusable multiplication process in VHDL, we can create a multi-channel audio mixer simply by sequentially using the same multiplier unit for all audio signals. As the clock of the FPGA with at least 100 MHz is around 2000 times faster than the audio clock with 48 kHz, a lot of sequential multiplications are no problem, and the latency is around 200 ns for this block. In total, there are 22 channels: 2 channels from an external ADC, 2 from an SD Card, 2 from the S/PDIF input and 16 from the UltraNet input. Each of the 22 mono audio samples is fed into an audio mixing block, where it is multiplied with its corresponding 8-bit volume factors.

The first seven bits of each volume factor are used to control the volume of the corresponding input signal between $-\infty$ dB_{FS} (0%) and 0 dB_{FS} (100%). Using the last bit allows an additional volume gain of +6 dB_{FS} (200%) to boost weak input signals. As all samples are processed as signed integers, the resulting 44 channels (22 for the left and 22 for the right) are summed using a simple addition. In case the multiple 24-bit input signals contain high volumes (which would result in clipping and digital distortion in the sum), the output signal is processed on 36 bits to avoid clipping.

Digital or Analog?

What good is the best logic if we don't hear anything yet? So, we must find a solution to connect some speakers to our new audio mixer. If you have an Audio-Video Receiver (AVR) in your home-cinema, you can use its TOSLINK input. From the website ackspace.nl [2], I found an implementation in VHDL of an S/PDIF transmitter which outputs differential Manchester-encoded AES/EBU signals, directly fed with our 24-bit logic vectors. But in case the system driving your speakers only has analog inputs, we must find another solution.

PWM is the tool of choice for most things, but for an audio signal we would get an unsatisfactory result: at modulation frequencies



Listing 1: Computing coefficients for a peak filter.

```
double V = pow(10.0, fabs(gain)/20.0);
double K = tan(PI * frequency/48000);
double norm; // we will normalize to b0
double coeff_a[3];
double coeff_b[3];
if (gain >= 0) { // coeffs for positive gain-values
    norm = 1.0 / (1.0 + 1.0/Q * K + K * K); // norm = 1/b0
    coeff_a[0] = (1.0 + V/Q * K + K * K) * norm;
    coeff_a[1] = 2.0 * (K * K - 1.0) * norm;
    coeff_a[2] = (1.0 - V/Q * K + K * K) * norm;
    coeff_b[0] = 1.0; // normalizing to b0 reduces to 5 coeffs
    coeff_b[1] = coeff_a[1];
    coeff_b[2] = (1.0 - 1.0/Q * K + K * K) * norm;
}else{ // coeffs for negative gain-values
    norm = 1.0 / (1.0 + V/Q * K + K * K); // norm = 1/b0
    coeff_a[0] = (1.0 + 1.0/Q * K + K * K) * norm;
    coeff_a[1] = 2.0 * (K * K - 1.0) * norm;
    coeff_a[2] = (1.0 - 1.0/Q * K + K * K) * norm;
    coeff_b[0] = 1.0; // normalizing to b0 reduces to 5 coeffs
    coeff_b[1] = coeff_a[1];
    coeff_b[2] = (1.0 - V/Q * K + K * K) * norm;
}
```

around 48 kHz, the resolution is reduced to only a few bits. To overcome this, the triangle generator needed for PWM generation would need to run at about 805 GHz! We took some effort to keep audio samples at 24-bit – we are not prepared to give that up now. An external I²S DAC could be a solution; ICs like the PCM5102A for example are available for under 5 €. While reading audio articles, however, one term kept appearing repeatedly: Direct Stream Digital (DSD). If the articles are to be believed, DSD is supposed to have extraordinary audio qualities. DSD is using a Pulse Density Modulation (PDM). While PCM audio transmits clocked audio samples, PDM transmits a continuous audio stream.

For this, PDM uses a high-frequency base-clock in the range of several MHz. For 48 kHz audio we can use 100x oversampling with 4.8 MHz, a frequency an FPGA can deal with easily. From [3] we can get a PDM modulator in VHDL which takes a logic-vector with a 24-bit signed audio sample and a clock signal. As output, we will get a pulse-density modulated signal. The charm of this solution is the easy conversion to a real analog signal: a simple low-pass filter. Using the cut-off frequency of 48 kHz and choosing a resistance of 10 kΩ as a target, we can calculate the capacitance with the following equation:

$$C_{LP} = \frac{1}{2 \cdot \pi \cdot R \cdot f_c} = \frac{1}{2 \cdot \pi \cdot 10k\Omega \cdot 48kHz} \approx 330pF$$

330 pF is a standard value for ceramic capacitors. From now on, our Arduino board can generate one or more nice and clean audio signals, and that for a couple of cents instead of the price for an I²S DAC. Wonderful!

