£10.95 | €13.00 (DE) | €14.90 (other EU countries) | $13.95

elektor MEDIA & LEARNING | MAG

25009 * SINCE 1961 *

CIRCUIT SPECIAL 2025
ELEKTORMAGAZINE.COM

SPECIAL EDITION

CIRCUIT SPECIAL 2025

Design Ideas,
Tips & Tricks

Packed with Detailed Projects

- ESP32 Audio Transceiver Board
- Neon Lamp Dice
- DDS Generator with ATtiny
- Capacitance Meter
- Solid-State Foot Warmer
- Turntable Speed Calibrator
- USB Measurement Adapter
- Small Audio Mixer
- Parking Sensor Tester
- Vacuum Cleaner Automatic Control
- Infrared Remote-Controlled Dimmer
- Lamp Audio Amplifier

... and much more!

COLOPHON

Volume 51, No. 25009
August & September 2025
ISSN 0932-5468

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

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

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

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

EDITORIAL

Jens Nickel

International Editor-in-Chief, Elektor Magazine

Open Doors

Have you ever developed something that should be more than a loosely wired prototype that (only) does what it is supposed to do in your little room? Did you need to think about cost-saving, readily available components, easy-to-build hardware, and maintainable software? Did you consider robust connectors, an inexpensive enclosure, and drilling templates? Or a stock of spare parts and the necessary tools for real-world use? Solving the many challenges is time-consuming, but also a lot of fun. And perhaps, like me, you're be forced to acquire new skills. In my case, new skills included board design with KiCad, purchasing PCBs, simple metalworking and streaming audio.

I can only encourage you to walk through doors that are wide open. There have never been so many possibilities for developers, so many free tools, and so many really cheap modules and components. In addition, there are lots of video tutorials, well-documented code libraries and, last but not least, a wealth of well-described projects on the internet from which you can learn new tricks. One such place where you can find many projects on electronics topics such as power supplies, automation, audio, and the like, is on our Elektor Labs platform. Over the last few months, my colleagues and I have again selected and edited the best circuit submissions from Elektor Labs and via email, and prepared them for you. We've created clearly comprehensible circuit diagrams in the traditional Elektor style so that you can be even more inspired by the projects. You will find our selection in this issue.

There is not enough space here for a list of my favorite projects in this issue, but some examples include the simple but clever solution that allows power banks to be connected in parallel, or the DDS generator with ATtiny. I also like the automatic vacuum cleaner on-switch from Giuseppe la Rosa from our Italian community. Then there's a small audio mixer, an amplifier that works with a halogen lamp, and a USB adapter for measuring current. Just take a look for 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

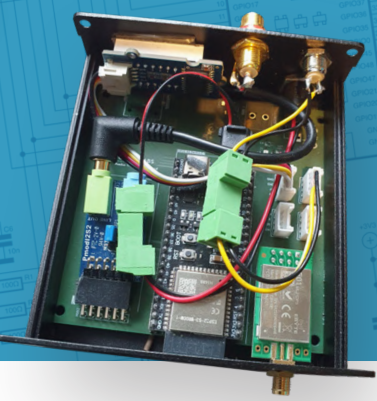
International Editor-in-Chief: Jens Nickel | **Content Director:** C. J. Abate | **International Editorial Staff:** Hans Adams, Asma Adhimi, Mahy Arafa, Roberto Armani, Eric Bogers, Jan Buiting, Rolf Gerstendorf (RG), Ton Giesberts, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-Francois Simon, Jörg Starkmuth, Clemens Valens, Brian Tristam 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

NEON Lamp Dice

69

ESP32 Audio Transceiver Board (Part 1)

SD Card WAV File Player Demo



84

Regulars & Features

BONUS CONTENT

Check out the free Circuit Special bonus edition of Elektor Mag!

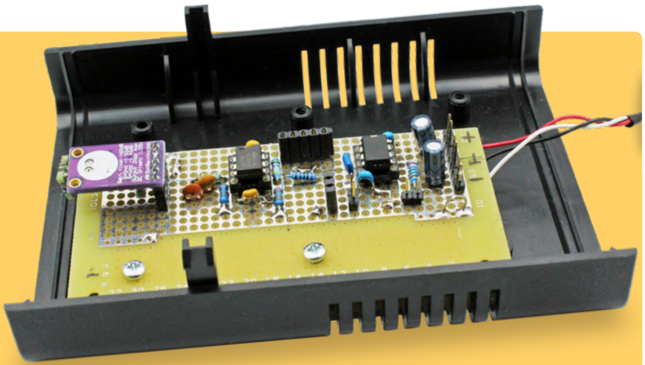
- NTP Clock: Always On Time, You'll Never Be Late!
- DHT11-Based Hygrostat: A Multi-Purpose Moisture Controller
- Easy I²C Shutdown
- Connector-Programmed Power Supply

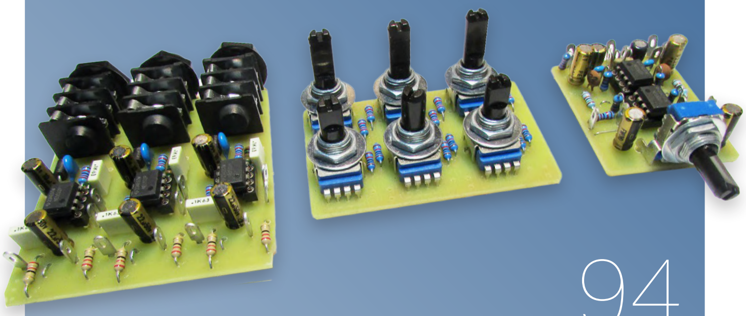
And much more!

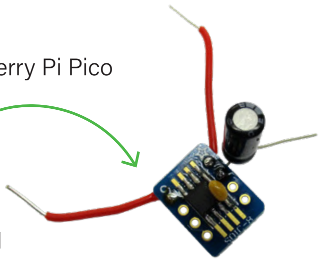
www.elektormagazine.com/circuits-and-circuit-design

Projects

Volume 51, No. 25009 / 539
August & September 2025







Immerse Yourself in Circuits & Circuit Design

Visit Elektor's Circuits & Circuit Design page for projects, videos, and tutorials!

www.elektormagazine.com/circuits-and-circuit-design

Next Editions

Elektor Magazine September & October 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 Wireless & Communication.

- Navigating Wireless Protocols
- Accurate Positioning with Bluetooth
- Satellite Tracking Using LoRa
- Wireless Audio Transmission
- AM Transmitter
- Performance Tests with the RP2350
- Model Car Remote Control with an ESP32
- Raspberry Pi Zero Web Streaming Camera

Elektor Magazine's September & October 2025 edition will be published around **September 10, 2025**.

Arrival of printed copies for Elektor Gold members is dependent on shipping times.

01 USB Measurement Adapter

Testing Current and Signal Quality of USB Ports

By Alfred Rosenkränzer (Germany)

When peripherals connected to a PC do not function or behave erratically, a thorough fault analysis of the USB port — beyond simple replugging or trying different PCs — often proves more revealing. The compact adapter introduced here allows you to insert an oscilloscope probe into the USB connection. This enables both current measurement and visualization of USB communication.

This article presents a compact, practical tool for measurements on USB 2.0 connections. Key features include:

- Static and dynamic measurement of the supply current for connected devices.
- Peripherals can be powered directly via the measurement adapter.
- Straightforward signal quality measurement of high-speed data lines.
- Two LEDs indicate the presence of a 5-V supply from both the PC and the peripheral device.

Circuit Description

Figure 1 shows the relatively simple circuit, and **Figure 2** depicts the compact layout of the adapter PCB, with layout files available online [1]. The populated board in **Figure 3** is missing diode D1. The PC connects to J1, a USB type B socket, using a USB-A-to-USB-B cable. The peripheral device connects to J2, a USB type A socket.

Current flow is detected as a voltage drop across the shunt resistor R3 via connector JP3. Static current measurements can be taken using a multimeter (typically in the millivolt range). For detecting rapid current

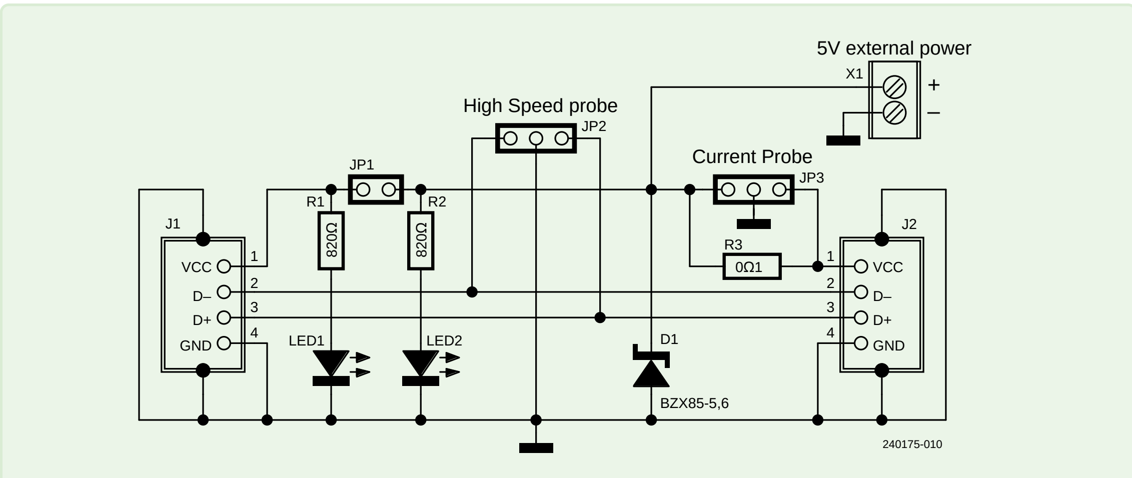


Figure 1: The USB measurement adapter circuit is relatively straightforward.

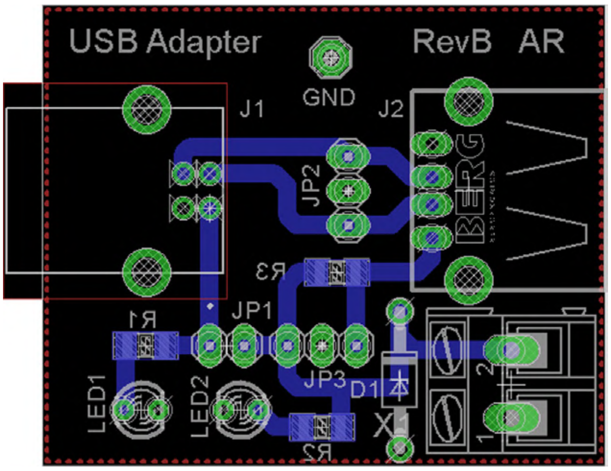


Figure 2: The compact PCB layout.

Component List

Resistors
(all SMD 1206)
R1, R2 = 820 Ω
R3 = 0.1 Ω

Semiconductors
D1 = Zener diode, 5.5 V, e.g., BZX85-5.6
LED1, LED2 = LED, 3 mm

Miscellaneous
J1 = USB-B socket, PCB mount
J2 = USB-A socket, PCB mount
JP1 = 2-pin jumper
JP2, JP3 = 3-pin header, 0.1″ pitch
X1 = 2-pin screw terminal, 5 mm pitch
PCB

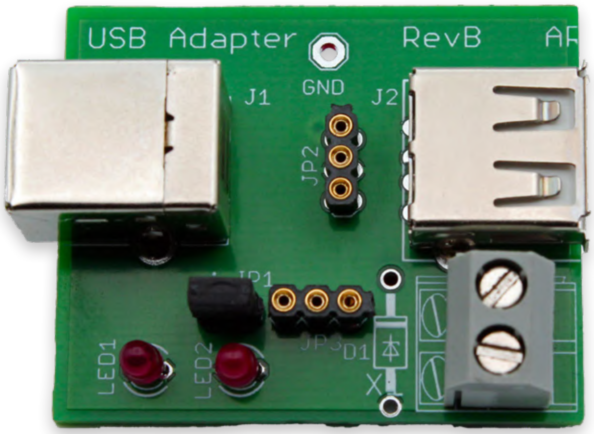


Figure 3: The populated USB measurement adapter board.

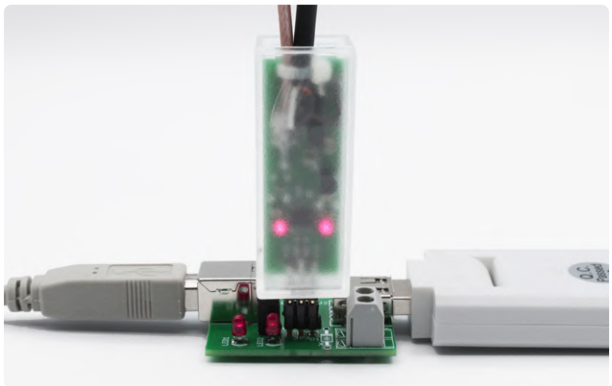


Figure 4: The USB adapter with connected current probe.

changes, an oscilloscope should be connected to JP3 — ideally with a dedicated current probe. Two variants of such probes, enabling even galvanically isolated measurements of fast signals, have been detailed previously in Elektor [2][3]. The pinout of connector JP3 pinout allows direct attachment of these probes. **Figure 4** shows the USB measurement adapter with the Current Probe 2.0 connected. **Figure 5** illustrates the current profile of a USB Wi-Fi dongle that frequently draws high currents, often causing weaker 5-V plug-in power supplies to shut down.

Jumper JP1 allows for disconnecting the 5-V power line between the host (J1) and the peripheral (J2). In such cases, external power can be supplied to the peripheral through connector X1 — ideal when the peripheral's current demand exceeds what the PC's USB port can deliver. USB 2.0 ports are limited to 500 mA, which may already cause instability in some peripherals.

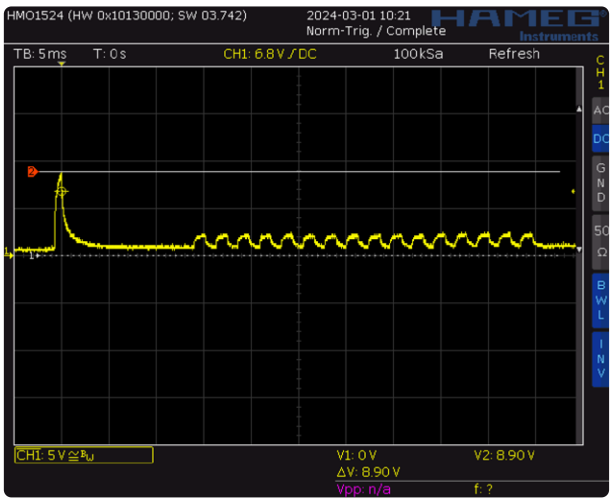


Figure 5: Current profile of a peripheral device with pulsed high current draw.

Figure 6: The USB adapter with a high-speed differential probe connected.

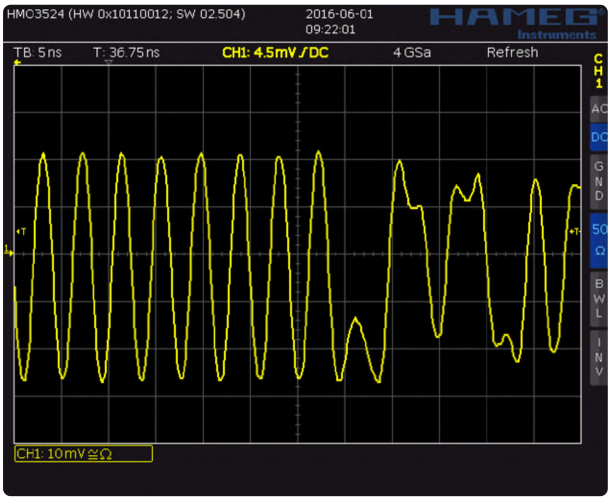


Figure 7: USB data signal waveform.

Zener diode D1 not only limits overvoltage (possibly by sacrificing itself) but also offers reverse polarity protection through its antiparallel diode. However, this is only a basic, temporary protection measure. Special care must be taken when connecting a lab power supply to ensure its output voltage remains below D1's Zener voltage. LED1 and LED2 indicate the presence of 5-V power on the host and device sides, respectively.

Data Signals

To evaluate data signals and their integrity, a high-speed oscilloscope with a suitable probe can be connected to JP2. A compatible active, differential high-speed probe has also been described previously in Elektor [4]. **Figure 6** shows this probe attached to the USB measurement adapter. **Figure 7** displays the observed data signals on a 350-MHz oscilloscope.

Naturally, the USB measurement adapter is not limited to USB 2.0; it can also be used to analyze slower peripherals, supporting USB 1.0 and 1.1 signals. The author still has a limited number of unpopulated boards available.

240175-01

Questions or Comments?

If you have any questions or comments about this article, email the author at alfred_rosenkraenzer@gmx.de or contact Elektor at editor@elektor.com.

WEB LINKS

[1] PCB Layout USB Measurement Adapter: https://www.elektormagazine.com/labs/usb-messadapter
[2] Alfred Rosenkränzer, "A Simple Current Sensor Probe," Elektor 9-10/2016: https://www.elektormagazine.com/magazine/elektor-201609/39807
[3] Alfred Rosenkränzer, "Differential Oscilloscope Current Probe 2.0," Elektor 11-12/2020: https://www.elektormagazine.com/magazine/elektor-159/59112
[4] Alfred Rosenkränzer, "Active Differential Probe (V2)," Elektor 3-4/2017: https://www.elektormagazine.com/magazine/elektor-201703/40232

02 4...20 mA Current Output for Arduino Uno

A Reliable, EMI-Insensitive Current Loop Interface

By Giovanni Carrera (Italy)

Current signals at 4...20 mA are frequently used to send process signals to controllers in industrial applications. Compared to an analog voltage signal or digital communication, the current loop offers many advantages. This project allows it to be obtained from a PWM signal, which can be generated by any microcontroller.

The purpose of this project is to provide a 4...20 mA output from a PWM signal generated by an ATmega328 microcontroller (and numerous other chips, such as the PICs). One of the more interesting applications of this circuit would be to replace or to realize a smart sensor with Arduino.

The 4...20 mA standard [1] was introduced a few decades ago, to replace the existing 3...15 psi pneumatic standard for controlling remote actuators. It soon became quite popular for its versatility and for the inherent robustness to electrical disturbances. The current loop between the transmitter and receiver is usually supplied by an external 24...30 V DC power source.

Analog Output

In this project, we use a microcontroller to create a controllable DC voltage; with some analog circuitry we can realize a standard 4...20 mA analog output. However, the ATmega328 chip on the Arduino UNO does not have a DAC to create a DC voltage. The easiest way is to use an available PWM output and filter the generated signal with a passive RC filter, to obtain an analog signal whose amplitude is proportional to the duration of the pulses.

This expedient creates a considerable noise, due to the frequency of the PWM itself. To eliminate this noise, I used a second-order, active low-pass Sallen-key type filter. The frequency of the Arduino PWM (with a 16 MHz clock) on pin 9 is about 490 Hz, so I used a very low cutoff frequency value of 11 Hz, but with enough bandwidth for the majority of industrial controls.

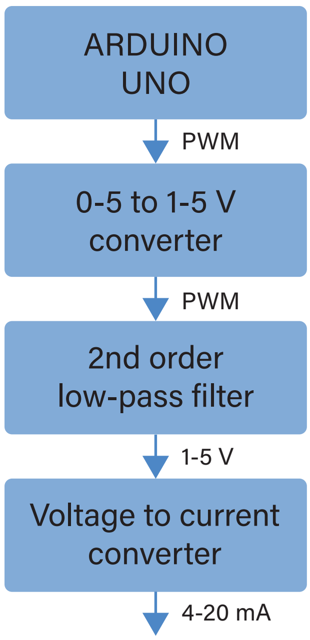


Figure 1: The block diagram of the system.

By connecting the filter directly to the PWM output, a signal which varies from 0 V to 5 V is obtained, which would give an output current of 0...20 mA. The pulse's duration is programmed with an 8-bit word. Just using the range from 1 V to 5 V, would waste 1/5 of the full-scale range and, therefore, reduce the resolution. To improve the current resolution from 20/255 to 16/255, I modified the minimum voltage level of the pulses, raising it from 0 V to 1 V, giving at the output a 4 to 20 mA current. The block diagram is shown in **Figure 1**.