More Bass!

If your audio file or the live input of the ADC sounds a bit flat, you might want to add some equalizing functions. As our FPGA has thousands of spare logic elements, we can add a parametric equalizer. A quick search on the Internet reveals that IIR filters are quite universal. The IIR equation uses a few coefficients, needs a storage for previous input and output samples and uses mainly multiplications:

$$y[n] = \frac{1}{b_0} (a_0 \cdot x[0] + a_1 \cdot x[-1] + a_2 \cdot x[-2] - b_1 \cdot y[-1] - b_2 \cdot y[-2])$$

A diagram explaining the coefficient in brackets on the right is given in **Figure 1**. The division by b_0 can be moved to the microcontroller by dividing all other coefficients by b_0 . Two or more of these filters can be put in series so that we are able to adjust our audio-sample at several frequencies. With the 6 coefficients a_x and b_x we can control the behavior of the filter. The online tool "Biquad calculator v3" [4] next to the *Audio EQ Cookbook* [5] can be used to calculate all coefficients to form a low-pass, high-pass, peak or shelf filter with a single difference equation. Calculating a general Peak-EQ with frequency, quality

factor and gain control, the microcontroller calculates the coefficients, as shown in **Listing 1**.

As b_0 equals 1, there are 5 remaining coefficients (most of them are far below 1.0) to be transmitted to the FPGA by UART. There we must calculate the equation in real time for each audio sample. But wait: our FPGA has no floating-point unit and cannot compute that with the required resolution... so we must think around the corner and use a simpler math. A solution is to multiply all coefficients with numbers in base 2, do some integer multiplication in the FPGA and divide using bit shifting at the end – the so called fixed-point arithmetic. For conventional parametric equalizers it is sufficient to use Q30 numbers [6], meaning we have 30 bits reserved for the fractional part leaving two bits for the integer part. So, multiplying with $2^{30} = 1,073,741,824$ will do the job.

The Cyclone 10CL016 on the Arduino Vidor 4000 is a quite small FPGA with only 112 9x9 bit DSP multipliers. As each IIR filter needs a multiplication of our 24 to 36-bit audio signal with five 32-bit coefficients, our DSP slices in the FPGA would not be sufficient for multiple equalizers.

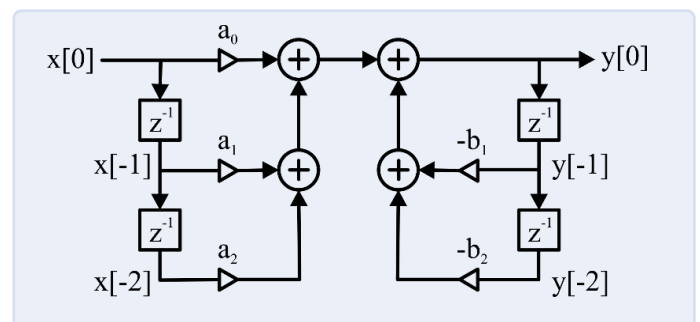


Figure 1: IIR-filtering (right hand term of equation 2) explained visually.

Following the volume control of the audio signal previously in this article, we can reuse a single multiplication process several times to save DSP slices. Using a pipelined VHDL structure we can multiply the coefficients using the same multiplication process five times. Again, as the FPGA is using a 100-MHz clock, each IIR filter is calculated within 130 ns here, while a new audio sample is received after 20.83 μ s — we have plenty of time.

The calculation then is straightforward: looking at the equation of the IIR filter, we calculate a fixed-point multiplication of the feedforward signals (i.e., multiplying the incoming signals "x" with the "a" coefficients), shift 30 bits to left, build the sum, calculate multiplication of feedback signals, (i.e., multiplying the output signals "y" with the "b" coefficients), shift 30 bits to right and subtract the signals. After thirteen clock cycles, we obtain the result of the IIR filter, and we can reset the sequence. This means, each filter adds another 130 ns to the signal. In the final project, our Cyclone 10 has enough free logic elements for up to five stereo EQs.

Let's Rumble

As a trumpet player in a big band, I like some good and low bass. So, I pondered how to integrate a subwoofer to the system. I was aware of several kind of crossover filters, and Elektor had already shown some great active audio crossover filters in the past. Up to now, we have a great audio device, which means, we shouldn't make any concessions, right? A 4th-order Linkwitz-Riley crossover with a 24 dB/octave slope to separate signals between subwoofers and tweeters should be good enough for my expectations. The main advantages of such sharp filters are a 360° phase shift to stay in phase and the absence of any frequency boost at the crossover point – after all, we want to do the equalizing intentionally via our fresh PEQ section.