The Circuit

Figure 2 shows the complete diagram of the circuit. To obtain pulses with a voltage span ranging from 1 V to 5 V, I had to use a 1 V source realized with U1A and Q1 transistor that works practically as a switch. The transistor Q1 inverts the PWM signal, so the number of the PWM duty-cycle must be complemented in software. The operational amplifier U1B operates as a separator; the filter uses U1C.

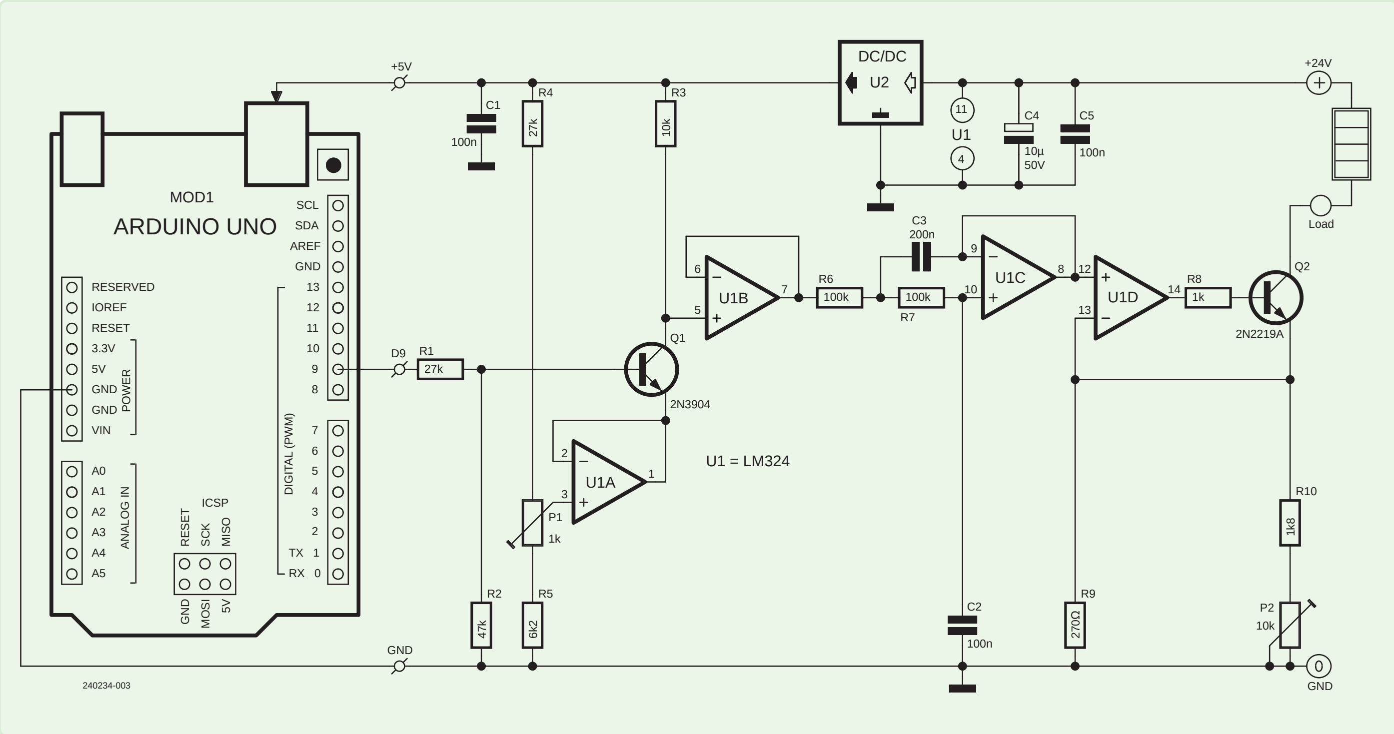


Figure 2: Schematic diagram of the 4...20 mA PWM current generator.

The voltage/current converter uses U1D and Q2. In detail, the smoothened voltage — ranging from 1 V to 5 V — present at the output of the Sallen-key filter reaches the non-inverting input of U1D, that progressively biases the base of Q2, increasing its collector-emitter current. This current is sensed by the network R10-P2-R9, which produces a feedback voltage to mitigate the same flow through the inverting input of U1D. In this way, when the maximum voltage of 5 V is applied to pin 12, with a proper adjustment of P2, the current on the loop is stabilized at 20 mA.

Trimmer P1 is used to adjust the minimum output current (4 mA) and P2 to adjust the maximum (20 mA). The theoretical value of the emitter resistor is Re = 5/0.02 = 250 Ω, but that does not consider the tolerances of the voltage supply of the Arduino and of the resistors' values. The resistor R8 is used as U1D output current limiter in case of absence of load.

A step-down converter is a suitable solution for powering the system because of the 24 V., This value can be varied from 12 V to 30 V, depending on the load circuit. Arduino UNO has a +5 V output pin; it is not recommended using it as a power input pin as this would be in parallel with the internal regulator. But it can be powered at +5 V using the USB connector. Other boards, for example, the Arduino Pro Mini, have a +5 V input.

The capacitors used for the filter must be measured with a capacimeter. For my prototype, I selected for C3 some 220 nF capacitors to search for a value that approached 200 nF; and for C2, I have selected a value half of C3. Q1 is a transistor that must have a low $V_{ce(sat)}$ and Q2 must have a current gain of at least 100 and a V_{ceo} of at least 40 V with a minimum power of 500 mW. The operational amplifier U1 must also be suitable for single-rail power supply, such as for example an LM324.

The component side layout of my prototype is shown in **Figure 3**. The resistor soldered on the top of the board is a precision load used for the calibration of the system. Q2 has a small heat sink because, at 20 mA and with a low voltage load, as in this case, it dissipates (24-3-5) V * 0.02 A = 320 mW. In these circumstances, it is better to reduce the 24 V.

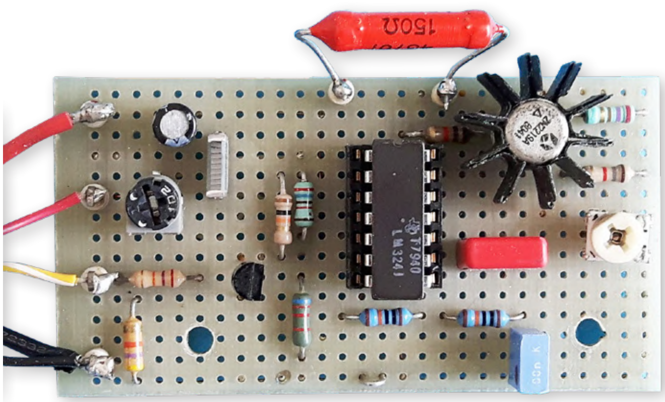


Figure 3: The finished prototype with the precision 150 Ω resistor used as a reference load (top) for testing.

Component List

Resistors
(All 5%, unless diff. indicated)
R1 = 27 kΩ
R2 = 47 kΩ
R3 = 10 kΩ
R4 = 27 kΩ, 1%, metal film
R5 = 6.2 kΩ, 1%, metal film
R6 = 100 kΩ, 1%, metal film
R7 = 100 kΩ, 1%, metal film
R8 = 1 kΩ
R9 = 270 Ω, 1%, metal film
R10 = 1.8 kΩ, 1%, metal film
P1 = 1 kΩ trimmer
P2 = 10 kΩ trimmer

Capacitors
C1...C3, C5 = 100 nF, Mylar
C4 = 10 µF, 50V, Electrolytic

Semiconductors
U1 = LM324 Quad. Op-Amp
U2 = DC/DC Converter 5 V output
Q1 = 2N3904 or equivalent
Q2 = 2N2219A or equivalent

Miscellaneous
Arduino UNO board

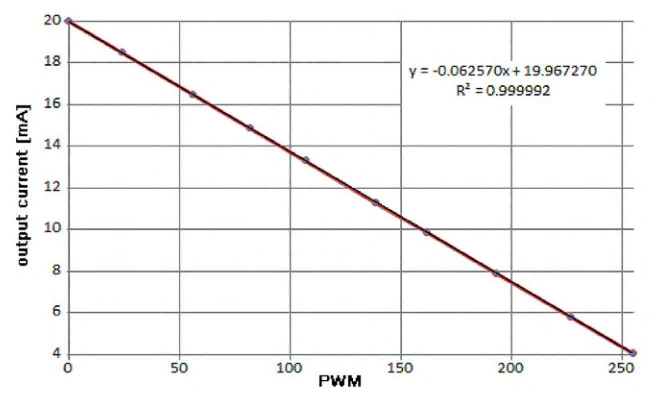


Figure 4: The PWM/current output chart: pretty linear, indeed!

Tests

To test the system, I used an Arduino with an LCD and a potentiometer connected to analog input A0; as pin PWM, I used D9. The program is simple: it reads the potentiometer, converts 10-bit to 8-bit analog reading and produces the PWM.

```
#include <LiquidCrystal.h>
int PWMpin = 9;     // PWM out on digital pin 9
int analogPin = 0;  // potentiometer connected to A0
int val = 0;        // variable to store the read value
char spacestring[17] ="                ";

// initialize the library with the
// numbers of the interface pins
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
  pinMode(PWMpin, OUTPUT);
        // sets the pin as output
  lcd.begin(16, 2);
        // set up number of columns and rows
  lcd.setCursor(0, 0);
        // set the cursor to column 0, line 0
  lcd.print("Stalker PWM test");
        // Print a message to the LCD
}
void loop() {
  val = analogRead(analogPin) >> 2;
        // read the potentiometer as 8 bit number
  analogWrite(PWMpin, val);
  lcd.setCursor(0, 1)
  lcd.print(spacestring);
  lcd.setCursor(0, 1);
  lcd.print(val);
  delay(500);
}
```

I reported on the spreadsheet the PWM values and the measurements in volts made on a precision resistor (150 Ω ±0.5%) that worked as a load. The PWM/output current diagram is shown in **Figure 4**. The linearity is excellent, as confirmed by the coefficient of determination R2 = 0.999992.

The diagram shows a negative slope; if you want a positive slope, which may be more natural for coding, the PWM value must be — as said — complemented to 255 (`val = 255 - val;`).

In my program, you could generate a new value every 500 ms (2 Hz), but you could reduce this period to 100 ms (10 Hz).

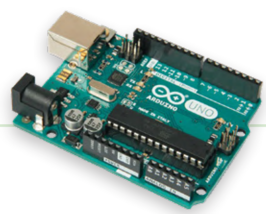
240234-01

Questions or Comments?

Do you have any technical questions or comments about this article? Contact the author at giov.carrera@gmail.com or the editorial team of Elektor at editor@elektor.com.

About the Author

Giovanni Carrera holds a degree in Electronic Engineering. As a university professor in the Faculty of Naval Engineering in Genoa, Italy, he taught numerous courses, such as naval automation and the simulation of ship propulsion systems. Carrera started working in the late 1970s with the 6502 CPU and then moved on to other processors. Today, he enjoys designing and developing analog and digital electronic circuits, many of which he has written about on his blogs (ArduPicLab and GnssRtkLab) and in various magazines.



WEB LINK

[1] Fundamentals of 4...20 mA current loops: https://tinyurl.com/a73t35j5

03 Vacuum Cleaner Automatic Control

Keep Your Tools' Work Area Clean

By Giuseppe La Rosa (Italy)

Woodworking produces a lot of dust, which is harmful to health. This design allows a vacuum cleaner to be combined with any power tool for cutting or sanding, enabling both to be turned on simultaneously, but delaying the vacuum at power off.

Dust raised by woodworking tools — besides being harmful to our lungs — tends to fall back into the work area and cover it, making the use of the tool inaccurate and even more dangerous!

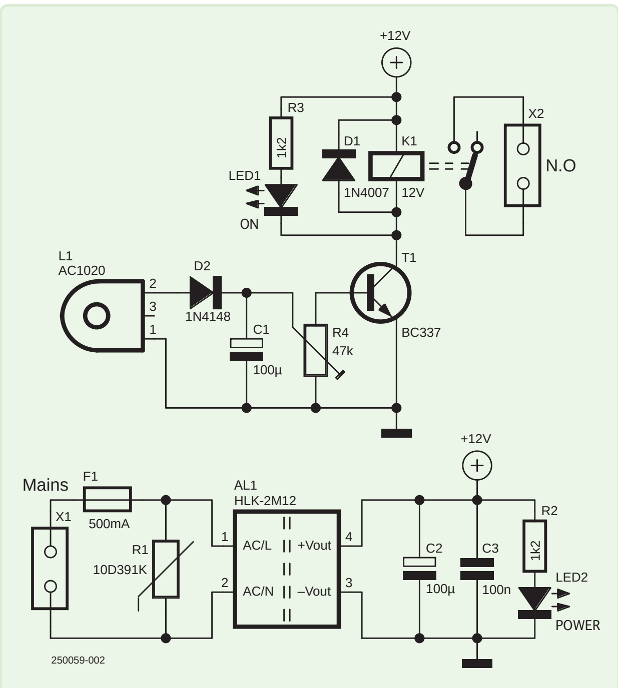


Figure 1: Circuit diagram of the Vacuum Cleaner Automatic Control.

This project is designed to be incorporated into a multi-socket power outlet. The circuit detects the current drawn by the power tool through an inductive sensor, automatically starting a vacuum cleaner when a current flow is detected and shutting it off about two seconds after the current stops. The overall maximum power applicable to the outlets is 1,800 W.

Circuit Diagram

Figure 1 shows the circuit diagram. As stated above, the project's task is to detect the current flowing over a primary power line to which a power tool is connected, activating the power supply on a secondary outlet via relay K1.

To achieve this, we use the L1 sensor, an AC1020 toroidal current transformer by Talema. This transformer generates the voltage needed to drive transistor T1. On this current sensor, to increase detection sensitivity, it is necessary to wind three turns of the L wire — of at least 1.5 mm² cross-section — of the line to be controlled, as depicted in the practical wiring diagram in **Figure 2**.

According to the operating principle of the transformer, a small alternating voltage — with the same frequency as the mains voltage — is generated at the ends of L1 by magnetic induction. This voltage is rectified by diode D2, smoothed by the electrolytic capacitor, C1, and then applied to trimmer R4, which allows the switching threshold of the transistor T1 to be adjusted.

When the voltage across the base of T1 exceeds a certain value (about 0.65 V), it starts conducting, and the relay is energized, sending AC power to the electrical outlet where the vacuum cleaner is connected.

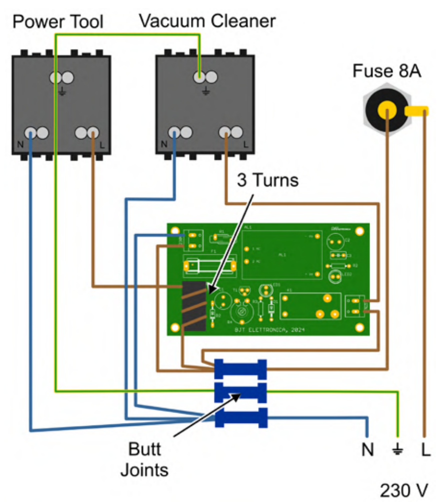


Figure 2: Overall wiring diagram among PCB, fuse holder, mains sockets and current sensor.

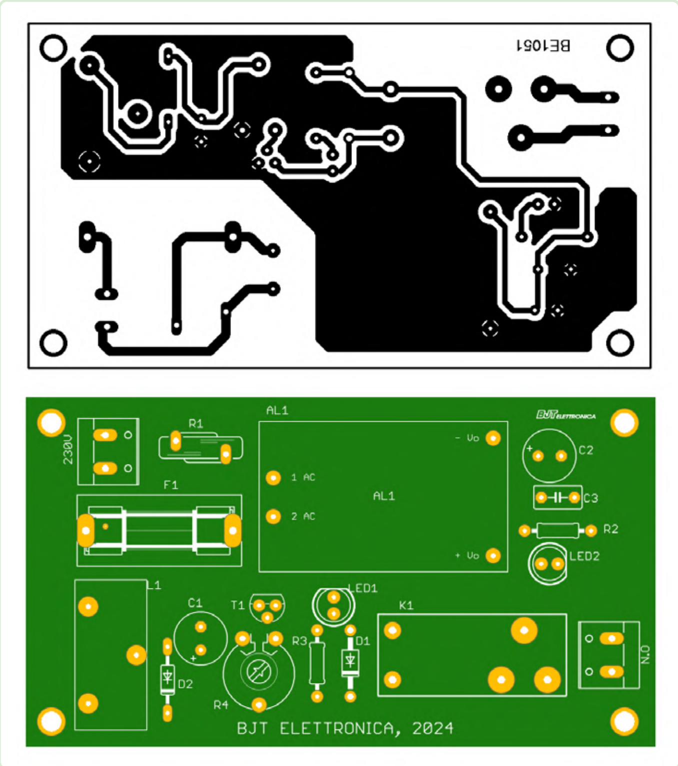


Figure 3: Layout of the soldering side (top) of the PCB and silkscreen of the component side (bottom).

Figure 4: The finished prototype, completely wired, before the initial testing.

When the current flow detected by the sensor is interrupted, the residual electrical charge of C1 keeps T1 on for a further 2 to 3 s — a useful delay to complete the evacuation of debris from the vacuum hose.

Diode D1, connected in parallel with relay K1's coil, protects T1 from the negative voltage spikes generated by the winding when its power supply is cut off. The circuit draws about 25 mA when the relay is energized; otherwise, its power consumption is negligible. A small 12 V PCB power supply module, Al1, was provided to power the whole circuit. The AC input of this module is protected by fuse F1 and varistor R1 for mains voltage spikes.

Practical Realization and Testing

The board is a single-sided type and can be easily handcrafted. The PCB layout shown in **Figure 3** (top) was used for the project. The board was made using the classic photo-etching method, and then, with the help of the silkscreen shown at the bottom of Figure 3, the planned components (see component list) were soldered.

When finished, the board should be housed inside a socket outlet box, as visible in **Figure 4**, which shows the complete realization and all the necessary links between the board and the Schuko sockets. The power socket that supplies the power tool must have its L wire wound for three turns inside L1, as described earlier. To secure the board to the enclosure, a 3D-printed holder was used, and the respective STL file, along with the PCB layout, can be downloaded at [1].

Once the wiring of the board is complete, a test can be carried out by connecting a power tool and a vacuum cleaner. If the relay does not activate, you will need to adjust trimmer R4. A video of the project in action is available at [2].

250059-01

Questions or Comments?

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

Component List

Resistors
R1 = 10D391K Varistor
R2, R3 = 1.2 kΩ, 0.25 W
R4 = 47 kΩ Trimmer, horizontal mount

Capacitors
C1, C2 = 100 µF, 16 V
C3 = 100 nF, polyester

Semiconductors
D1 = 1N4007
D2 = 1N4148
T1 = BC337
LED1 = 5 mm LED, orange
LED2 = 5 mm LED, green

Miscellaneous
L1 = AC1020 current transformer
AL1 = HLK-5M12 power module
K1 = 12 V relay, SPDT, PCB type
1× fuse holder 5×20 mm, PCB type
F1 = fuse 5×20 mm, 500 mA T
X1, X2 = 2-pole screw terminal block

WEB LINKS

[1] Download of extra files: http://tinyurl.com/elektorlabsdl
[2] Video of the prototype in action, YouTube: https://youtu.be/nwSn8p9JpT0

04 DDS Generator with ATtiny

By Burkhard Kainka (Germany)

Learn to use the ATtiny3216 to generate sine waves with a simple Direct Digital Synthesis approach. Achieve accurate frequency control with an internal DAC in a few steps.

The ATtiny3216 has an internal DAC with an 8-bit resolution, which can be used for a sine wave generator. Because only this one output is used here, you can also use a smaller controller like the ATtiny814.

The circuit diagram of **Figure 1** shows an experimental setup with which the Tiny can be programmed. The UPDI input (Unified Program and Debug Interface) of the ATtiny is connected to a serial USB converter for the PC. For software development and uploading the firmware to the controller, either the Arduino IDE or VSCode with PlatformIO can be used. Not everyone knows that the Arduino IDE does not only allow the usual code structure with `setup` and `loop`. `setup` and `loop` also have disadvantages, especially for time-critical tasks. For example, a timer interrupt is automatically set up that counts milliseconds in the background. This would lead to short interruptions and thus to disruptions of the output. For time-critical tasks, you can therefore also use pure C programming with the `main` function in the Arduino IDE and also access registers of the AVR controller directly. This allows you to keep a close eye on what is actually being executed.

Software

A simple sine wave generator can be built according to the Direct Digital Synthesis (DDS) principle. Data from a previously calculated sine wave table is output at regular intervals via a DA converter. The output frequency is determined by the step width through the table.