The calculation at the FPGA side follows the calculation for the IIR filter, except that the number of coefficients has been doubled: five coefficients for feedforward and five for the feedback signals. We also need to increase the accuracy of our fixed-point calculation for this kind of filter to at least Q35, to allow crossover-frequencies below 100 Hz.

As Q35 is above typical 32-bit values, we must use a 64-bit signed integer within the Arduino code as well. I'm using a 64-bit union type for dual use as double and 64-bit signed integer directly (**Listing 2**).

To test the implemented filters, I measured the input-output-characteristics with a Bode 100 Vector Network Analyzer using different EQ settings. The analyzer generates a frequency sweep over a desired range. Here, I took frequencies between 10 Hz up to 100 kHz. As we are using an audio sample rate of 48 kHz, we do not expect any signal above 24 kHz and that's exactly what we can see in **Figure 2**.

The graph also shows the effect of peak, bandpass and notch filters as well as high-pass and a Low-Shelf-Filter with different gains. As you can see, we can achieve the desired frequency responses. In general, we get a signal-to-noise ratio of at least 60 dB which is quite good, but I could spend some more effort on the analog part of the circuit. On the other hand, we can also get some noise under control digitally.

Getting Dynamics and Noise Under Control

Static audio mixing is one thing, but what if your source has some surprising loud pops and hisses? To get these nasty things under control, dynamic compressors and noise gates have been developed in the past. The basic function of a compressor is to reduce the volume when a threshold value is exceeded. So, we need something to compute and control the gain, as well as a time-controller for adjusting attack and release. The calculation for the gain looks like this:

$$gain_{target} = \frac{\frac{|sample| - threshold}{ratio} + threshold}{|sample|}$$

For implementing in VHDL, we must consider that we need two divisions. Unfortunately, divisions require lot of logic if we just use the "/" operator. To fit a three-way audio compressor into the remaining logic elements, I used a simple bit-shifting for calculating the ratio in fixed base 2 steps (2:1, 4:1, 8:1, etc.) and for a better audio performance an SRT division algorithm (or Radix 2 division) [7] for the calculation of the gain itself. The final compressor takes 1244 LEs and 4 multipliers.

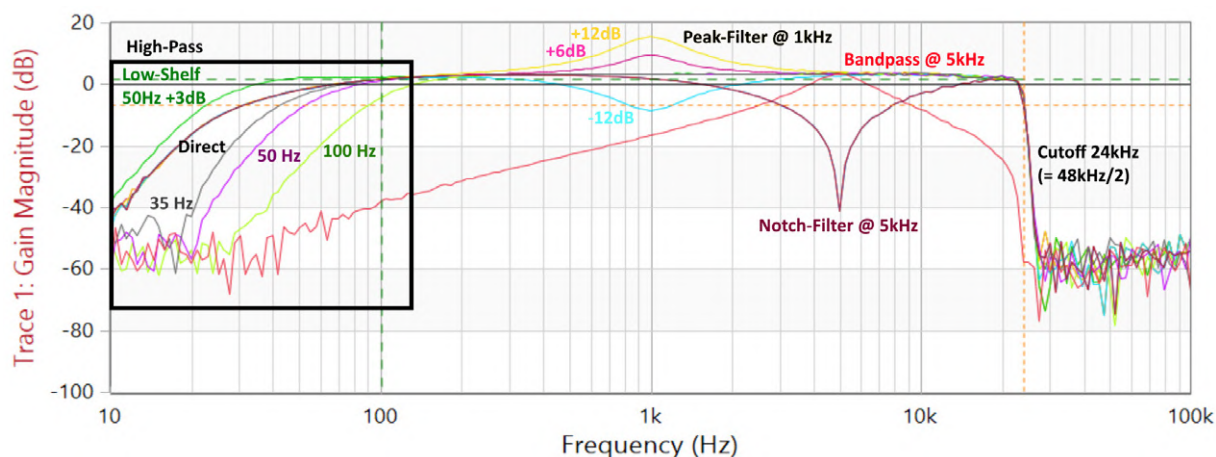


Figure 2: Frequency response of different filter settings.



Listing 2: Computing coefficients for a Linkwitz–Riley filter.

```
double wc = 2.0 * PI * frequency;
double wc2 = wc * wc;
double wc3 = wc2 * wc;
double wc4 = wc2 * wc2;
double k = wc / tan(PI * (frequency/48000));
double k2 = k * k;
double k3 = k2 * k;
double k4 = k2 * k2;
double sq_tmp1 = sqrt(2.0) * wc3 * k;
double sq_tmp2 = sqrt(2.0) * wc * k3;
double a_tmp = 4.0 * wc2 * k2 + 2.0 * sq_tmp1 + k4 + 2.0 * sq_tmp2 + wc4;
union_64b coeff_a[5];
union_64b coeff_b[5];
if (isHighpass) {
    // coefficients for HighPass-Filter
    coeff_a[0].d = k4 / a_tmp;
    coeff_a[1].d = -4.0 * k4 / a_tmp;
    coeff_a[2].d = 6.0 * k4 / a_tmp;
    coeff_a[3].d = a[1].d;
    coeff_a[4].d = a[0].d;
} else {
    // coefficients for LowPass-Filter
    coeff_a[0].d = wc4 / a_tmp;
    coeff_a[1].d = 4.0 * wc4 / a_tmp;
    coeff_a[2].d = 6.0 * wc4 / a_tmp;
    coeff_a[3].d = a[1].d;
    coeff_a[4].d = a[0].d;
}
coeff_b[1].d = (4.0 * (wc4 + sq_tmp1 - k4 - sq_tmp2)) / a_tmp;
coeff_b[2].d = (6.0 * wc4 - 8.0 * wc2 * k2 + 6.0 * k4) / a_tmp;
coeff_b[3].d = (4.0 * (wc4 - sq_tmp1 + sq_tmp2 - k4)) / a_tmp;
coeff_b[4].d = (k4 - 2.0 * sq_tmp1 + wc4 - 2.0 * sq_tmp2 + 4.0 * wc2 * k2) / a_tmp;
// convert to Q35-format
for (int i=0; i<5; i++) {
    coeff_a[i].s64 = coeff_a[i].d * 34359738367;
    coeff_b[i].s64 = coeff_b[i].d * 34359738367;
}
```

Now, when a signal has a level above the threshold, the gain-computer will reduce the gain depending on the amount of overshoot and the selected ratio. The attack and release timing is realized via a first order low-pass filter with a switched coefficient α depending on the current state (attack = α_A or release = α_R):

$$\text{gain}[n] = (\alpha_x \cdot \text{gain}[n-1]) + (1 - \alpha_x) \cdot \text{gain}_{\text{target}}$$

The switching between the coefficients and the general control of the effect is done using a small state machine with only five states: *s_Idle*, *s_Attack*, *s_Active*, *s_Hold* and *s_Release*. While attacking and releasing is done using the low-pass filter, for the hold a simple counter with precalculated number of audio samples based on the audio sample rate is used.

In the end, a noise gate is very similar, as it has an almost identical

structure. Only the direction for comparing the signals and the gain is different: $\text{gain}_{\text{closed}} = 0\% \dots 90\%$ for a closed gate and $\text{gain}_{\text{opened}} = 100\%$ when opened. The timing for attacking and releasing the gate can be done using the same low-pass filter approach as the compression.

Final FPGA Structure and Practical Build

The FPGA has its desired structure to handle all kind of different digital audio data in real time. Handling data with logic vectors should avoid problems with slightly different clock rates if all systems are using 48 kHz as sample rate. Certainly, this is far from a final product you would buy for a professional event, but it is a solid opener into this world.

Figure 3 shows the final structure with all implemented blocks for conversion, volume control and equalization. Within Quartus you can add or remove single functions within the graphical interface of the block diagram. Depending on your machine the synthesizing process will take between 3 to 8 minutes until a new bitstream file is created.