The C program [1] in **Listing 1** generates a sine wave signal at 1 kHz. The frequency can be hardcoded between 1 Hz and 20,000 Hz in the program. The loop in the main program initializes the sine table `dds[n]` with 256 interpolation values.

The program uses a timer interrupt with a repetition frequency of 20 MHz / 305 = approx. 65500 Hz, so that the generated frequency

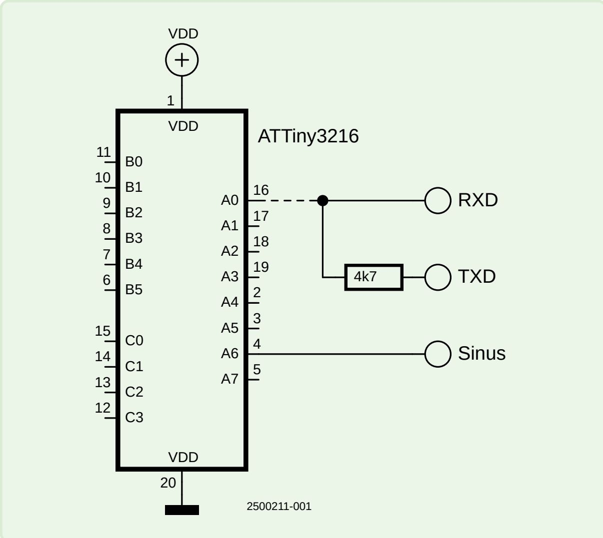


Figure 1: Circuit diagram of the experimental setup for programming the ATtiny3216. The dashed connection is only required when uploading the program.

has a resolution of 1 Hz. This allows you to add a numerical value of the frequency in `freq` without further conversion to the phase accumulator `phase`. This variable then points to the next value in the sine table that is used for the output. The sine table is only addressed with the upper eight bits of the phase accumulator (`DAC0.DATA = dds[phase>>8];`), but fractions of the phase jumps are also added up.

Practice

At a frequency of 256 Hz (**Figure 2**), the following value is read from the sine table for each interrupt. For smaller frequencies, it takes longer

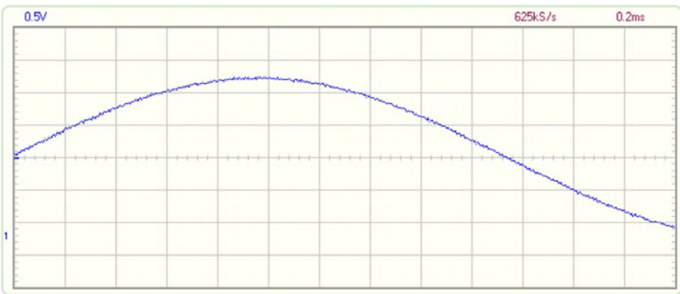


Figure 2: At 256 Hz, the transitions are still smooth and hardly recognizable.

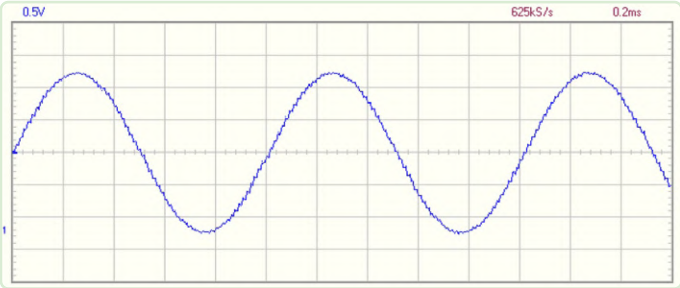


Figure 3: At 1 kHz, individual stages of the sampling are clearly visible.

for a new interpolation value to be output. And at higher frequencies, the output moves through the sine table in larger steps; the highest frequencies are then only displayed with a few interpolation values. Each frequency is generated with the accuracy of the internal clock generator of approximately 1%.

At 1 kHz, the individual steps are already clearly recognizable (**Figure 3**); at 10 kHz, they can no longer be overlooked. If the generator is to be used for a wider frequency range, it is recommended to add a steep-edged low-pass filter.

Translated by Juergen Kellner — 250211-01

Listing 1: C Program

```
//Sinus3216
#include <avr/io.h>
#include <avr/interrupt.h>
#include <math.h>

uint8_t dds[256];
uint16_t phase= 0;
uint16_t freq = 1000;  //1 kHz

ISR(TCA0_OVF_vect) {
    PORTA.OUT = PIN3_bm;
    DAC0.DATA = dds[phase>>8];
    phase += freq;
    PORTA.OUT = 0;
    TCA0.SINGLE.INTFLAGS = TCA_SINGLE_OVF_bm;
}

int main(void) {
    sei();
    _PROTECTED_WRITE (CLKCTRL_MCLKCTRLB, 0); // 20 MHz
    for (uint16_t n=0; n<256;n++){
       dds[n]=127+127.0*sin(3.1415*n/128.0);
    }
    PORTA.DIR |= PIN3_bm;
    VREF.CTRLA = VREF_DAC0REFSEL_2V5_gc;
    DAC0.CTRLA = DAC_OUTEN_bm|DAC_ENABLE_bm;
    DAC0.DATA  = 0;  // DA an PA6
    TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV1_gc;
    TCA0.SINGLE.PER = 305; // 65,536 kHz
    TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;
    TCA0.SINGLE.CTRLA |= TCA_SINGLE_ENABLE_bm;
    while (1) {
    }
}
```

Questions or Comments?

Do you have any technical questions or comments about this article? Please email the author at b.kainka@t-online.de or contact Elektor at editor@elektor.com.

About the Author

Burkhard Kainka (b-kainka.de) is a radio amateur and a well-known author of many Elektor books and articles. After many years working as a physics teacher, he started his own business as a developer and author in 1996.

WEB LINK

[1] Software download: https://www.elektormagazine.com/labs/elektor-articles-software-downloads

05 Opamp-Tester V2

New PCB — Now Also Suitable for SMDs

By Alfred Rosenkränzer (Germany)

At the end of 2024, my opamp tester for audio applications [1] was published. It enables evaluation of various opamps under identical conditions. The original version used through-hole metal film resistors. This new PCB version also allows the use of SMD resistors.

Once a selection of potentially suitable opamps has been made, often the only viable next step is to order samples and test them thoroughly. Harmonic distortion, for example, depends on input level, gain, output level, load, and frequency. Datasheets often provide these figures only for specific conditions, which do not necessarily represent all use cases and are not always easily comparable between manufacturers. To enable tests under identical conditions, this small test PCB was developed. With this tester, ICs can also be conveniently selected for similar parameters, or the best-performing examples identified.

Operating Principle

The circuit's simplicity makes a block diagram unnecessary. **Figure 1** shows that the circuit includes two dual opamps in DIL packages. IC1 is configured for non-inverting operation, and IC2 for inverting. The inputs of IC1A and IC1B are biased to ground via R1 and R6, respectively, establishing a defined input impedance of approximately 50 kΩ. Since the non-inverting opamp inputs are high-impedance, this is essential. R3 is not populated at IC1A, so the gain is +1. For IC1B, the gain is +10, based on the ratio (R8 + R9) / R8. These gain values can easily be adjusted by changing resistor values or by populating R3 as needed.

For IC2A, the gain is exactly –1 due to the R12 / R13 ratio. R16 and R17 set a gain of –10 for IC2B. With two dual opamps, the test setup thus offers four configurations: inverting and non-inverting, with gains of 1× and 10×. The values shown in **Figure 1** are suited to typical audio applications. Note that the input impedances for the inverting circuits are considerably lower — around 3.1 kΩ for IC2A and approximately 1 kΩ for IC2B — than for the non-inverting configurations.

Each opamp output is connected to two headers. A 220 Ω resistor links the two-pin connectors to measurement equipment. These series resistors suppress potential ringing when driving capacitive loads (e.g., oscilloscope inputs). The value is not critical and can range from 50 Ω to 1 kΩ. The primary purpose is to prevent oscillation with capacitive loads. The four-pin headers can accommodate various loads, enabling tests under different operating conditions.

Assembly

This new PCB version allows testing with different types of resistors. As with the previous version, through-hole components can be mounted on the top side, while the bottom side supports SMD resistors in 1206, 0805, and 0603 packages. MELF 0204 resistors also fit the 1206 pads. This explains the many parallel resistor options seen in Figure 1. The top layout (**Figure 2**) is similar to the previous version. **Figure 3** shows the bottom layout with SMD pads positioned between the through-hole pads.

The numbering of top-side components (connectors, opamps, capacitors, and through-hole resistors) remains the same as in the original version. The logic for SMD resistor designation is as follows: 0603 resistors have an offset of 20, and 1206 resistors have an offset of 40.

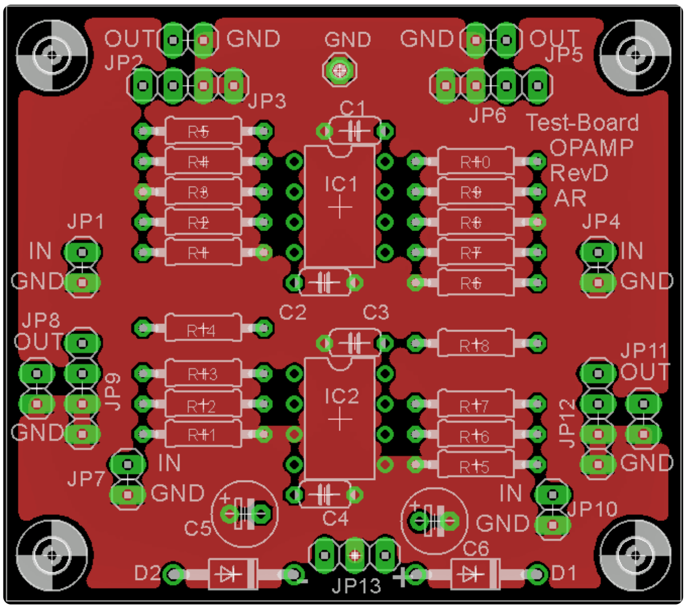


Figure 2: The top layout of the PCB is largely unchanged from the first version of the circuit.

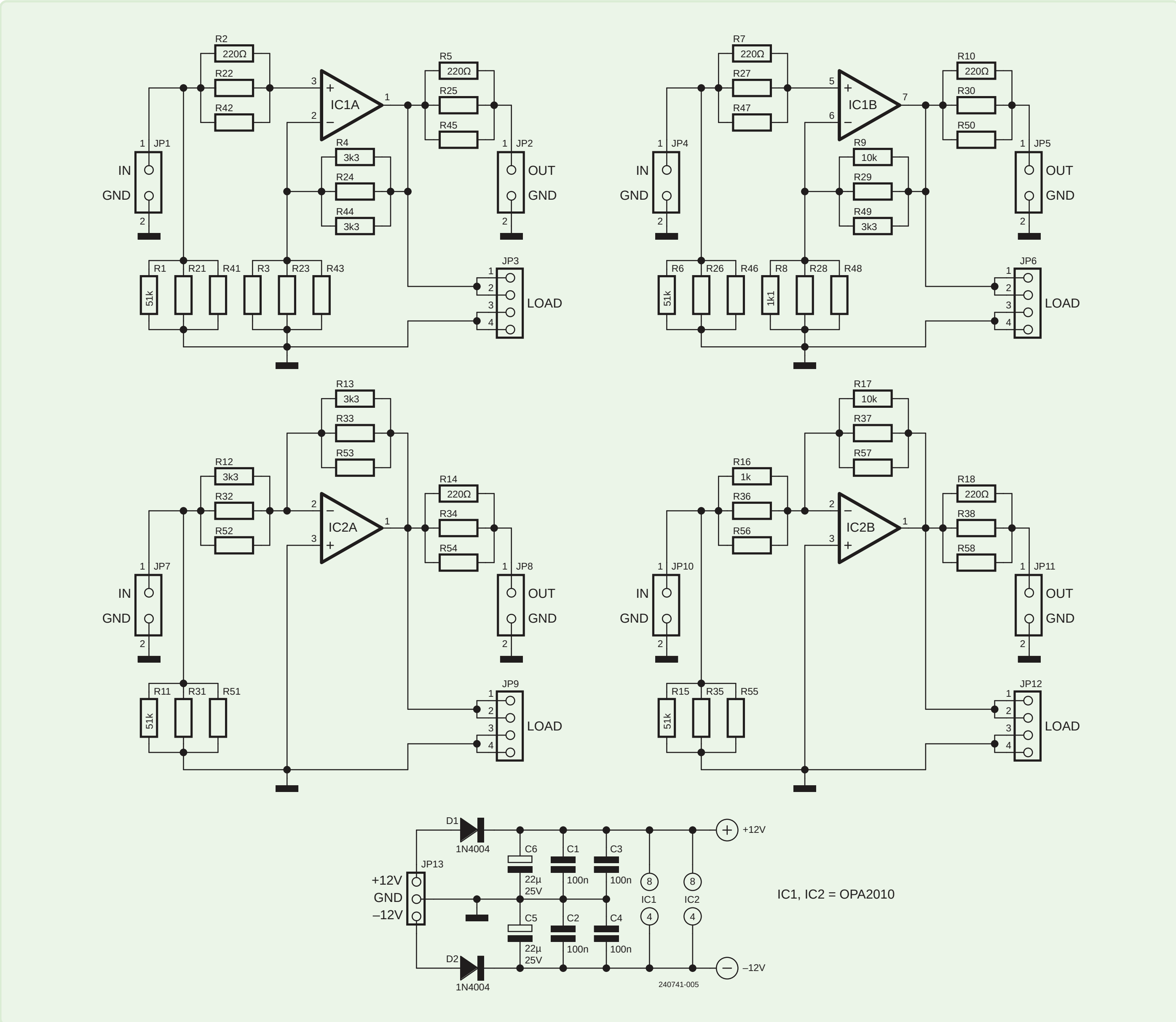


Figure 1: The opamp tester circuit is fairly simple: Two dual opamps and a few passive components are all it takes.

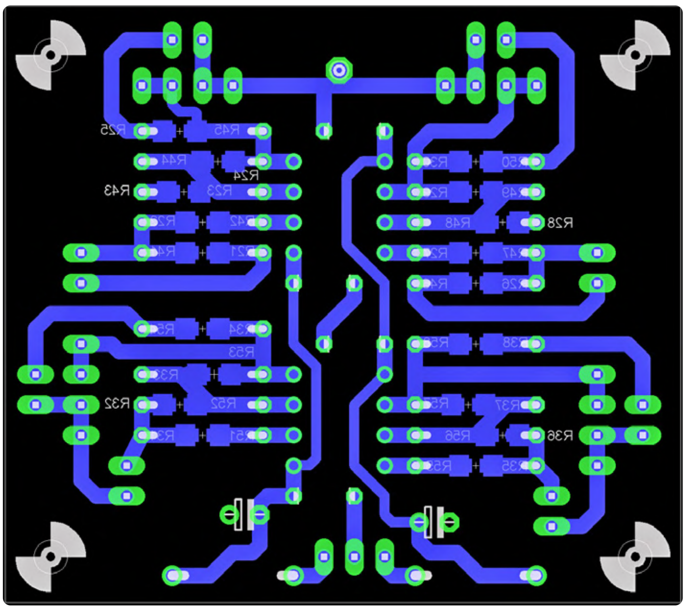


Figure 3: The bottom layout shows the pads for the SMD resistors.

For instance, the SMD equivalents of through-hole resistor R5 at the output of IC1A are R25 (0603) and R45 (1206). Only one resistor variant should be installed — either R5, R25, or R45.

The PCB layout files are available for download [2].

Additional Notes

Power is supplied symmetrically via JP13. Two electrolytic capacitors placed directly at the connector and two 100 nF capacitors per IC ensure decoupling. Voltage regulators are deliberately omitted to allow testing of opamp performance at different supply voltages. Newly added are D1 and D2 for reverse-polarity protection.

High-quality turned-pin DIL sockets are provided for the opamps, making it easy to swap ICs without soldering. For SMD ICs, breakout boards (BoBs) are required — these can be custom-made or purchased from various sources, including online platforms. **Figure 4** shows a populated board with an opamp on an SMD breakout board, and another opamp in DIL format. In my experiments, these adapters did not introduce any measurable influence.

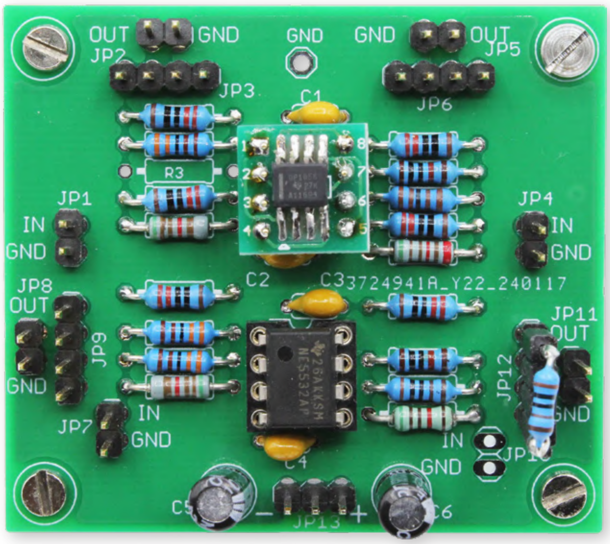


Figure 4: A board assembled with through-hole resistors. IC1 is on an SOIC8 breakout board; IC2 is in a "large" DIL package. A load is plugged into output JP12.

To test opamps for audio use, a low-noise setup is recommended. All resistors should be metal film types. The 4-pin headers, JP3, JP6, JP9, and JP12, allow loads to be plugged in — loads such as resistors mounted on matching 4-pin SIL socket strips. This enables quick swapping of load configurations with varying resistance and/or capacitance.

As mentioned earlier, this board can also be used to test resistors. You can compare through-hole and various SMD types, provided the same opamps are used. While differences between metal film resistor types may be minor compared to differences between opamps (with consistent resistance), but resistor noise characteristics are still of interest.

To measure minor distortion, suitable measurement instruments are required. Over the years, I have published several Elektor articles on measurement techniques [3]. My review of the QA403 by QuantAsylum covers an interesting audio analyzer [4]. Specific filters can extend the measurement range into extremely low distortion levels, as shown in [5].

240741-01

Component List

Resistors

1% metal film, through-hole

R1, R6, R11, R15 = 10 k
R2, R5, R7, R10, R14, R18 = 4k7
R3 = omitted*
R4, R12, R13 = 3k3
R8 = 1k1
R9, R17 = 10 k
R16 = 1 k

SMD 0603

R21, R26, R31, R35 = 10 k
R22, R25, R27, R30, R34, R38 = 4k7
R23 = omitted*
R24, R32, R33 = 3k3
R28 = 1k1
R29, R37 = 10 k
R36 = 1 k

SMD 1206

R41, R46, R51, R55 = 10 k
R42, R45, R47, R50, R54, R58 = 4k7
R43 = omitted*
R44, R52, R53 = 3k3
R48 = 1k1
R49, R57 = 10 k
R56 = 1 k

Capacitors

C1...C4 = 100 n, film, 2.54 mm/0.1" pitch
C5, C6 = 22 µF/25 V, electrolytic, 2.54 mm/0.1" pitch

Semiconductors

D1, D2 = 1N4004
IC1, IC2 = e.g. TL072, dual opamp*

Miscellaneous

4 IC sockets, 8-pin, turned contacts
JP1, JP2, JP4, JP5, JP7, JP8, JP10, JP11 = 2-pin header, 2.54 mm/0.1" pitch
JP3, JP6, JP9, JP12 = 4-pin header, 2.54 mm/0.1" pitch
JP13 = 3-pin header, 2.54 mm/0.1" pitch
Matching 4-pin female headers for loads*
PCB

* see text

Questions or Comments?

Do you have questions or comments about this article? Email the author at alfred_rosenkraenzer@gmx.de, or contact Elektor at editor@elektor.com.

WEB LINKS

[1] Alfred Rosenkränzer, "Opamp Tester," Elektor 11-12/2024: https://www.elektormagazine.com/magazine/elektor-358/63323
[2] PCB layout files on Elektor Labs: https://www.elektormagazine.com/labs/opamp-und-widerstands-tester
[3] Articles by Alfred Rosenkränzer in Elektor: https://tinyurl.com/4kb8wm7w
[4] Alfred Rosenkränzer, "Comparing the QuantAsylum QA403 to the Gold Standard," elektormagazine.com: https://www.elektormagazine.com/review/quantasylum-qa403-audio-analyzer-review
[5] Alfred Rosenkränzer, "A Fliege Notch Filter for Audio Measurements," Elektor 9-10/2022: https://www.elektormagazine.com/magazine/elektor-272/60926

06 550-mW "Lamp" Audio Amplifier

Get the Warm Sound of Vacuum Tubes With Ease

By Mike Wens (Belgium)

The design of an A-class audio amplifier is not necessarily complicated or unattainable for most people. On the contrary, with a few tricks, it is possible to build a simple, low-power audio amplifier with enviable sonic performance and have fun doing it!