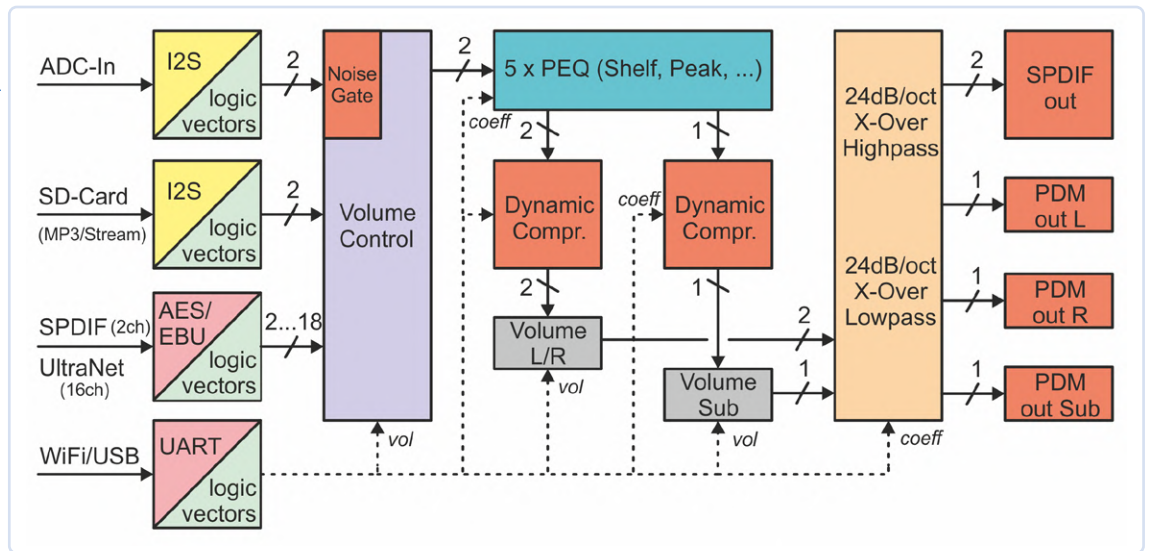


Figure 3: Overview of the whole FPGA-structure.

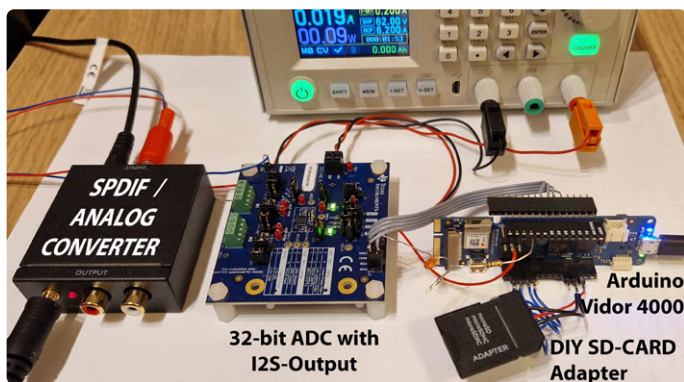


Figure 4: The test setup. Here a small S/PDIF-to-Analog converter was used to make the output audible.

For testing, all the modules (S/PDIF converter, ADC and the Arduino Vidor 4000) were connected together using jumper wires. A picture of the setup is shown in **Figure 4**.

(Mobile) Controlling

Now that the FPGA is doing its job, we can get back to the NINA W102 module again. First, we must find a solution to update the firmware of the ESP32 chip inside the module. Luckily, ESP32 chips have a serial bootloader that can be activated using a special GPIO together with the RESET signal. The FPGA tunnels these pins to the SAMD21, that parses incoming ASCII commands from the USB connection. Commands starting with `samd` will be processed by the SAMD. All other commands will be passed to the ESP32/NINA. With this command structure, this system can be reused for your own projects quite easily. Sending the following command triggers the bootloader of the ESP32 and passes all USB-communication through the FPGA to the ESP32:

```
samd:update:nina
```

From now on, we can use the usual Arduino toolchain for the ESP32 or the Python script `esptool.py` directly to upload new code to the ESP32 controller. A press on the reset button of the MKR board or a power cycle terminates the upload mode.

The implemented webserver (see part 1 [1]) offers several functions to control the whole system (see **Figure 5**). Next to general playback

controls you can change the mode and parameters for all five PEQs, the crossover and the dynamic processing in real time. A nice feature: a JavaScript function calculates the frequency response and shows the current state of the five PEQs connected in series in one diagram. One disadvantage of the NINA W102 is the relatively small 2 MB of Flash memory available, where 1.9 MB are reserved for the application and 190 kB for SPIFFS (using the "Minimal" ESP32 partition scheme). Thus, we cannot use the internal flash for storage, but we must load all files from an SD-card. Unfortunately, there is no SD card holder on the Arduino Vidor 4000. I used an old MicroSD card adapter that I had and soldered wires to it, as you can see in the **Figure 4**.

A function within the JavaScript file `src/ajaxrfrsh.js` is polling update-data from the ESP32 by calling the page `/json`. The C-code is then collecting all information and will return the data to the client in JSON-format which will be parsed and put to the individual HTML-elements using `document.getElementById()`. On loading the main-page two other functions are called: `loadTitles()` and `initEQ()`. `loadTitles()` from the file `src/player.js` calls the page `/toc` that instructs the ESP32 to call the internal C function `SD_getT0C()` that crawls through the SD card and collects all playable audio files (the file formats currently supported are MP3 and WAV) and sends the titles together with their size back to the client device. `initEQ()` from `src/eq.js` finally creates the canvas for the EQ diagram using the component `Plotly` from the file `src/plotly.js`. On each change of one of the EQ parameters, the functions `setPEQ()` and `updateEQ()` are called to send values to the ESP32 and update the EQ response diagram.

As we have three devices that allow bidirectional communication, I have implemented some command-interpreter to the system; so it is also possible to control the device from a simple terminal program on a PC or another microcontroller. In normal mode, all ASCII-based commands sent from a computer to the SAMD21 will be forwarded to the ESP32. There are commands for mixing and playback, and configuring the parameters like volume and filter coefficients. See the **Command Overview** text frame.