This project was done in 2022 with help from my 8-year-old son Louis and 11-year-old daughter Lena. Both were interested in their dad's tube — in Flemish referred to as "Lamp" — amplifier projects and wanted to build their own one. As I found voltages of several hundreds of volts a little high to have them play with it, I came up with another design of amplifier that was supposed to use real incandescence lamps.

A small, simple, low-power pure A-class bipolar transistor amplifier would have been perfect. The project would have to include several aspects of building something: a very basic explanation of a transistor acting as an amplifier for audio signals, sawing and drilling an aluminum chassis, mounting connectors, switches, potentiometers etc. onto the chassis and finally wiring and soldering it. All of which was to be done by my two youngest children, safely guided by the author. My oldest daughter Anna could have helped with the sonic assessment, too.

Circuit and Components

The whole thing was built using home-stocked components, some over-50-year-old Motorola 2N3055s in TO-3 case and a BD115 in TO-39 with gold-plated leads, by Philips, I think. The schematic is shown in **Figure 1**, and its original version was drawn in LTSpice, whose file is available for download at the Elektor Labs page for this project [1]. Although the exact transistor models for BD115 were not available, it did allow for simulating the basic functionality as a first step.

A Darlington configuration creates a sufficiently high input impedance to the signal (music) source; the 2N3055, on its own, has a $h_{fe} \approx 70$ (current amplification), so a base current of 1 mA results in a collector current of approximately 70 mA. For the full power of 550 mW in an 8-Ω load, the AC current (peak) will be about 370 mA, so about 5 mA AC base current is required for full drive. A value of 5 mA is a bit on the high side for most (Bluetooth) audio sources.

In the Darlington configuration, the base current for the 2N3055 is delivered by the BD115, and it flows through the main collector load. Hence, the audio source needs to deliver 70 times less current (BD115 has $h_{fe} \approx 70$). The resulting 70 µA can easily be delivered by any audio source. R1 is placed to protect the 2N3055 from over current in its base, and will also linearize the Darlington a bit for a lower distortion.

Another advantage of Darlington is that the DC bias voltage at the base of the BD115 will be the sum of its base-emitter diode and that of the 2N3055, or about:

$V_{BE\ BD115} + V_{R1} + V_{BE\ 2N3055} = 0.7\ V + 0.6\ V + 0.4\ V = 2.7\ V$ (measured).

This is well compatible with the 1 V to 2 V RMS output voltage capability of most sources and provides the right amount of input signal amplitude dynamic range.

A simple potentiometer (P1, 250 kΩ) biases the circuit to provide the minimum base current in the BD115, allowing to play with the set point on the collector of the 2N3055 output stage, bringing it to around 6 V — or half way the 12 V supply voltage.

The Lamp

A 12 V/10 W halogen lamp (nowadays obsolete and replaced by LEDs) acts as a load for the

A-class output stage. The halogen lamp works at half its rated supply voltage, resulting in a current of about 650 mA and a "cooler operation" in respect to its rated values. A nice side effect is the warmer light glow and the increased lifetime of the lamp itself.

But why a lamp, and not a resistor? Well, the resistor would run quite hot at 3.9 W and, without giving any visual indication, would be a risk for your fingers! Moreover, the initial Lamp is an important feature of this amplifier.

In fact, the lamp is a positive voltage (rather positive temperature) dependent resistor: at 6 V it has about 6 V / 0.65 A = 9.2 Ω resistance, where at the rated 12 V it will show about 14.4 Ω. This helps to limit the current through the output transistor at high output signal amplitudes. In this case, the 2N3055 is in saturation and only has a voltage drop of about 0.2 V.

Hence, the lamp effectively protects it against uncontrolled increase in the collector current.

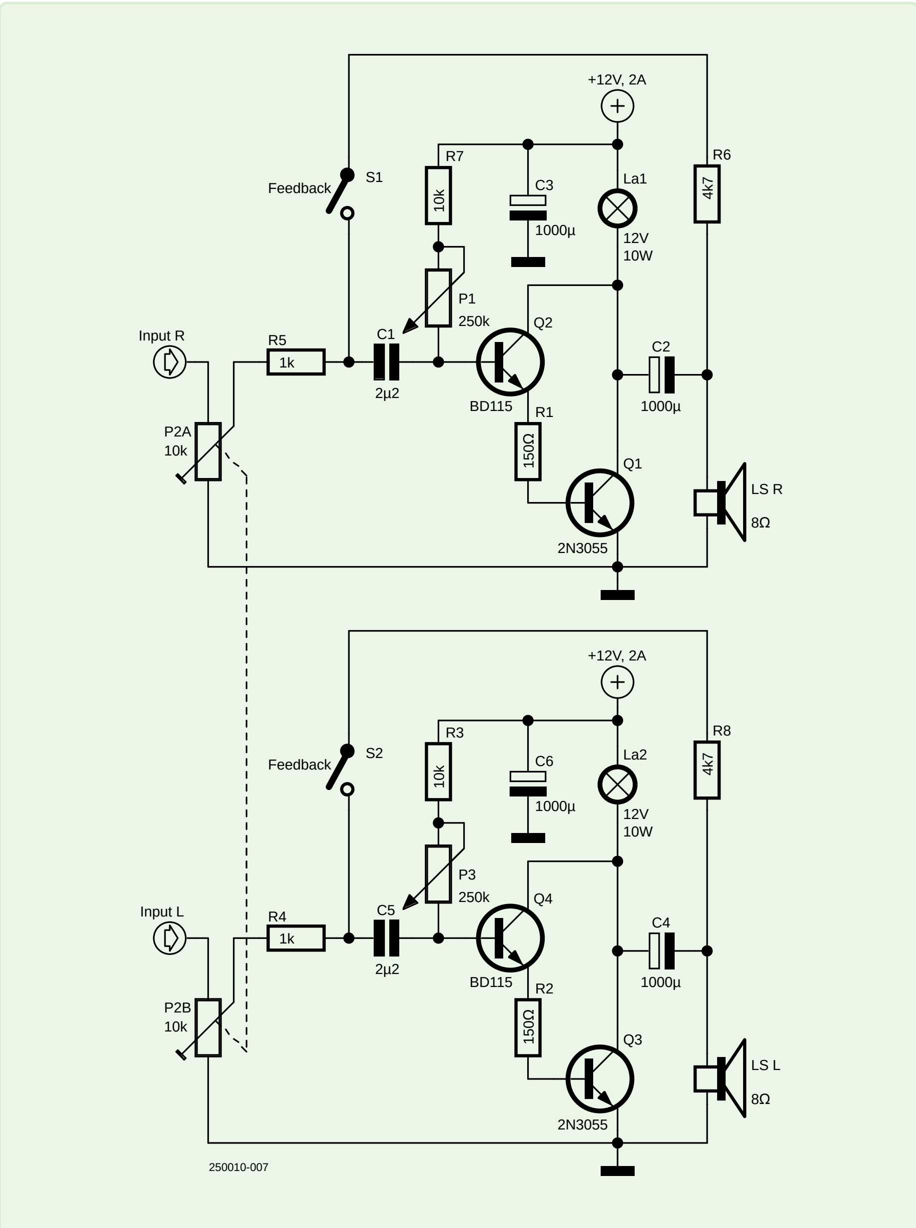


Figure 1: Schematic diagram of the Lamp amplifier. It can also be simulated in LTSpice.

An additional advantage will be a softer clipping behavior, which is beneficial for the sound. Don't try using an LED instead, it simply won't work!

The signal is applied through a 10 kΩ audio logarithmic potentiometer, which is in turn connected to RCA inputs to achieve volume control. The slider of the volume pot goes to the base of the BD115, but only after passing through a 2.2 µF decoupling film capacitor. These capacitors separate the DC component (about 2.7 V) at the BD115 base from the pure AC signal of the audio source.

For the same reason, the loudspeaker is connected in series with a 1000-µF electrolytic decoupling capacitor. A loudspeaker can only work with AC signal and doesn't tolerate any DC component, that would cause the voice coil to deflect and saturate the magnetic field: an awful event for both the loudspeaker and the sound quality. Not even to mention that a direct current, in this case, would likely heat, and even burn the voice coil.

Feedback

As an additional feature, a feedback switch is foreseen. The R6 / (R5 + R_{input}) ratio determines the amount of feedback applied. The output signal is 180° shifted from the input signal. So, by feeding back a small portion of the output signal to the input, we are counteracting the input signal. Amplification will be reduced with this feedback, but in return, we'll get less distortion and a lower output impedance.

Without feedback the amp will sound louder and more "open", with feedback it will sound more controlled but less "bright." We let you judge what you prefer, but in (high-end) tube amps, less feedback is generally considered better by some audio enthusiasts.

Finally, the Lamp Amp needs power. To keep things safe and simple, I used a standard wall-plug type switched-mode power supply that is rated as 12 V, 2 A DC. Inside the amplifier, additional 2 × 1000 µF capacitors (C3 and C6) act as a local buffer on this 12 V power supply. A (kill) switch allows to turn off the 12 V supply.

Figure 2: Real amplifiers glow in the dark! A rearview of the Lamp amp, built on a piece of aluminum rectangular profile.

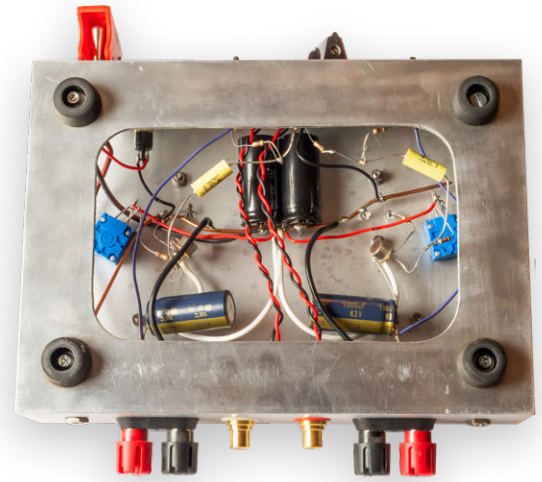


Figure 3: The bottom view, showing the point-to-point air wiring technique.

Practical Realization

The chosen aluminum chassis was large enough for hosting two identical sections of this circuit, thus realizing a stereo version with independent bias and feedback controls for each channel. The chassis acts as a heat sink for the two 2N3055 power transistors, each dissipating about 6 V × 0.65 A = 3.9 W. A piece of copper tubing effectively protects the lamps from burning fingers and improves the overall look of the design. Make sure to mount the 2N3055 with a mica isolation, as the TO-3 housing is electrically connected to the collector; apply heat dissipating compound on both sides of the mica, to ensure a good thermal transfer. Nylon washers are also needed, to isolate the mounting screws.

The little amplifier has a kind of steam-punk-looking. After building it, the first thing we did was to pair it with a Bluetooth player and a set of old loudspeakers. J.J. Cale sounded very relaxed "After Midnight." The amp was born and played remarkably well and "warm" — no transistor harshness to be spotted.

A little bit of feedback, operated by two switches in the front, allowed us to control the sound from very open to a bit more controlled, depending on the style of music and mood. Since then, we spent hours playing music through the little monster, which is always nice quality time and my perfect excuse to introduce some older music.

The rear side of the finished prototype is shown in **Figure 2**. The usage of the old-school point-to-point air wiring eliminated the need for a PCB, as you can see in **Figure 3**. A massive 2.5-mm^2 solid copper wire, anchored to the chassis, works as a ground connection, as well as a mechanical fixation (solder) point for the "floating components." Input signal wire twisting gives sufficient resilience against external EMI, lowering or avoiding the hum.

Some Measurements

So, what can this little thing do, and what are the specs? Since the engineer in me is difficult to suppress, I performed some late-night measurements using an oscilloscope and build-in waveform generator. The results for a sinewave input signal with and without applying feedback are shown in **Figure 4**. The output signal (green) looks almost like a sine wave, too. The slight asymmetry hints 2nd harmonic distortion, but gets better with feedback.

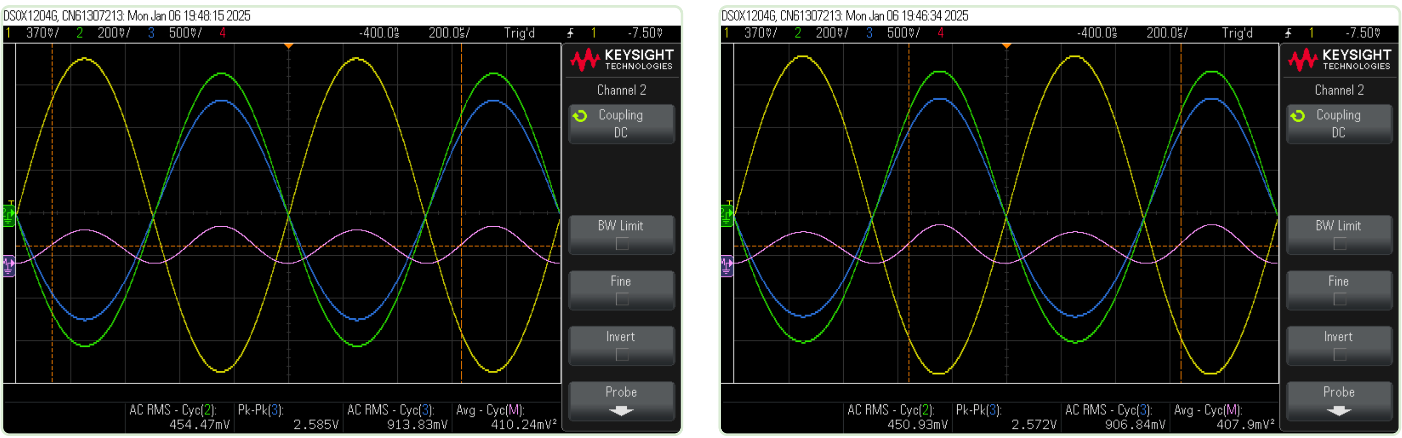


Figure 4: Measurements with a 1 kHz sinewave input signal. CH1: V_{in}, CH2: V_{out}, measured over 4 Ω (a measure for the output current, done on the low-side resistor of the 8 Ω load, made by two 4 Ω resistors in series), CH3: V_{out}, CHM = CH2 × CH3 (pink, a measure for the output power). The screenshots show the amplifier's response in open-loop (left) and with feedback (right).

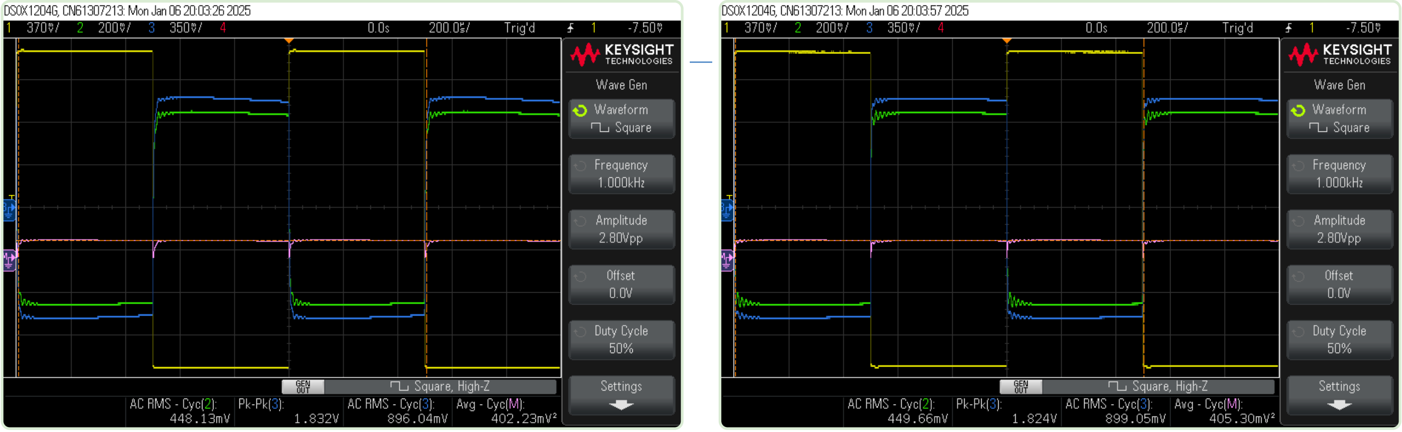


Figure 5: Measurements with a 1 kHz square-wave input signal. CH1 = V_{in}, CH2 = V_{out}, measured over 4 Ω (a measure for the output current), CH3 = V_{out}, CHM = CH2 × CH3 (pink, a measure for the output power). The screenshots are showing the amplifier's response in open-loop (left) and with feedback (right).

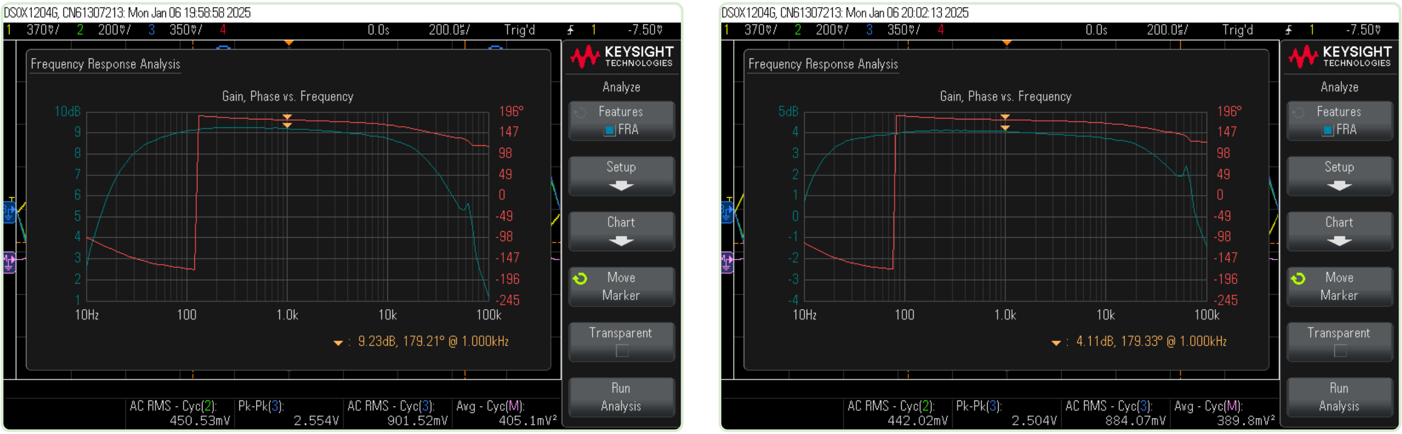


Figure 6: The Bode plots, showing phase (red) and amplitude (blue) for open-loop (left) and feedback operation (right), with an 8 Ω load.

The same setup, but with a square-wave response, is depicted in **Figure 5**. The response shows slight ringing, which predicts a peak in the Bode plot (frequency response) at high frequency. Feedback doesn't help here, it only straightens the wave a bit (so less distortion).

The Bode plot shown in **Figure 6** is impressive! It improves even with feedback (the phase step of 180° is an artifact of my oscilloscope). The peak at high-frequency is there, but since it's approx. around 50 to 60 kHz, it won't be heard. One can clearly see that feedback improves the frequency response from 35 Hz...18 kHz to 17 Hz...29 kHz at +/-1dB. A less wanted effect is the reduction of overall gain.

The distortion analysis can be done with the FFT (Fast-Fourier-Transform) function of the digital oscilloscope. When applying this over sufficient cycles, it gives reasonably accurate results. With no feedback, the FFT result is shown in **Figure 7**. A high 2nd order (-25 dB, or 5.6%), but very low 3rd order (-49 dB or 0.35%) harmonics explains the nice "tuby" sound.

Figure 7: FFT using a 1 kHz sinewave input signal, where CH1 = V_{in}, CH2 = V_{out}, CHM = FFT, with 100 mW on 8 Ω in open-loop, 2nd harmonic (left) and 3rd harmonic (right).

Figure 8: FFT using a 1 kHz sinewave input signal, where CH1 = V_{in}, CH2 = V_{out}, CHM = FFT, with 100 mW on 8 Ω with feedback, 2nd harmonic (left) and 3rd harmonic (right).