Full-featured Digital Audio Mixing Console

The final device is impressive: it mixes multiple channels of 24-bit digital audio with a 5-band parametric stereo equalizer, a noise gate, dynamic compression, and an LR24 crossover for a 2.1 speaker-system with analog and digital output — everything on a PCB of only 20 cm²

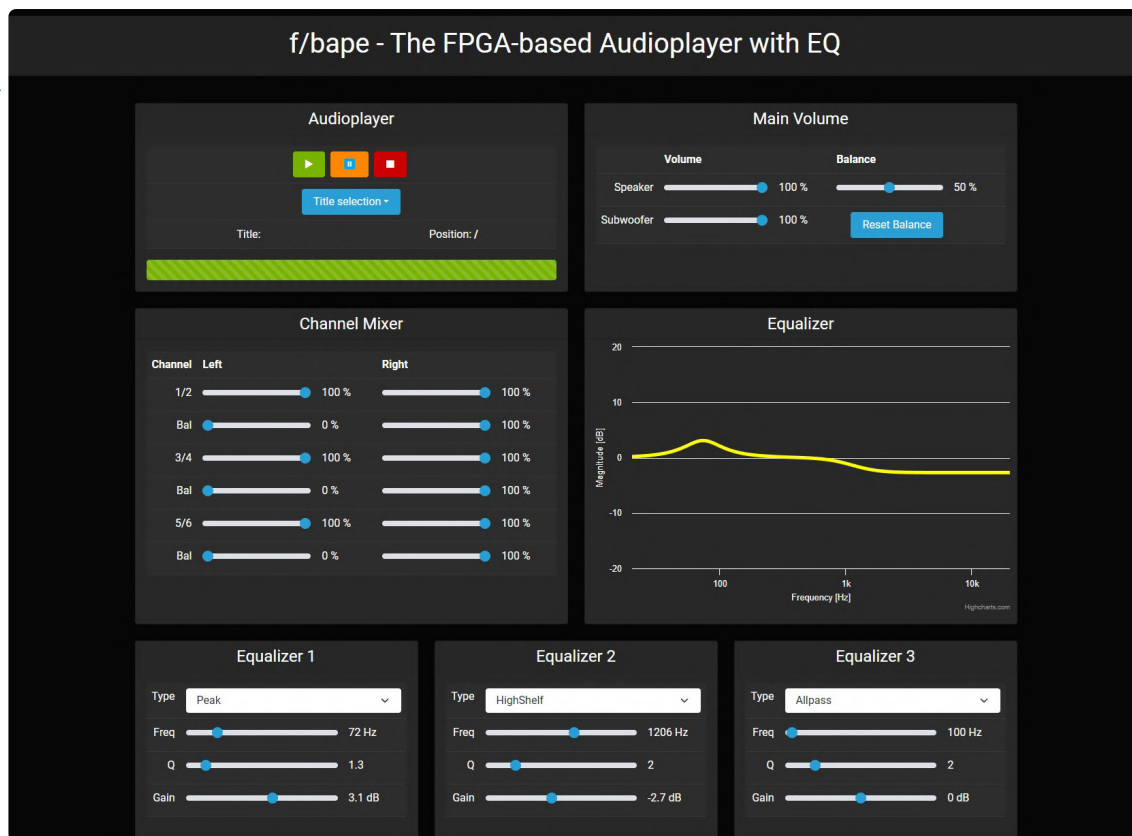


Figure 5: Web interface with player-controls and equalizer.

of surface area! Dealing with audio and an FPGA is a great opportunity to get in touch with digital signal processing in general, having fun at the same time as you can hear the results directly while learning.

In the process, I learned that dealing with fixed-point integers entails the problem that your data can clip. That led me to the solution: the audio mixer should be processing the audio internally on 36 bits, while

Command Overview

In normal mode, all ASCII-based commands sent from a computer to the SAMD21 will be forwarded to the ESP32. But if you prefix with `samd:`, the device will try to interpret the command itself.