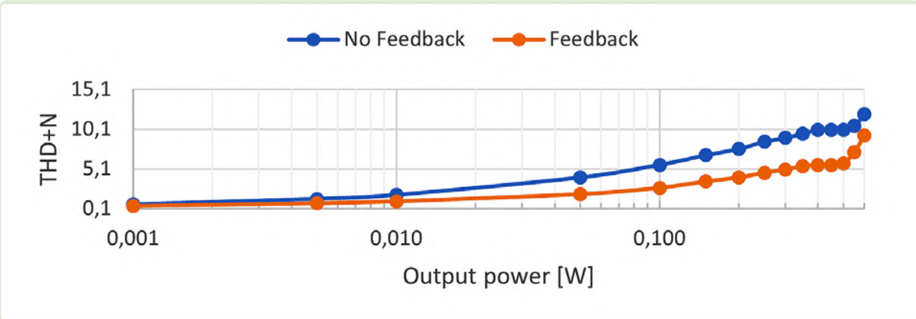


Figure 9: Total Harmonic Distortion + noise (THD + N) plots, as a function of the output power.

With feedback the FFT results are shown in **Figure 8**. A lower but still fairly high 2nd order (-30 dB, or 3.2%), but very low 3rd order (-68 dB, or 0.04%) harmonics explain the still nice "tuby", but less "airy" sound.

Finally, I couldn't resist myself from digging up my audio distortion meter and accompanying low-distortion sine wave generator. This allows an easier measurement of the total harmonic distortion over a wide output power range (1 mW until 600 mW clipping level), as shown in **Figure 9**.

Let It Glow!

This project turned out to be very fun and interesting to carry out, as it involved electronics, mechanics, machining and audio engineering. The kids had lots of fun and were proud of our joint achievement!

The Lamp amplifier seems to me a nice project to explore some basic principles of analog electronics, like resistance, capacitance, amplification, BJTs, biasing, feedback, impedance and so on. Years later, I found some guy who took this subject more seriously, but at the same time a bit too dangerous for a kids' project [2].

250010-01

Questions or Comments?

Do you have technical questions or comments about this article? You may contact the author at mike.wens@skynet.be or the editorial team of Elektor at editor@elektor.com.

Related Products

- **Douk Audio P6 mini Tube Preamplifier**
  www.elektor.com/21014
- **Bob Cordell, *Designing Audio Power Amplifiers* (2nd Edition) (Routledge 2019)**
  www.elektor.com/19150

About the Author

Mike Wens rolled into electronics at a young age when his parents gave him an experimental kit with transistors, which was then amplified when the dad of a friend gave him a treasure of old 60, 70s and 80s *Elektuur* magazines. Later on, he became an IC designer and engineer and conducted a PhD in microelectronics (on-chip DC/DC converters) at KULeuven, Belgium. Audio and tube amps have always been a hobby, which he's been sharing for over 20 years with his father-in-law. Today Mike is the co-founder of a company (*www.mindcet.com*) that designs and tests ICs (Chips) for various applications, including satellites, biomedical implants and D-class amplifiers.

WEB LINKS

[1] Elektor Labs page for this project: https://www.elektormagazine.com/labs/550-mw-lamp-audio-amplifier
[2] Zen Amp Project webpage, Pass DIY: https://tinyurl.com/yeyut8da

07 Fuse Guard

Monitoring a Fuse with a Flashing LED

By Hans-Norbert Gerbig (Germany)

Measuring is good; signaling is better! This circuit with only four components indicates with an active flashing LED when a mains fuse has failed.

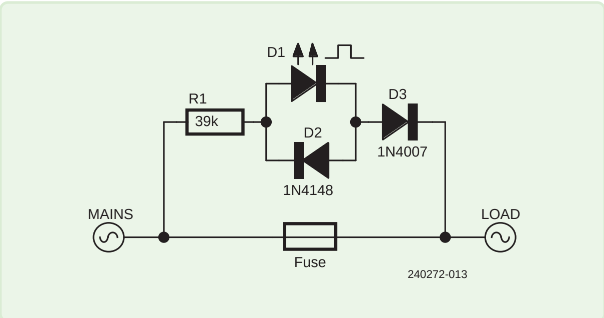


Figure 1: The super-simple circuit of the fuse guard.

The term "circuit" is almost an exaggeration for this four-component solution. The fuse guard is connected in parallel to a mains fuse. If the fuse is OK, no voltage drops across it and everything is fine. However, if the fuse is triggered or "blown," almost the full mains voltage of 230 V≈ is present at the monitoring electronics via the connected load. All that is then needed to make an ordinary flashing LED light up is a little current or voltage limitation and rectification.

Current Limitation

Figure 1 shows the circuit: When the fuse is triggered, the mains voltage is applied to R1 and the cathode of D3. R1 has a dual function: On the one hand, its resistance simply limits the current flowing. On the other hand, it also indirectly limits the voltage drop across the LED. If the current is small enough, the voltage dropping across D1 also remains small enough (in the 5 V range), which is beneficial for the blinking LED and its integrated electronics. An explicit voltage limiter is therefore not required. Diode D2, which is anti-parallel to D1, reduces the potential reverse voltage to a maximum of -0.7 V, which is not dangerous for D1.

Essentially, the circuit would be complete at this point, and three components would be sufficient for the function as a fuse monitor. However, with D3, another diode ensures that only half-waves generate current at R1, so only half the power loss occurs. At a voltage of 230 V≈, only about 3 mA_{eff} flows, which results in a dissipation of just under 0.7 W. Therefore, a 1 W version that is suitable for mains voltage is sufficient for R1.

Assembly

The fuse guard can easily be set up on a small experimental strip-board (**Figures 2** and **3**). According to the IPC2221 standard, the distances between the strips are, in principle, sufficient for mains voltage, at least in dry ambient conditions. Nevertheless, the areas between which mains voltage is present should ideally have larger distances in the range of ≥2 mm. To achieve this, it may be necessary to remove some copper from the conductor tracks.

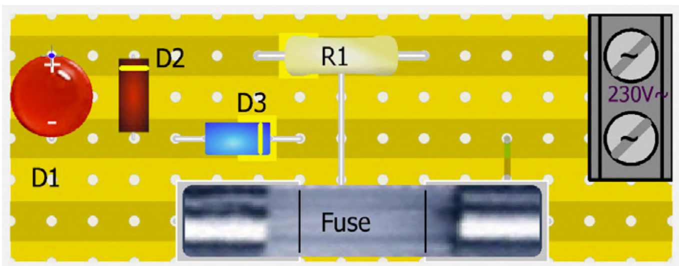


Figure 2: Proposal for populating the experimental board.

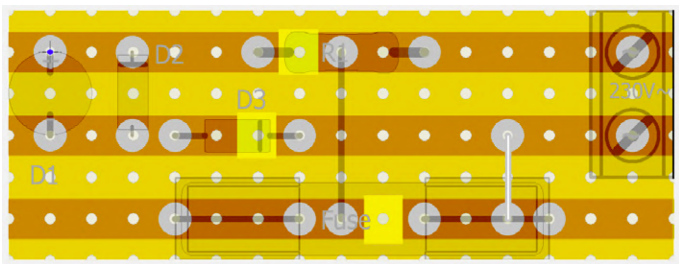


Figure 3: "Wiring" on the back of the experimental board.

As already mentioned, the fuse guard is connected in parallel to the mains fuse to be monitored. If a fuse blows, causing a power outage, the circuit becomes active and flashes. Red or yellow are suitable LED colors — a green LED would misleadingly imply that the fuse is "good." **Caution with mains voltage: Since the circuit is not electrically isolated from the mains, it must be installed so that it is safe from touch.** Elektor does not like to lose readers to Zeus, who, as you know, is the one responsible for lightning!

Translated by Jörg Starkmuth — 240272-01

About the Author

Hans-Norbert Gerbig taught at the Franziskaner-Gymnasium-Kreuzburg secondary school in Grosskrotzenburg, Germany, where he also supervised various electronics working groups. He was already fascinated by radio electronics during his school days. Now that he is retired, he can devote himself wholeheartedly to his technical hobby.

FEATURED TOPIC

Visit our **Circuits & Circuit Design page** for articles, projects, news, and videos.

www.elektormagazine.com/**circuits-and-circuit-design**

Related Products

- **FNIRSI S1 True RMS Smart Multimeter**
  www.elektor.com/20645
- **ZD-11E PCB Holder**
  www.elektor.com/20314

08 HQ RIAA Preamplifier

Get the Most Out of Your Vinyl Records!

By Thierry Clinquart (Belgium)

Despite the success of digital media, the old-fashioned vinyl records have never been more popular, nowadays. However, the correct reproduction of an analog record requires the conditioning of the pickup signal, according to a non-linear function specified by the RIAA standard. This High-Quality RIAA Preamplifier design brilliantly solves this problem, with outstanding results!

Since I started writing this article on Elektor Labs, I have noticed that over the last three years the "vinyl-mania" has grown, becoming even stronger through plenty of vintage events. In our cities, we see fairs where you can find equipment and record collections.

Why This Design?

While brands such as Thorens, Dual, Lenco, and many others are still well known, there is a slight downside when it comes to today's listening systems. With Hi-Fi giving way to digital interfaces and Bluetooth, many times the RIAA preamp is missing from the new generation of devices for listening to your records. However, an auxiliary input in RCA or 3.5 stereo mini jack may still be available. That's why I suggest you make your own preamp.

The circuit is based on Analog Devices's AN-124 application note [1]. Initially, I had replaced the SSM2015 input preamp (obsolete) with a THAT1510 by THAT Corporation. The latter can provide a higher gain level for MC (Moving Coil) cartridges, a kind of pickup where the output levels are often below 1 mV.

However, since most people use MM (moving magnet) cartridges on their turntables, I decided to leave the THAT1510 for MC cartridges use, and wanted to simplify the final design by switching to another IC, a NE5532 by Texas Instruments. I implemented it through the whole signal chain of this project.

This project — available at its respective Elektor Labs page [1] with all the downloadable materials — could be integrated into one of my many other audio designs, such as [2] and [3].

Circuit Diagram

The schematic diagram of the RIAA Preamplifier is shown in **Figure 1**. The 47 kΩ resistor R_{IN} loads the cartridge to its typical impedance. The capacity value of C_{IN} is 47 pF, but may vary (see the pickup manufacturer's specifications). C1 prevents any DC component from entering the cartridge. C3 acts as a high-pass filter, cutting-off the very low end of the audio spectrum. The input stage has a gain of 28 dB, or 20 log 1 + (47 kΩ / 1.2 kΩ).

For RIAA signal equalization, we know that during recording the low frequencies are attenuated, while the high portion of the audio spectrum is boosted. When it comes to playback, the output signal from the pickup must undergo inverse conditioning, that is, amplification of low frequencies and attenuation of high frequencies.

In an RIAA filter, there are three time constants:

- Treble = 75 µs: C7 × [(R9 × R11) / (R9 + R11)]
- Midrange = 318 µs: C7 × R11
- Bass = 3180 µs: C9 × R13

The filter components must have a tolerance of ± 1%. Here, I'm not providing any details on the IEC-RIAA standard. For those who are interested, this topic is addressed in the AN-124 Application Note mentioned above, available at the author's Elektor Labs page for this article [1].

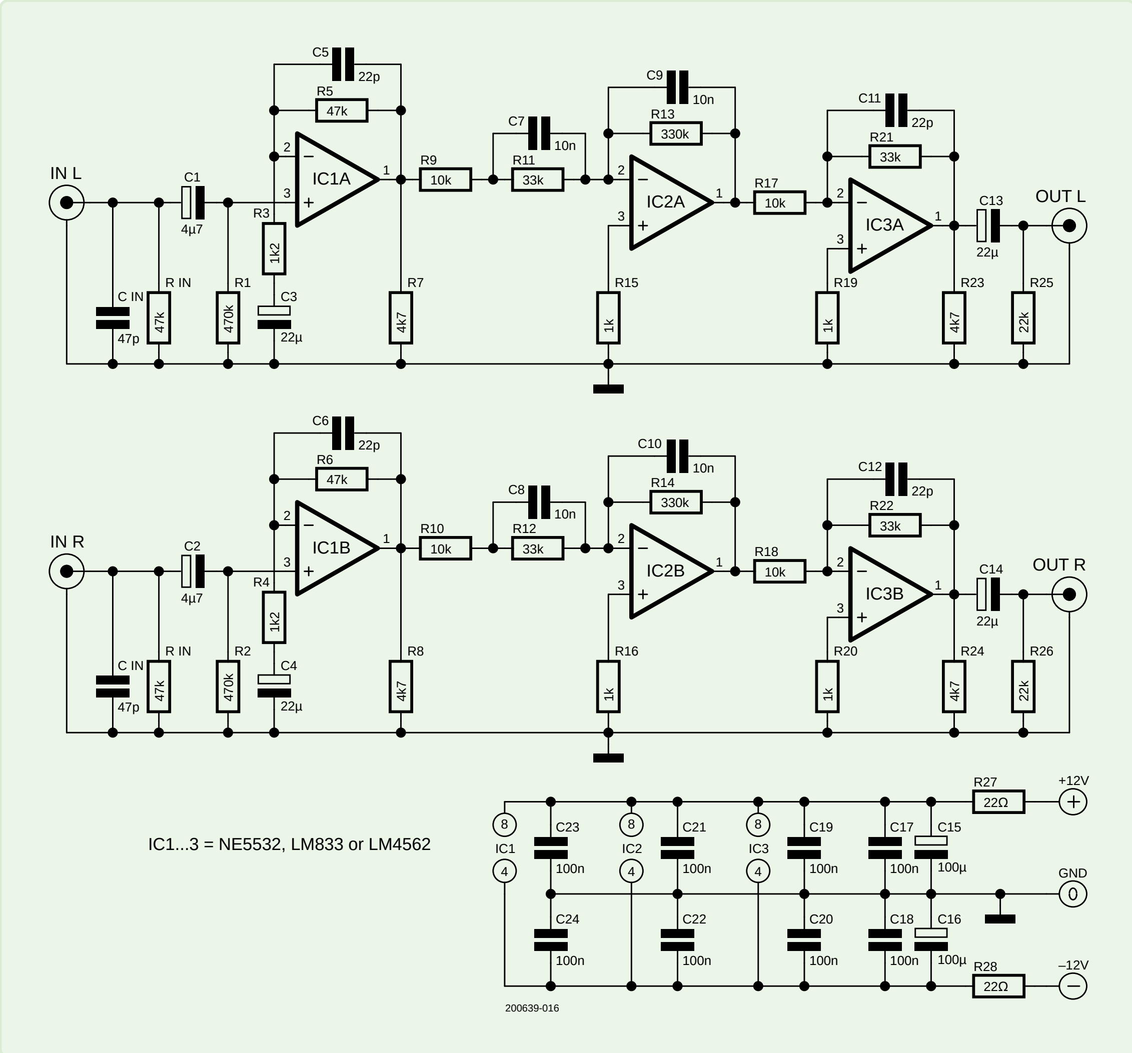


Figure 1: Circuit diagram of the HQ RIAA Preamplifier.

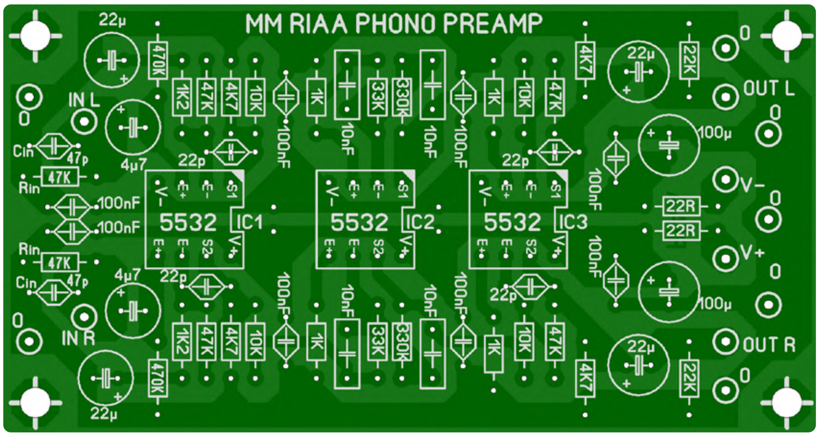


Figure 2: Layout of the solder side of the PCB — in its latest version — featuring an enforced noise filtering circuit.

For the output stage, the gain is 10 dB, or 20 log 33 kΩ/10 kΩ. This gives us an overall gain of 38 dB. The gain can be adjusted, as some cartridges can sometimes output a surprisingly high signal level. This was undoubtedly my case, with an Ortofon pickup cartridge.

In the design, I have included the usual pull-down resistors, R1 and R25, so that the inputs and outputs are not floating.

R27, C15, C17 and R28, C16, C18 enforce the power supply noise rejection capabilities of the project. The three NE5532 ICs are decoupled by C19 to C24. **Figure 3** shows the silkscreen side (top) and the solder side (bottom) of the latest version of the PCB.

The prototype of this HQ RIAA Preamplifier is shown in **Figure 3** (without ICs). As anticipated, for the critical sections of the circuit, make sure to use metal film, low-noise resistors with a tolerance of ±1%.

It's Time to Play a Record!

All in all, the circuit is simple to realize and inexpensive, offering an unrivalled quality, compared to the budget required! Now you're ready to find some old sounds to illustrate your audio productions or fuel your podcasts!

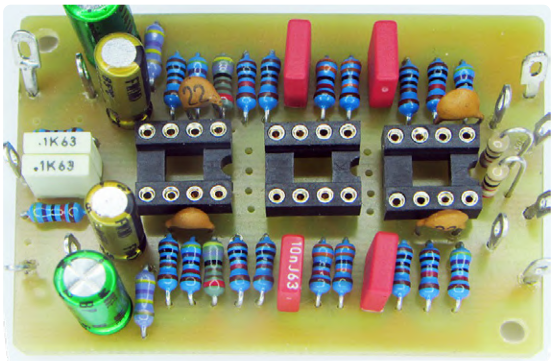


Figure 3: The finished prototype of the preamplifier, without the ICs.

You may not want to stop at the RIAA preamp and maybe like to restore turntables suffering from AC control issues with the existing relays. You could replace them with SCR-based opto-couplers. The MOC3041 by ON Semiconductor, for example, is also offering the "zero-cross" feature, that will prevent any insertion current peak and the startup noise caused by the self-inductance of the AC motor.

To ensure the best performances, you might consider inserting this design in a small metal housing, to get rid of any potential interference source, and add your own personal touch to make your dream come true.

200639-01

Component List

Resistors

(All 0.25 W, Metal Film, 1%)
RIN (L&R) = 47 kΩ
R1, R2 = 470 kΩ
R3, R4 = 1.2 kΩ
R5, R6 = 47 kΩ
R7, R8, R23, R24 = 4.7 kΩ
R9, R10, R17, R18 = 10 kΩ
R11, R12 = 33 kΩ
R13, R14 = 330 kΩ
R15, R16, R19, R20 = 1 kΩ
R21, R22 = 33 kΩ or 47 kΩ (*)
R25, R26 = 22 kΩ
R27, R28 = 22 Ω
(*) (for gain adjustment, if needed)

Capacitors

CIN (L&R) = 47 pF, Ceramic
C1, C2 = 4.7 µF, 25 V, Radial
C3, C4, C13, C14 = 22 µF, 25 V, Radial
C5, C6, C11, C12 = 22 pF, Ceramic
C7...C10 = 10 nF, Polyester
C15, C16 = 100 µF, 25 V, Radial
C17...C24 = 100 nF, Polyester

Semiconductors

IC1...IC3 = NE5532P, Texas Instruments

About the Author

As an electronics technician, Thierry Clinquart discovered the famous µA741 operational amplifier in 1980 during his studies at the Don Bosco Institute in Tournai (Belgium). At the time, it was so much easier to create an audio setup than with transistor technology. He has followed the evolution of this ancestor through the TL071, NE5534 up to the present day with "audio grade" products from Texas Instruments, Analog Devices, JRC, THAT Corp, etc. All the circuits Thierry presents on Elektor Labs are linked together to create customized modules. To reduce wiring, he fits Neutrik A-series connectors directly on PCBs, using Sprint Layout software to optimize their design and to ensure a consistent packaging.

Questions or Comments?

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

Related Products

- **Elektor Audio Collection (USB Stick)**
  www.elektor.com/19892
- **Paul Hetrelezis, *Retro Audio* (Elektor, E-book)**
  www.elektor.com/18207

WEB LINKS

[1] Elektor Labs page for this project: https://tinyurl.com/384vwbcw
[2] Thierry Clinquart, "Small Audio Mixer," Elektor Circuit Special 2025: https://www.elektormagazine.com/240262-01
[3] Thierry Clinquart, "Jack In and Jack Out," Elektor Circuit Special 2024: https://www.elektormagazine.com/magazine/elektor-350/63012

09 Turntable Speed Calibrator

An Arduino-Based 100–120 Hz Strobe Light Generator

By Peter Simoons (The Netherlands)

Adjusting the speed of turntables that do not have an internal calibration device has become problematic since the advent of high-frequency LED lighting in homes. This simple Arduino-based project brilliantly solves the problem. Let's see how!

Some time ago, I was asked to adjust the speed of a record player. Someone had tampered with the adjustments, for obscure reasons. Theoretically, this should have been an easy job. But in practice, I had to tackle two problems:

- The light sources in my house were all LED-based, which, being powered at high frequency, didn't have the required 100 Hz brightness modulation.
- I did not have a stroboscope disk anymore.

I started facing the issue of the lack of a 100-Hz-modulated light source in my house. Since I had some hardware in front of my nose, as a leftover from another experiment, I used an Arduino board, (that could either be a UNO R3 or Nano R3 type) with a 5-mm white LED and a few other components.

I could have searched the house for another usable, old-fashioned, incandescence-based light source, but this immediate solution was easier. Aside from a small power bank, this design may become a mains-independent hand-held light source. Due to the square-wave 100% modulation of the LED's power, the reading of an optical target is much clearer than with any ordinary tungsten lamp.

Circuit

The basic schematic diagram of this project is shown in **Figure 1**. It's all about an Arduino board — used to generate the needed 100 to 120 Hz pulses — driving Q1, an N-channel BS170 FET by Onsemi. R1 limits the peak current flowing through the white LED1, whilst R2 ensures the correct turn-off of Q1.

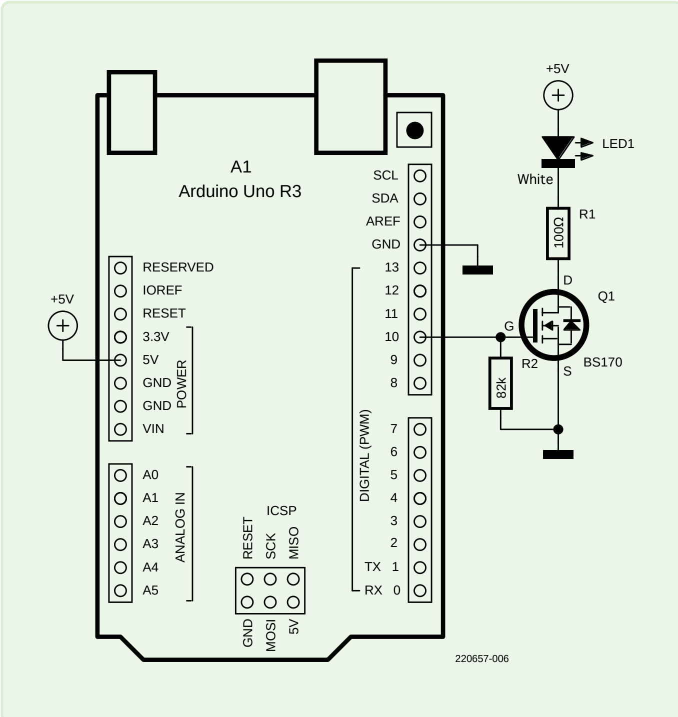


Figure 1: Circuit diagram of the Turntable Speed Calibrator.

To reduce the 16-MHz Arduino clock to 100 Hz light pulses, a division by 160.000 is required. A first division by 16 yields 1 MHz, then 4 times a division by 10 yields 100 Hz. A division by 10 is a division by 2 plus a division by 5. So, somewhere, we need to divide by 16 (2^4) and by 625 (5^4), which cannot be done by an 8 bit counter/timer. So, the use of the 16 bit timer/counter of the Arduino is required. The alternative is a software division, but it is much simpler to let the hardware do the job.

The Arduino will produce a 100 Hz pulse with a duty-cycle of approximately 25%, and that works fine with my disks (see next paragraph). The resulting Arduino sketch has a four-line setup part and an empty loop. It uses some registers of the ATmega328 [1] on the Arduino directly (see **Listing 1**).

Listing 1: Arduino-Code for 100 Hz.

```
#define LED_WHITE 10

void setup()
{
  pinMode(LED_WHITE,OUTPUT);

/*
 * init timer 1
 * - prescaler = /64 --> 16 MHz --> 250 kHz
 * - phase and frequency correct --> 125 kHz
 * - counts to 1250 for 100 Hz
 * - OCR1B set to 312 for 25% duty cycle
 *
 * CS1x = 011    -- divide by 64
 * WGM1x = 1000  -- Phase and frequency correct PWM,
 *               -- counter endvalue in ICR1
 */

  TCCR1A = (1 << COM1A1) | (1 << COM1B1);
      // WGM1x = xx00; COM1ax = 10; COM1bx = 10

  TCCR1B = (1 << CS10) | (1 << CS11) | (1 << WGM13);
      // WGM1x = 10xx; CS1x = 011

  ICR1 = 1249;        // 1250 - 1
  OCR1B = 312;

  return;
}

void loop()
{
  return;
}
```

Stroboscopic Patterns

So, the first issue was solved. Then the time came to find a suitable optical target for our new LED-based stroboscopic calibrator. A quick search on the Internet suggested to me that stroboscope disks were still available, but at considerable costs. The prices started at about €10. Definitely too much, for a one-time job.

Therefore, I decided to create my own targets with the help of a software tool that proved to be quite useful: Galva 2.0 (now 3.0), a freely downloadable French program that you can find at [2]. This software is released for free, for non-commercial use. The tool, created to draw analog measuring instrument scales, was perfect for creating the necessary patterns. However, its use is not immediately intuitive and requires a minimum understanding of the various commands, to obtain a satisfactory result.

For this reason, in addition to the Galva program source files for stroboscope disks for the various speeds, I have prepared ready-to-use PDF versions. For normal, daily use, the *Stroboscoop_45_33.pdf* will do, see **Figure 2**. It contains the patterns for the adjustment of 33 1/3 RPM and for 45 RPM. Another file, *Stroboscoop_78_16.pdf* contains targets for the antique 78 RPM records and the ultra-rare 16 2/3 RPM ones. A stroboscopic disk suitable for all four speeds is also available (see **Figure 3**).

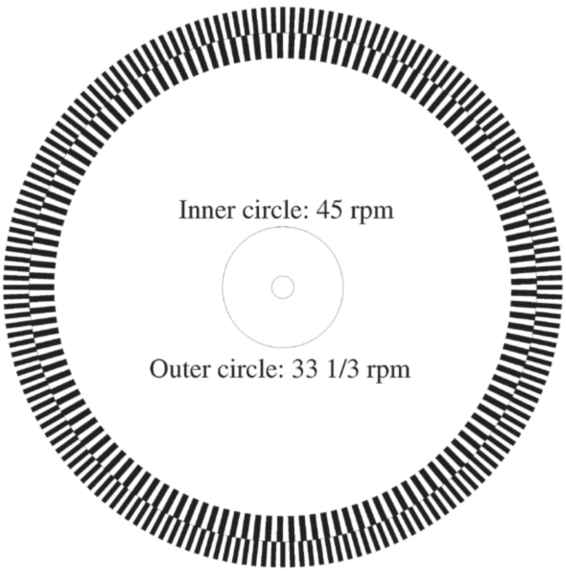


Figure 2: Stroboscopic disk pattern for 45 RPM and 33 1/3 RPM, 50 Hz.

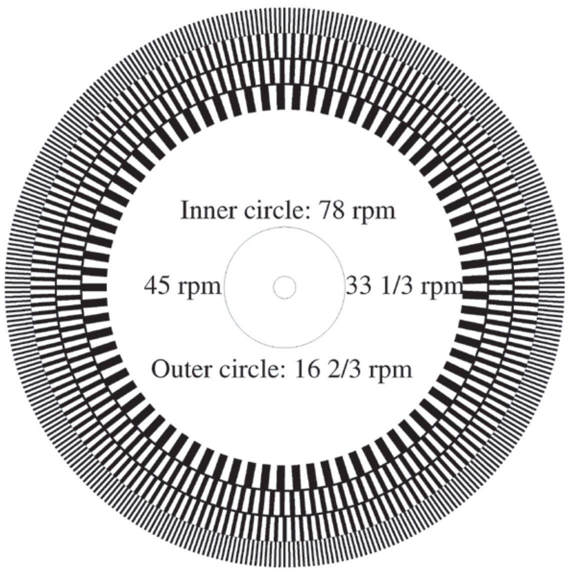


Figure 3: All-speed, combined patterns for 16 2/3, 33 1/3, 45 and 78 RPM records.

After printing "at real size," there are two circles in the middle of the disks to help with creating the proper spindle hole: 7 mm for the standard hole and 38 mm for the large spindle adaptor.

Please note: These disks only work with light that is modulated at 100 Hz, such as the emission of a traditional tungsten (or fluorescent) lamps in a 50 Hz AC mains. (Both the negative half and the positive half-wave of the mains cause a higher level of light; hence the light modulation has a frequency of 100 Hz.) For those who live in a 60-Hz area, I've also prepared a four-speed stroboscope disk — *Stroboscoop_60Hz.pdf* — and a modified version of the Arduino program that creates the related 120-Hz light pulses.

This design worked great, and the turntable could easily be adjusted to run at the correct speed. The disks in Galva and PDF format (50 Hz or 60 Hz version), as well as the two different *.ino* sketches for the Arduino board (100 Hz or 120 Hz output), are available for download at the Elektor Labs pages for this article ([3] and [4]).

220657-01

Questions or Comments?

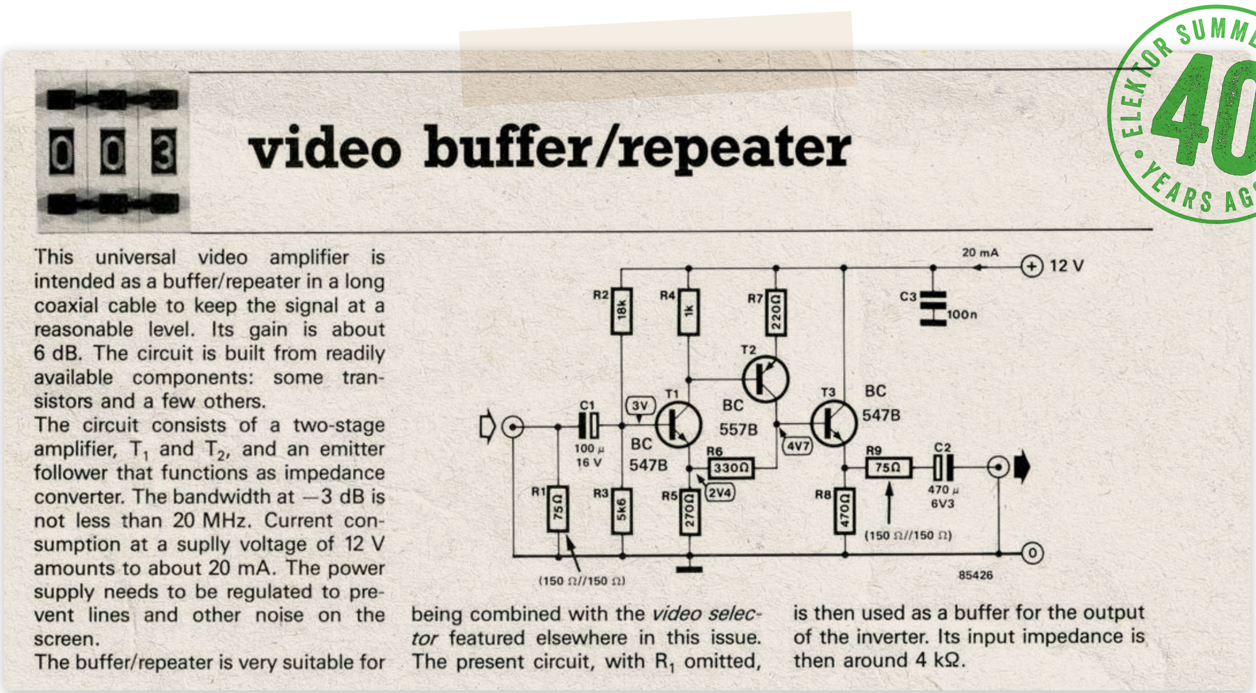
Do you have technical questions or comments about this article? You may contact the author at peter.simoons@kpnmail.nl or the editorial team of Elektor at editor@elektor.com.

About the Author

Peter Simoons (1950) studied at the University of Twente. Thereafter, he mainly worked with SCADA (Supervisory Control And Data Acquisition) systems and data historians. After retirement in 2015, he picked up the Arduino to play with. It looks like the first of his professional projects, which involved assembly language programming of an Intel 8080-based microprocessor system: Simple hardware, no operating system. Only a different language. Peter has been a subscriber to Elektuur/Elektor since 1966.

WEB LINKS

[1] ATmega328P datasheet, Microchip: https://tinyurl.com/ATmega328P-datasheet
[2] Galva SW webpage, Jean-Paul Gendner: https://www.f5bu.fr/galva-download/
[3] Elektor Labs page for this project (part 1): https://tinyurl.com/4cjycena
[4] Elektor Labs page for this project (part 2): https://tinyurl.com/mryd5njw



003

video buffer/repeater

This universal video amplifier is intended as a buffer/repeater in a long coaxial cable to keep the signal at a reasonable level. Its gain is about 6 dB. The circuit is built from readily available components: some transistors and a few others.

The circuit consists of a two-stage amplifier, T_1 and T_2, and an emitter follower that functions as impedance converter. The bandwidth at −3 dB is not less than 20 MHz. Current consumption at a supply voltage of 12 V amounts to about 20 mA. The power supply needs to be regulated to prevent lines and other noise on the screen.

The buffer/repeater is very suitable for being combined with the *video selector* featured elsewhere in this issue. The present circuit, with R_1 omitted, is then used as a buffer for the output of the inverter. Its input impedance is then around 4 kΩ.

Infrared Remote-Controlled Dimmer

Control Your Halogen or LED Floor Lamp Effortlessly and with Style

Figure 1: General view of the PCB.

By Denis Cuynat (France)

This dimmer, specially designed for floor lamps, allows you to control the lighting remotely with the press of a button, no matter where you are in the room. All of this, using any infrared remote control you already have!

No more getting up to switch the lamps on or off: Control them with ease from your couch. In addition to classic dimmer functions, this lighting controller offers two adjustable favorite lighting levels and a delayed turn-off function, ideal for falling asleep peacefully.

In total, six commands are supported: *On/Off*, *brightness down*, *brightness up*, *favorite preset 1*, *favorite preset 2*, and *delayed turn-off* (10 minutes by default). The dimmer does soft transitions between on and off states. With the rotary encoder version of this project (**Figure 1**), the minimum and maximum light levels can be set by the user; in the version without the rotary encoder, these settings are set in the code. This project can use 120 V or 230 V mains voltages, with automatic detection of either 50 Hz or 60 Hz frequency. It supports halogen lights as well as dimmable R7S LED bulbs; I had success with the model number 118150 from Osram.

Reuse Your Remote

To control the dimmer, a great variety of infrared remotes can be used thanks to the learning capability, which makes it compatible with several protocols. These include Pulse Distance Modulation (PDM) from 6 to 48 bits, which covers NEC, JVC, RCA, Mitsubishi, Panasonic, and others. It also supports Sony's Pulse Width Modulation (PWM) in 12-, 15-, and 20-bit formats, as well as Philips RC5 and RC6 protocols, which use Manchester encoding [1].

In PDM mode, pause duration determines bit values, while pulse duration remains constant. In PWM mode, the pulse duration determines bit values, while pause duration remains constant. The carrier frequency of remote controls is typically around 38 kHz (Philips uses 36 kHz, NEC 38 kHz and Sony 40 kHz). A minor mismatch of ±2 kHz slightly reduces receiver sensitivity but the transmission still works fine over a few meters.

Overview

As shown in **Figure 2**, the circuit is relatively simple. Its core component is the Microchip PIC16F1703 microcontroller, which includes a Zero Crossing Detection (ZCD) module. This feature eliminates the need for an external zero crossing detector circuit made of a diode bridge and an optocoupler, simplifying the circuit to just two resistors (R_{series}, R_{bias}, more on this below) and a capacitor (C6). The microcontroller is powered by +5 V via a 7805 voltage regulator. The PIC is using its internal 4 MHz clock, offering an instruction cycle time of 1 µs, which is handy for counting time using loops in assembly language. The accuracy and stability of this internal clock are good enough for this application.

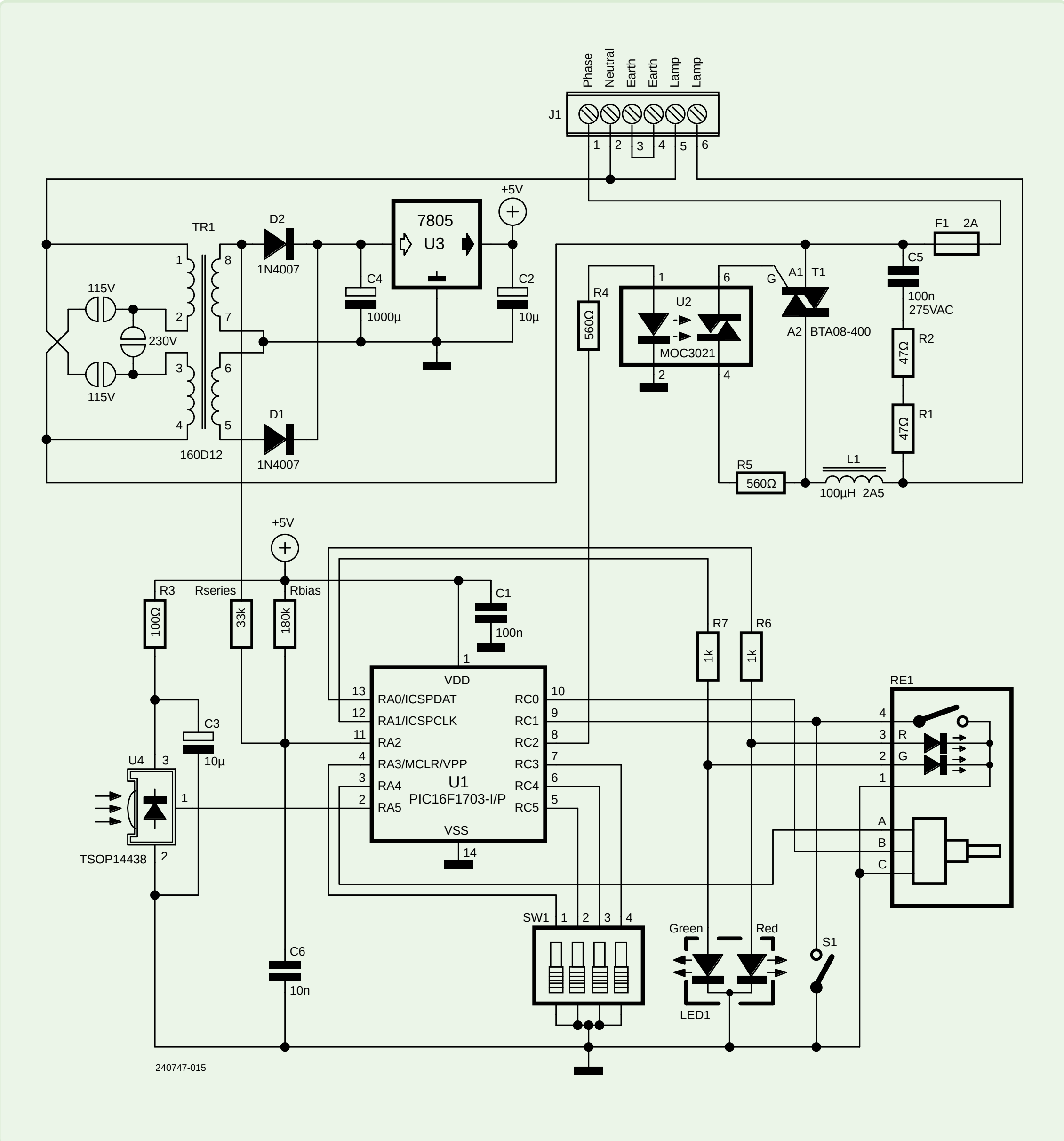


Figure 2: Schematic is made simpler by the PIC's built-in Zero Cross Detector.