Here is a selection of the most important commands for mixing and playback, executed by the ESP32 of the NINA SoC:

- › Play a desired file:
`player:file@filename.mp3`
- › Play a desired network-stream:
`player:stream@http://stream.m3u`
- › Play/pause/stop current file (or stream):
`player:play / ...:pause / ...:stop`
- › Request current file:
`player:currentfile?`
- › Request current position:
`player:position?`
- › Set the main or subwoofer volume between -140 dB and +6 dB:
`mixer:volume:main@-12`
`mixer:volume:sub@-5.8`

- › Set volume of channel 4 between -140 dB and +6 dB:
`mixer:volume:ch4@-8.5`
- › Set balance of channel 6 between left (0), center (50) and right (100):
`mixer:balance:ch6@25`
- › Set PEQ #2 to type "Peak" at 250 Hz with Quality 2.1 and a gain of +19.2 dB:
`mixer:eq:peq2@1,250,2.1,19.2`
- › Set the low or high-pass to 95 Hz:
`mixer:eq:lp@95`
`mixer:eq:hp@95`
- › Set the noise-gate to -60 dB with 30 dB range:
`mixer:gate1@-60,40,10,50,258`
- › Set the compressor 2 to -20 dB, ratio 4:1, +6 dB makeup:
`mixer:comp2@-20,4,6,10,10,151`

There are more commands for controlling the player position, the Wi-Fi configuration, or other parts of the system. Have a look into the file *Communication.ino* within the NINA sketch folder, or see the documentation itself [11].

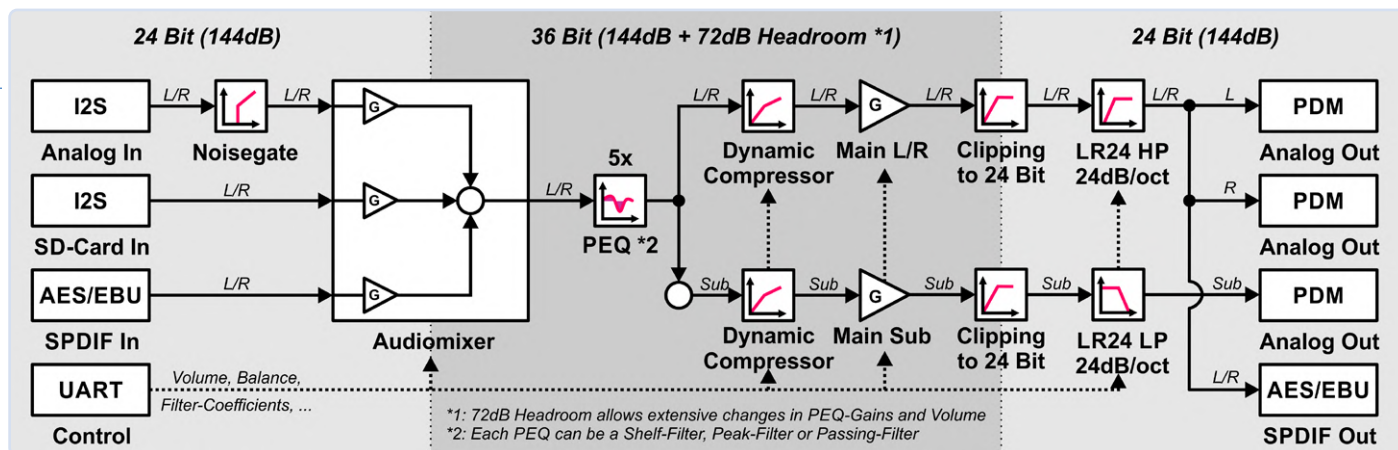


Figure 6: Overview of an alternate design using 36-bit vectors.

keeping 9-bit DSP-multiplicators in mind. Of course, this would use much more logic. An overview of this alternate design using 36-bit vectors is shown in **Figure 6**.

At the end, 94% of the available logic elements are used, leaving 854 LEs available for some more stuff. The achieved experiences regarding filters can easily be transferred to video or other types of signal-processing. But the best thing is the simple, frustration-free, and expandable realization using the Arduino MKR-boards to get a full-featured digital audio mixing console in HD-quality. You can find the whole project with all sources on Github [8]. This FPGA-based audio player can serve as a basis for other follow-up projects. For example, together with students of the University of Kassel we built a fully digital Class-D amplifier [9]. I also made a DIY expansion card for the Behringer X32, adding 32 extra channels to create a 64-channel mixing console [10]. These two projects build on the concepts covered here and showcase the versatility of this FPGA audio platform for advanced audio applications. ◀

230632-B-01

Questions or Comments?

Do you have questions or comments about this article? Feel free to contact the author at christian@noeding-online.de, or contact Elektor at editor@elektor.com.



About the Author

Dr.-Ing. Christian Nöding studied electrical engineering at the University of Kassel from 2003 to 2009. He then began working at the department of Power Electronics at the University of Kassel and completed his PhD in 2016. His main interests lie in the design and control of electronic power converters for decentralized energy supply, stage and lighting, as well as music and digital audio technology.