The rotary encoder RE1 integrates two LEDs illuminating its axis and a push button. A "Lite version" of the dimmer is also available, removing the rotary encoder in favor of a push button paired with a bi-color LED.

For safety, the high-voltage and low-voltage sections are isolated by the transformer TR1 and the optocoupler U2 (a MOC3021 with Triac output). The Triac is protected against false triggering caused by noise on the mains supply by a classic RC snubber, composed of resistors R1 and R2 and capacitor C5. The resistors are connected in series to ensure that the voltage across each resistor remains well below the maximum allowable voltage for this type of component. On its part, inductor L1 is used to limit the rise time of the current when the Triac is fired. Without this inductor, mains current harmonics would be produced at frequencies high enough to cause considerable interference problems.

The infrared receiver U4 takes care of receiving the pulses from the IR remote control. I've used a TSOP14438 from Vishay, but others are also compatible (see below). These receivers are often sensitive to power supply noise, so manufacturers recommend to filter the power rail. Here, this is done with resistor R3 and capacitor C3.

An array of four switches in a DIP package, SW1 to SW4 (marked SW1 on the schematic for simplicity) can be used to set options, but is not strictly necessary. For example, SW1 enables filtering in case an IR receiver of the TSOP17xx family is used; SW3 enables a lower resolution mode for the dimming which allows for faster reaction times; SW4 disables the LEDs. Details can be found in the source code, which could also be modified according to other needs.

Dimming in detail

The dimmer is of the leading edge type (see **Figure 3**). Light intensity is determined by the delay between the AC signal's zero-crossing and the Triac's activation. The longer the delay, the lower the brightness. Zero-crossing detection is managed by the ZCD module integrated in the PIC, via the R_{series} resistor, which limits the current to 300 µA. The delay is controlled by `Timer2`.

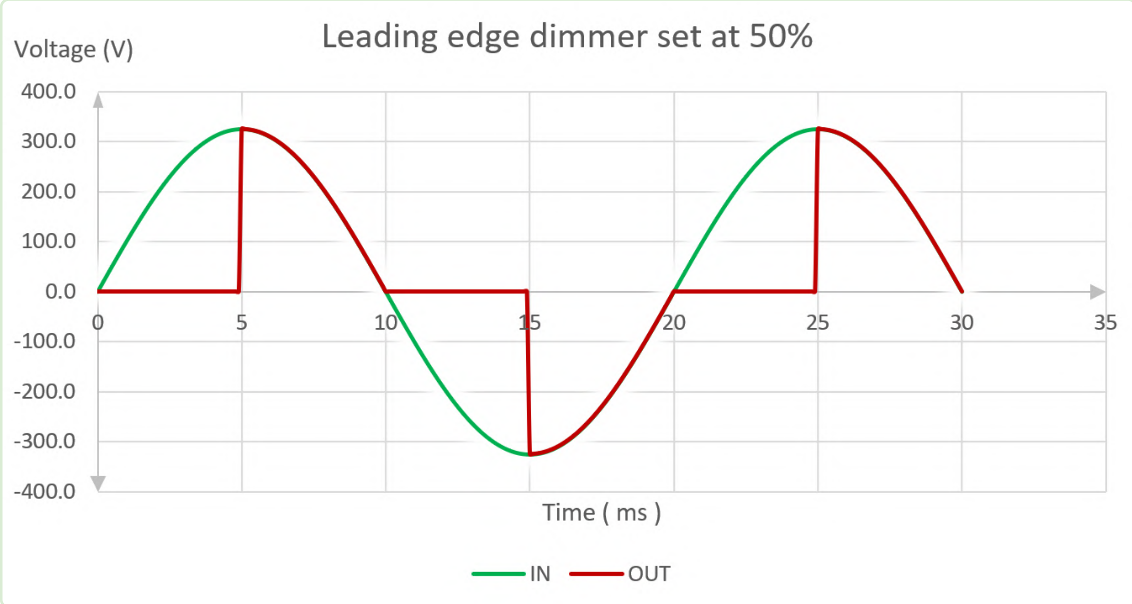


Figure 3: Principle of a Leading Edge dimmer.

According to the Zero-Cross Detection Module Technical Brief from Microchip [2], the ZCD triggers at a voltage V_{CPINV} (typically 0.75 V), not at 0 V. The voltage offset from 0 to V_{CPINV} causes the zero-cross event to occur too early as the V_{IN} waveform falls, and too late as the V_{IN} waveform rises. The actual offset time produced can be calculated using the equations given in the technical brief. At 50 Hz, T_{offset} is 231 µs. **Figure 4** illustrates this delay. This offset is compensated by a bias resistor (R_{bias}) to adjust the detection threshold to 0 V.

The values for the R_{series} and R_{bias} are calculated as per [2], as follows:

R_{series} [kΩ] = V_{peak} / 0.3

Here, V_{ac} = 7.3 V and V_{peak} [V] = $V_{ac} \times \sqrt{2}$ = 10.32 V

So, R_{series} = 10.32/0.3 = 34.41 kΩ. The closest E12 value is chosen, R_{series} = 33 kΩ.

For the biasing resistor, the formula is R_{bias} [kΩ] = (V_{cc} – 0.75)/I_{bias}, where I_{bias} [mA] = 0.75/R_{series} [kΩ]. With V_{cc} = 5 V, the formula gives R_{bias} = 187 kΩ. Here again an extreme accuracy is not needed, the closest common value of 180 kΩ will be fine. The capacitor C6 (10 nF) forms a low-pass filter with R_{series}, preventing high-frequency noise from affecting detection. The Technical Brief from Microchip is very interesting and I recommend having a look at it for more detail.

Software

The program is written entirely in assembly language, developed using the Microchip MPLABX v6.00 IDE. Despite using assembly, the code

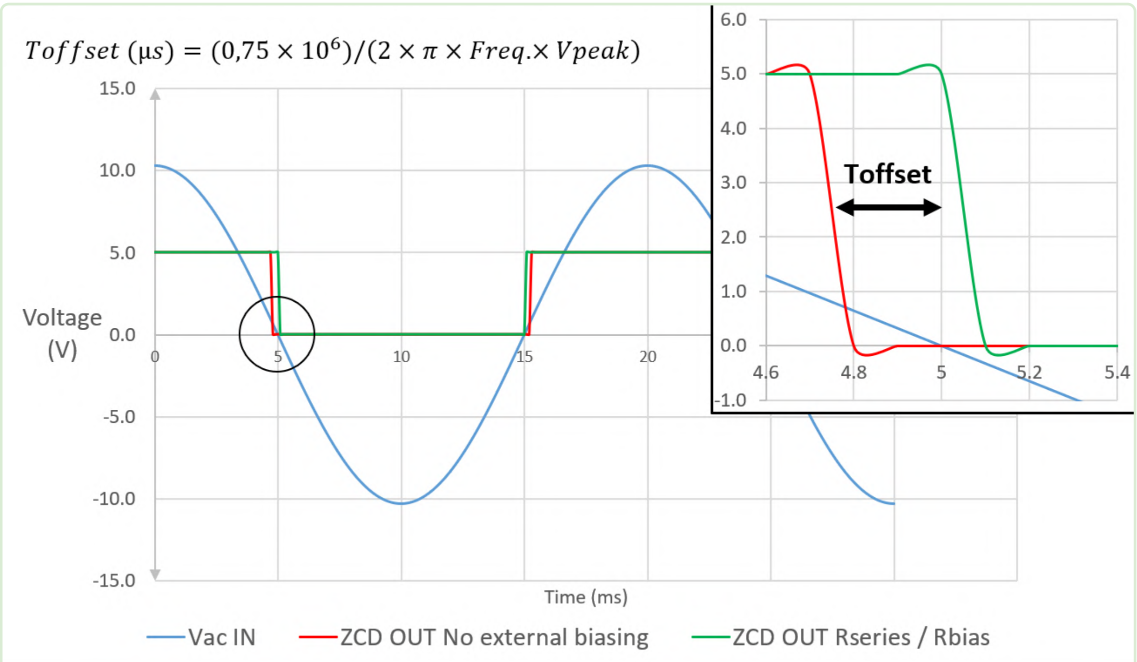


Figure 4: Offset correction of the Zero Cross detector.

remains structured, clear, and well-commented, making it accessible for those familiar with PIC microcontrollers. The full source code and compiled HEX files are available on the Elektor Labs page of this project [3].

The `Protocol_Analysis` and `Pulse_Pause_Analysis` routines measure pulse and pause durations or the remote control signal, using `Timer0` and store them in SRAM. This analysis determines the encoding type, the bit count, the bit duration averages for PDM and PWM, and the RC5/RC6 protocol verification.

Once the protocol is identified, the corresponding routine (`PDM_IR1`, `SIRC_IR1`, `RC5_IR1`, `RC6_IR1`) decodes the signal into a 32-bit format corresponding to remote key functions. All parameters and codes are stored in a dedicated flash memory area.

Building the Dimmer

The dimmer is easy to make, featuring a single-sided PCB with mostly through-hole components. The only two exceptions are the capacitor C6, which is SMD to save space and allow it to be positioned as close as possible to pin 11 of the PIC, and the tactile switch S1 which is used in the Lite version. The infrared receiver footprint accommodates multiple models in the TSOP12, TSOP14, TSOP17, TSOP18 or TSOP22 series, or the VS1838B (**Figure 5**). When using a halogen bulb, a heatsink is necessary for the Triac; this is not required when using dimmable LED bulbs which use much less current. Choose the operating mains voltage by soldering the appropriate solder bridges on the copper side of the PCB, which puts the two halves of the transformer's primary either in series or in parallel.

When using very low-power dimmable LED bulbs, the current in the Triac is very low, hence its circuitry can be simplified. R1, R2 and C5 can be removed, and the inductor L1 can be replaced with a piece of wire.

After checking that components are oriented and soldered properly, and checking the 5 V power supply voltage on the PIC's pins 1 and 14, the PIC can be programmed either using a development board or directly on the PCB using a programmer such as the PICkit 3 programmer and the Microchip IPE software [4]. For the *Lite* version, minimum and maximum brightness levels can be adjusted in the assembly code (`Offset_Value` and `Max_Value` constants). If flickering occurs at maximum power, increase `Max_Value`. To configure the dimmer and for help using it, refer to the user manual described in the text frame.

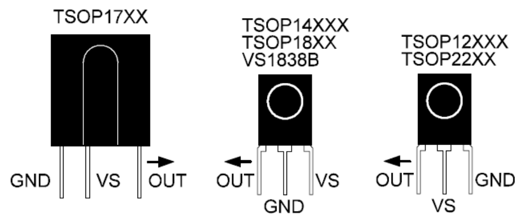


Figure 5: Various infrared receivers can be used.

User Manual

Remote Control Learning

- Press the push button until the LED turns green (press again to exit).
- Press a button on your remote until the LED turns red.
- Wait for the LED to turn green again. If it stays red, the protocol is not recognized, and the dimmer reboots.
- Press the remote button that will serve as *On/Off*; the LED turns red.
- Wait for the LED to turn green again.
- Press the remote button that will be used for the dimming function; the LED turns red.
- Wait for the LED to turn green again.

Repeat steps 6 and 7 for the *Brighten*, *Favorite 1*, *Favorite 2*, and *Delayed Turn-Off* buttons. At the end of learning, the LED stays red, and the dimmer reboots.

Configuring Favorite Lighting Levels

- Press the *Favorite 1* button until the LED turns red.
- Adjust brightness with the *Brighten/Dimming* buttons.
- Press the *Favorite 1* button again to save.
- Repeat for *Favorite 2*.

Configuring Minimum/Maximum Levels (rotary encoder version)

- Press the push button until the LED turns red.
- Adjust minimum brightness with the rotary encoder.
- Press to save.
- Adjust maximum brightness (without flickering).
- Press to save. The dimmer reboots automatically.

Figure 6: The dimmer unit installed on the lamp post.

Mechanical Build

This will naturally depend on the type of floor lamp you intend to use. In this case, a 105×80×32.5 mm enclosure from Hammond [5] was used, and mounted halfway up the lamp post so that the infrared receiver would be easy to aim at with the remote control. The infrared receiver sits behind the front panel made of dark acrylic, which is transparent to these wavelengths. The lamp post tube had an opening cut into it to route the wires; the result can be seen in **Figure 6**. Readers wishing to replicate the setup may need to adapt to suit different lamp models.

Compatibility and Speed

Out of approximately thirty remotes that were tested, only one, using the Nokia protocol, was incompatible with this dimmer. The NEC protocol is by far the most common. For best results, Sony remotes are recommended due to their speed. Panasonic remotes, using a 48-bit protocol without a repeat bit, are much slower and should be avoided. While it should be compatible, the dimmer has not been tested at 110 V mains voltage; the values of the components near U2 and T1 may require adjustment.

Overall, this custom dimmer was a fun project to design and program, and the result turns out to be intuitive and easy to use. I'm quite satisfied with the outcome of the mechanical integration to this floor lamp, which is both functional and visually pleasing. I hope this overview has *shed some light* on this project! Find the related files on the Elektor Labs page of this project [3].

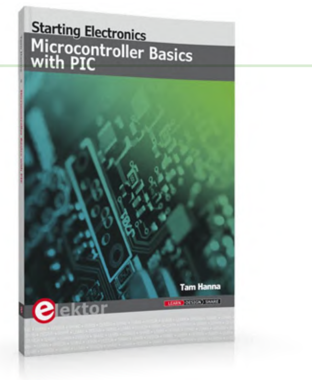
240747-01

Questions or Comments?

Do you have questions or comments about this article? Visit the Elektor Labs page of this project [3], or contact Elektor at editor@elektor.com.

Related Product

- **Tam Hanna, *Microcontroller Basics with PIC* (Elektor, 2020)**
  www.elektor.com/19188



Component List

Resistors (0.25 W)
R1, R2 = 47 Ω
R3 = 100 Ω
R4, R5 = 560 Ω
R6, R7 = 1 kΩ (rotary encoder version), 560 Ω (lite version)
Rbias = 180 kΩ (see text)
Rseries = 33 kΩ (see text)

Capacitors
C1 = 100 nF, 50 V, MLCC, 5.08 / 2.54
C2, C3 = 10 µF, 25 V
C4 = 1000 µF, 25 V
C5 = 100 nF, Class X2, 275 V AC
C6 = 10 nF, X7R, 0805

Inductors
L1 = 100 µH, 2.5 A (e.g. Würth 7447021), 21 mm max diameter

Semiconductors
D1, D2 = 1N4007
LED1 = Red / Green, 5 mm (lite version)
T1 = BTA08-400BRG
U1 = PIC16F1703-I/P
U2 = MOC3021
U3 = 7805 or 78L05
U4 = TSOP14438

Miscellaneous
F1 = Fuse, 5×20 mm, 2 A
HS = Optional TO-220 heat sink, 23×20×15 mm
J1 = connector, TE Connectivity 1546074-6, 5.08 mm pitch
Enclosure = Hammond Manufacturing RM2005MTBK
S1 = C&K KSC641JLFS (lite version)
SW1 = Omron A6FR-4101
TR1 = Transformer, Hammond Manufacturing 160D12, Primary 2× 115 V, Secondary 2× 6.3 V, 50/60 Hz 1 VA
RE1 = 12 mm Rotary Encoder, CTS 12CE2N25B24B24, illuminated shaft
Knob = GTP6M-13X16-S

WEB LINKS

[1] Manchester code, Wikipedia: https://en.wikipedia.org/wiki/Manchester_code
[2] Zero-Cross Detection Module Technical Brief (TB3138), Microchip: https://tinyurl.com/2ttp33et
[3] Project page on Elektor Labs: https://www.elektormagazine.com/labs/dimmer-controlled-by-a-universal-infrared-remote-control
[4] MPLAB Integrated Programming Environment, Microchip: https://www.microchip.com/en-us/tools-resources/production/mplab-integrated-programming-environment
[5] Hammond Manufacturing: https://www.hammfg.com/

11 How to Use switch...case on Strings in C++/Arduino IDE

By Xin Wang (Germany)

For remote control projects, a text-based serial protocol is often used. In the code for the receiver, switching to the corresponding commands can be done with switch...case in an elegant way. However, C++, used in the Arduino IDE, doesn't support the switch...case construction with strings. Here is a solution for that problem.

If you have just a few ASCII commands to handle, it's okay to use `if…else if…else`. But if you have many commands, you will lose overview easily. In this case, `switch…case` would be more convenient. Unfortunately, the Arduino IDE doesn't support `switch` with string objects. But with a trick, it is possible.

1. First, you must construct an enumeration with `enum`. An enumeration is a set of named integer constants. The integers are starting with 0, in the following example they go from 0 to 4:

```
enum Index { AB, BC, CD, DE, EF };
```

   As names for the integers in this list, it is convenient to use the commands of the serial protocol to be handled.

2. Then you must construct a string array with the same characters and the same order, for example:

```
#define numberofCommands 5
const String Command[numberofCommands]={ "AB", "BC",
"CD", "DE", "EF" };
```

   The definitions are now finished.

3. In the `loop`, the received command string in the variable `commandString` will be compared to the `Command` array first:

```
whichCommand = 0;
while ((whichCommand <= numberofCommands) &&
      (commandString != String(Command[whichCommand])))
{       // find command in array
            whichCommand++;
}
```

   Here is the trick: the increase of `whichCommand` stops at the position where the command is found in the command array. And `whichCommand` has the same value as the corresponding characters in the enumeration `Index`.

4. Now we can use `switch…case` to handle all commands, as they are represented by integers in the enumeration. `whichCommand` is the index of the command to be actually handled:

```
switch (whichCommand) {
            case AB:
            break;
            case BC:
            break;
            case CD:
            break;
            case DE:
            break;
            case EF:
            break;
            default:
            break;
}
```

250370-01

About the Author

Xin Wang, born in 1973, graduated from the University in Hanover in electrical engineering. After that he went to Munich for promotion and was member of scientific staff at the Technical University of Munich (TUM). Xin is the Winner of a Founders' competition 2006 in Germany. In 2010 he founded Demu Electronics in Shanghai for PCB and assembly, and since then he has been working as CEO.

12 Magnet Finder

With a Simple Hall-Effect Sensor

By Bas Schmidt (The Netherlands)

Build this simple tool and find the hidden magnets and sensors inside modern appliances. Perfect for repairs, tinkering, or just for fun!

The idea for this device came out of practical need, specifically during repair sessions at a Repair Café location, where I often volunteer. More and more home-barista coffee machines are brought in for repair. These often include built-in safety systems that prevent operation if any of the removable parts, like trays or containers, are not properly inserted. Instead of mechanical switches, which wouldn't last long in the humid and powder-filled environment inside such machines, manufacturers use proximity switches made of Hall-effect sensors, with hidden magnets in the removable parts.

Figure 1: The Magnet Finder in action, here detecting a screwdriver magnetizer.

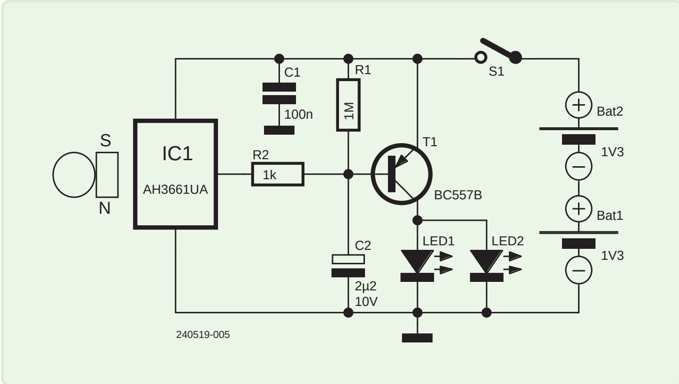


Figure 2: A very minimalistic circuit diagram.

Recently, one of these coffee machines refused to operate, showing a red icon despite all parts seemingly in place. That means either one of the sensors is faulty, or a magnet has fallen out of its proper location, hence the need to find both the sensor and its corresponding magnet! In that particular case, it turned out to be a total nightmare to find these, which were buried in black plastic parts. After this hassle, I decided to build a dedicated magnet finder while keeping it very simple and compact, as you can see in **Figure 1**.