Related Products

- ▶ **Dogan and Ahmet Ibrahim, *Practical Audio DSP Projects with the ESP32* (Elektor 2023)**
www.elektor.com/20558
- ▶ **Elektor Audio Collection (USB Stick)**
www.elektor.com/19892



WEB LINKS

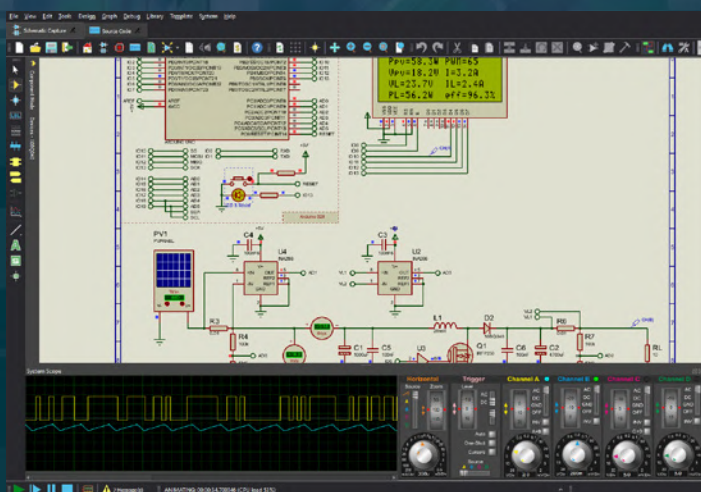
- [1] Christian Nöding, "An FPGA-based Audio Player with Equalizer," Elektor 3-4/2025: <https://www.elektor.com/230632-01>
- [2] An S/PDIF transmitter in VHDL, ACK.space: https://ackspace.nl/wiki/SP/DIF_transmitter_project
- [3] A PDM modulator in VHDL, FreeCores (GitHub): https://github.com/freecores/sigma_delta_dac_dual_loop
- [4] Earlevel Engineering, Filter Calculator: <https://www.earlevel.com/main/2021/09/02/biquad-calculator-v3>
- [5] Robert Bristow-Johnson, Audio EQ Cookbook: <https://webaudio.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html>
- [6] Q (number format) on Wikipedia: [https://en.wikipedia.org/wiki/Q_\(number_format\)](https://en.wikipedia.org/wiki/Q_(number_format))
- [7] SRT Division Algorithm: <http://www.lothar-miller.de/s9y/categories/24-Rechnen>
- [8] Author's GitHub repository: <https://www.github.com/xn--nding-jua>
- [9] Fully digital Class-D amplifier: https://www.youtube.com/watch?v=2TJE_azlhkl
- [10] DIY expansion card for the Behringer X32: <https://github.com/xn--nding-jua/xfbape>
- [11] Audioplayer Command Documentation, GitHub:
<https://github.com/xn--nding-jua/Audioplayer/blob/main/Documentation/Commands.xlsx>



**Brand new, high performance,
64-bit application framework
with full support for international
character sets and theming.**

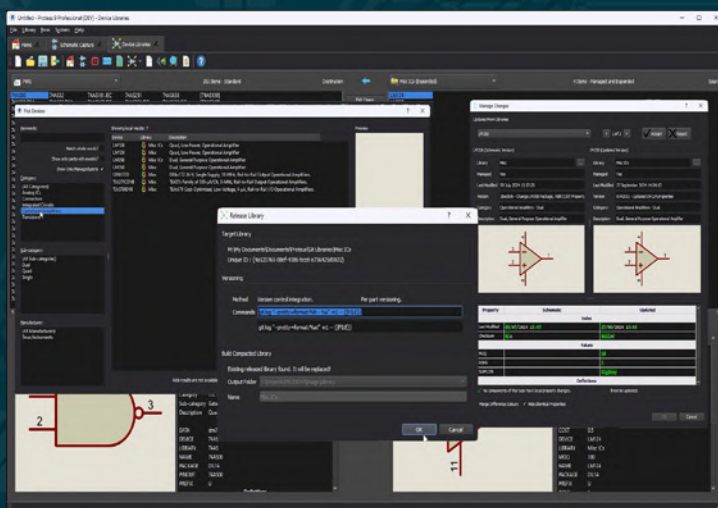
PROTEUS 9

...THE START OF A NEW ERA



Proteus VSM Simulation

- System Scope with live circuit probing.
- Live results on graph based simulations.
- Visual Designer for MicroPython.
- IoT Builder for MicroPython.



Proteus PCB Design

- Versioning of managed libraries with either repository or local folder.
- Software wide support for restricting use to managed libraries.
- Lightning fast, live power plane regeneration.
- Via Inspector tool for board analysis.



Labcenter  **www.labcenter.com**
Electronics
Tel: +44 (0)1756 753440

You design. We deliver.

The Newest Products for Your Newest Designs®



[mouser.com/new](https://www.mouser.com/new)



MOUSER
ELECTRONICS