Schematic

The circuit diagram (**Figure 2**) is minimalistic. At its core is an AH3661UA Hall-effect sensor in a TO-92 package. This sensor has an on/off output, and is sensitive enough to pick up small magnets, with a maximum guaranteed sensitivity of ±0.007 T, making it well-suited for this purpose; you can refer to the datasheet [1] for more details. This sensor can be powered from 2.4 to 5.4 V, which allows the whole circuit to be powered directly from two AAA batteries without any regulator. When the sensor detects a magnetic field, its output pin goes low, which switches on a BC557B transistor, which in turn lights up two green LEDs. Hans, a fellow user of Elektor Labs, suggested I add a base resistor to protect T1 and the open collector output of IC1; it is a good suggestion that I added to the schematic, even if I didn't use one in my prototype.

According to the LEDs' datasheet [2], their typical forward voltage is 3 V. Depending on the exact LEDs you use, you may want to add a series resistor to limit the current; in my build, I didn't find this necessary. C1 is used to decouple the power supply near the Hall-effect sensor, U1, and R2 limits the T1's base current.

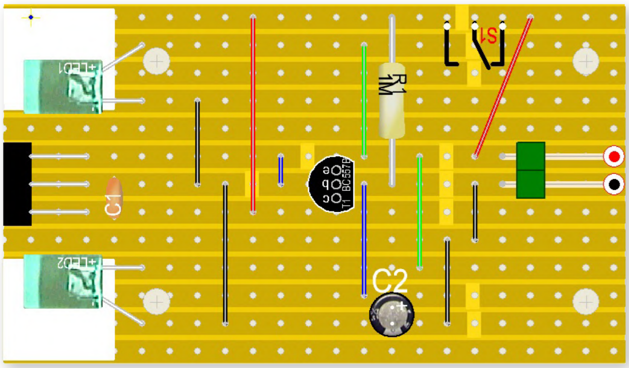


Figure 3: Stripboard layout is aided by LochMaster.

I added R1 and C2 to give a clear visual indication that the tool is working after you power it on. At power-on, C2 is discharged, which holds T1's base low for a brief instant, thus lighting up the LEDs even without any magnet nearby.

Building the Finder

The circuit is so simple that it doesn't require a proper PCB; it was built on a piece of perforated stripboard. A small plastic enclosure, just big enough to fit it together with a dual AAA cell holder, was used with two holes drilled for the LEDs. I used sPlan to draw the schematic, and LochMaster to help with the placement of components on the prototyping board (**Figure 3**); both pieces of software are sold by Abacom [3]. The corresponding files are available on the Elektor Labs page for this project [4]. The result can be seen in **Figure 4**.

Hunting for Magnets

Once powered on, the tool will show you whenever you bring it close to a hidden magnet in one of the plastic trays or elsewhere. Then the position of the Hall sensor in the machine can usually be deduced, and its function tested with a multimeter. Some sensors have a voltage output, while some are just (reed) switches that make contact in the presence of the magnet. Be careful, though: If you want to see if the sensor indeed works by approaching another magnet that you may have, make sure the coffee machine doesn't automatically start a coffee making cycle and burn you as a result.

In the end, this project has proven useful at the Repair Café, and amused my fellow repairmen. The circuit is easy to build, uses only

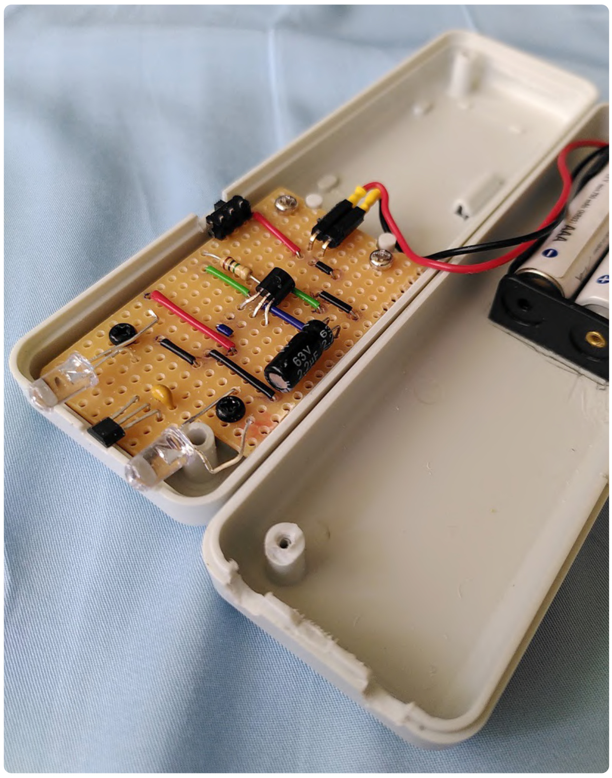


Figure 4: Completed build.

a few components and is a fun and easy afternoon project. While a steel screwdriver could also be used to detect magnetic attraction, this electronic approach is more fun for anyone who enjoys tinkering!

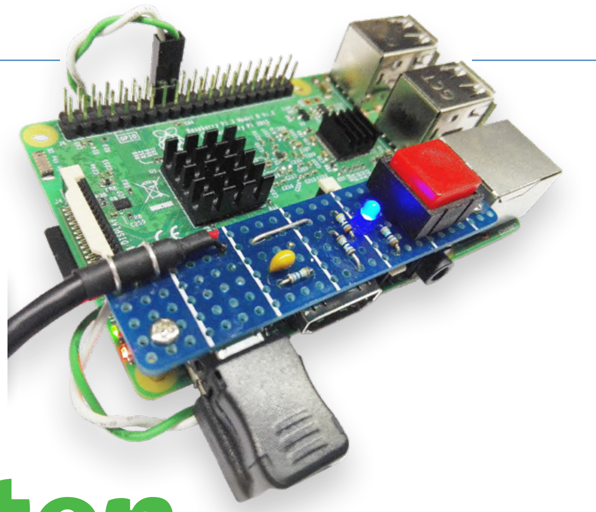
240519-01

Questions or Comments?

Do you have questions or comments about this article? Contact Elektor at editor@elektor.com.

WEB LINKS

[1] AH3661 datasheet, Conrad: https://tinyurl.com/ah3661datasheet
[2] Green LED datasheet, Conrad: https://tinyurl.com/greenledds
[3] Abacom website: https://electronic-software-shop.com/lng/en/electronic-software/
[4] Project page on Elektor Labs: https://elektormagazine.com/labs/magnet-finder



13 Raspberry Pi Smart Power Button

A Solution for Raspberry Pi Up to Model 4

By Antonello Della Pia (Italy)

The success and popularity of Raspberry Pi boards is certainly well-deserved; but over time, many users have noticed and complained about the lack of a power button. Finally, the desired button has been added to the new Raspberry Pi 5 model, but owners of earlier models may find this project useful.

Accessories that add power button functionality to the Raspberry Pi are nothing new. There are both commercial and DIY offerings, from the simple cable with a switch to sophisticated HATs (Hardware Attached on Top). However, the solution that I'm proposing has the advantage of a rather simple wiring diagram and does not require any programming or installation of scripts or automatic startup services.

A few components and the addition of three lines of text to the *config.txt* file make it possible to power on the Raspberry Pi board at the touch of a button, shut down the operating system correctly (even in headless systems), and automatically disconnect the board's power supply without having to unplug the USB connector, thus reducing the current draw to zero.

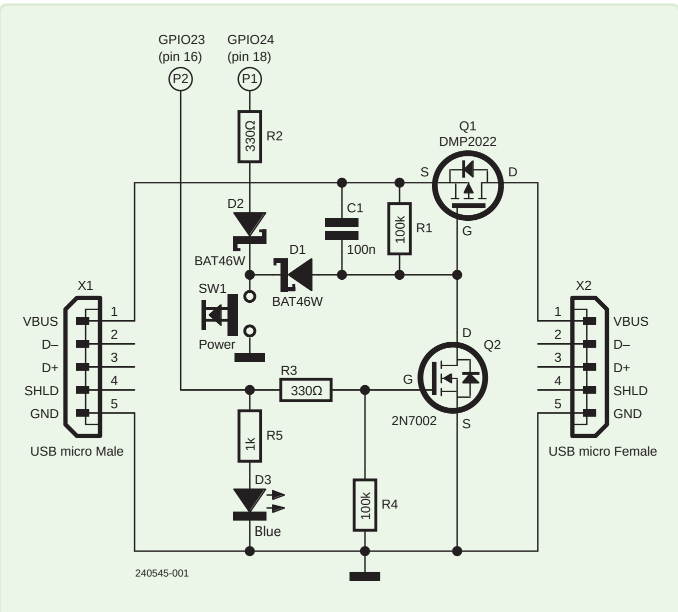


Figure 1: Circuit diagram of the Smart Power button.

Configuring the Pins

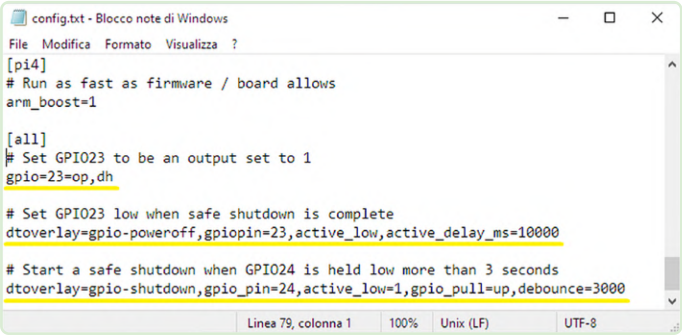
The circuit, illustrated in **Figure 1**, must be installed between a standard power supply and the Raspberry Pi board, and also connected to two available generic GPIO pins on the 40-pin header. Before connecting the circuit, it is necessary to define the behavior of these two pins using a feature introduced in the most recent versions of Raspberry Pi OS.

This feature is known as *Device Tree Overlays*, which allows instructions for hardware operation to be inserted in text form into the *config.txt* file. This file (which is loaded at each boot of the operating system) is located in the *bootfs* partition of the operating system's micro SD card. It is easily accessible even from Windows and can be modified using any text editor. The following three instructions must be added at the end of the file:

```
gpio=23=op,dh
```

This sets GPIO23 to be an output set to 1 (HIGH).

```
dtoverlay=gpio-poweroff,gpiopin=23,active_low,active_
delay_ms=10000
```


config.txt - Blocco note di Windows
File Modifica Formato Visualizza ?
```
[pi4]
# Run as fast as firmware / board allows
arm_boost=1

[all]
# Set GPIO23 to be an output set to 1
gpio=23=op,dh

# Set GPIO23 low when safe shutdown is complete
dtoverlay=gpio-poweroff,gpiopin=23,active_low,active_delay_ms=10000

# Start a safe shutdown when GPIO24 is held low more than 3 seconds
dtoverlay=gpio-shutdown,gpio_pin=24,active_low=1,gpio_pull=up,debounce=3000
```
Linea 79, colonna 1 100% Unix (LF) UTF-8

Figure 2: Example of modified ***config.txt*** *file, including comments.*

This will set GPIO23 low when safe shutdown is complete.

```
dtoverlay=gpio-shutdown,gpio_pin=24,active_low=1,gpio_
pull=up,debounce=3000
```

That starts a safe shutdown when GPIO24 is held low for more than 3 s. GPIO24 is set as an input with pull-up.

An example of a modified *config.txt* file, including comments, is shown in **Figure 2**.

The Circuit

Now the operation of the circuit becomes clearer. By pressing and holding the SW1 button, the MOSFET Q1 (P-channel) is turned on. This powers the Raspberry Pi board and boots the operating system. Within about 1 s, GPIO23 goes high, the blue LED lights up, the MOSFET Q2 (N-channel) goes on and keeps Q1 on and the button can be released.

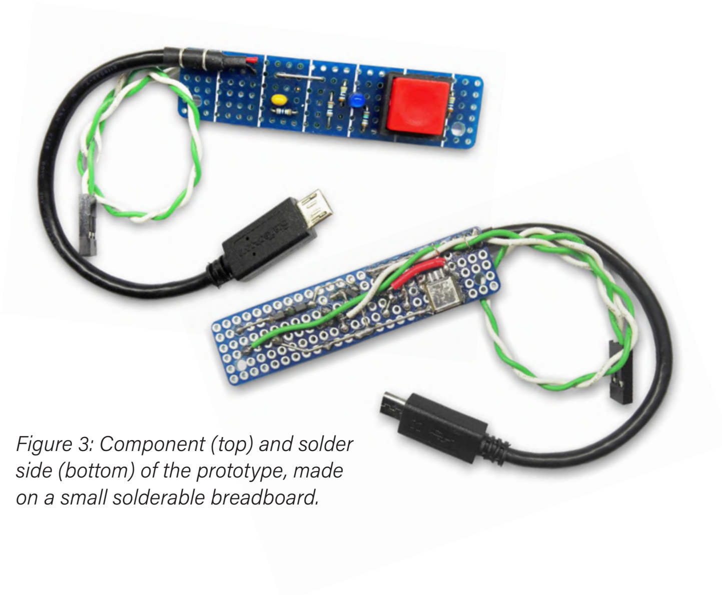


Figure 3: Component (top) and solder side (bottom) of the prototype, made on a small solderable breadboard.

If you shut down Raspberry Pi from the *Start* menu on the OS desktop, at the end of the process (*Safe Shutdown*) GPIO23 will go low, the blue LED, Q1 and Q2 will turn off and the power to the board will be cut. The same effect is achieved by pressing the button for more than 3 s. After this time, GPIO24 activates the *Safe Shutdown* procedure without the user having to access the operating system, which is very useful in headless systems.

To proceed with the schematic description: C1 and R1 prevent Q1 from turning on when the power supply is connected. D1 and D2 protect GPIO24 from the 5 V voltage present at Q1's gate (via R1) when Q2 is turned off. R4 is the pulldown resistor of Q2's gate, and R2 and R3 protect the GPIOs from overcurrent. The MOSFET Q1 was chosen for its low on-resistance value (about 16 mΩ @V_{GS} = -4.5 V) in order to minimize voltage drop and power loss.

The completed prototype is visible in **Figure 3.** It was realized on a small piece of solderable breadboard that can fit close to your Raspberry Pi board.

240546-01

About the Author

Since childhood, Antonello Della Pia was attracted to electricity and electronic devices. He holds an Electrical Engineering Technician high school diploma. Antonello has always cultivated and developed his passion for analog and digital electronics. Currently, he plays around with microcontrollers and programming, trying to improve his computer skills. Antonello likes to develop and propose projects that are as original as possible and — as he hopes — interesting, as well.

Related Product

- **Günter Spanner, *Raspberry Pi 4 OR 5 AND Pico* (E-book, Elektor, 2024)**
  www.elektor.com/20829

WEB LINK

[1] Elektor Labs page for this article: https://www.elektormagazine.com/labs/raspberry-pi-smart-power-button

Essential Maker Tips

Professional Insights for Everyday Making

Compiled by Brian Tristam Williams (Elektor)

The maker community thrives on shared knowledge — those hard-won tricks and techniques that transform frustrating projects into successful builds. Elektor reached out to leading makers and influencers to gather their most valuable tips, covering everything from PCB design strategies to component selection, workspace organization to advanced techniques. These insights represent years of collective workshop experience distilled into practical wisdom!

DIY GUY Chris

Fundamentals of Efficient Making

Desktop Essentials: Always keep a small magnifier (7× or 10×) on your desk, right next to your tweezer tool set. It's essential for inspecting circuit boards, both during assembly of your own designs and when doing rework on other boards.

Strategic Tool Investment: Continuously invest in advanced tools, especially for inspection and assembly. These tools not only help you work with a wide variety of components but also allow you to try out newer technologies. Consider it an automatic-investment in your future, since you always end up learning something valuable from each upgrade.

Modular Schematic Design: When designing PCBs, structure your schematics in logical blocks. For example, if you're building a battery-powered circuit that reads atmospheric data and uploads it to the cloud, group the atmospheric sensing components together, the MCU setup in a separate block, and the charging and power management circuits in another. Also isolate connectors in their own section. This modular approach allows you to reuse tested functional blocks in future designs.

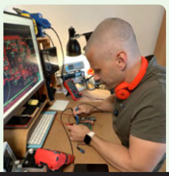
Debug with Test Points: Use oscilloscope and logic analyzer frequently. Regularly check signals, even if they "should work." Sometimes seeing the waveform reveals hidden issues like glitches, noise, or slow edges. Include test points on critical signals (power rails, communication lines, MCU IOs). It makes debugging and validation much easier, especially during prototyping or production testing.

Assembly Timing Strategy: Before applying solder paste, gather the components from your stock and place them on the table in the same layout and orientation as on the PCB. Then apply the solder paste and quickly perform pick-and-place. This minimizes the exposure time of the solder paste to ambient air. Otherwise, placing 100 components one by one after applying paste could take an hour and affect the paste's performance.

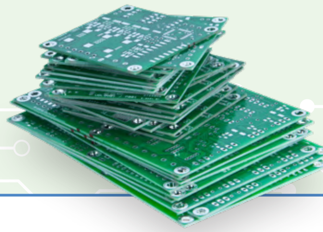
Workspace Organization: Always keep your workspace clean and well-organized. A tidy setup helps you make the most of your limited time in the home lab and keeps the workflow smooth and productive.

https://instagram.com/diyguychris

Source: Adobe Stock



That KiCad Guy (Petr Dvořák)



Source: HandmadePictures / Adobe Stock

Advanced PCB Techniques

Digilent Analog Discovery 2: Whenever I use the AD2 as an impedance analyzer, I feel like a real engineer — a genuine one. The impedance analysis of passive components, whenever you want to learn more about their "behavior," is still advanced. The Digilent Analog Discovery has zillions of modes and use-cases. My absolute favorite is the impedance analyzer.

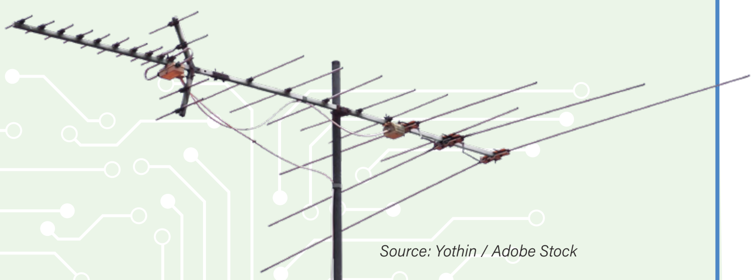
WLCSP Fine-Pitch Layout Trick: For ultra-fine-pitch WLCSP packages, I use a "fence hack." I only route to the perimeter balls — the ones around the outer edge of the chip — and leave the center balls unconnected. This avoids expensive micro vias and vias-in-pads that would normally be required to reach all the balls. The PCB can stay in the low-cost fabrication range when the design allows skipping the inner connections. Nobody's complained so far.

Spring-Loaded Pogo-Pins: These spring-loaded pins are expensive, but without them, your design would be cheap. Cheap because without such posts that enable us to test our devices, we would plan the project to fail. When I work on new hardware devices for my customers, we always discuss manufacturing testing. Even the highest manufacturing yield is lower than 100%. Always. And those few tenths of percent can make hundreds when the volume gets serious.

Interactive BOM for KiCad: One Interactive BOM option will save your life during assembly. By default, the plugin displays all components in your assembly guide, even ones marked as DNP (Do Not Populate). This creates confusion when you're trying to assemble the board because you'll see parts listed that you shouldn't actually install. You can change the plugin settings to hide DNP components so your assembly guide only shows the parts you actually need to solder.

3D-Printed Panel Key: Believe it or not, I didn't have a panel key until now, although I have like 100 screwdrivers. From time to time, I have to open a hallway switch box. I'd always used a big flat screwdriver or a pair of pliers. Instead of buying one, I 3D-printed one. The 80% infill should ensure solid firmness.

https://linkedin.com/in/petr-dvorak-hw



Source: Yothin / Adobe Stock

Baltic Lab (Sebastian)

Philosophy and Fundamentals

Embrace Failure as Learning: Failure is essential to innovation. Most inventors didn't succeed on their first (or second, or third...) attempt. Struggle is part of the process — real understanding comes when you push through it. Each setback offers insight and a learning opportunity. Learn from what went wrong and use it to refine your approach.

Define Your Own Limits: Don't let others decide what's possible for you. The engineering community is always on the front lines of turning the impossible into reality. Even if your goals seem out of reach, aiming high gets you further than settling early. Limits are often illusions — challenge them.

Clean, Stable Power: Power matters. Use clean, stable power sources — batteries if necessary — to eliminate the potential for undesired circuit behavior. Unreliable power supplies create unreliable results. Garbage in, garbage out.

Choose High-Quality Components: Cheap parts often lead to expensive problems. Saving money upfront may cost you more in the long run. Invest in quality parts. It pays off over time.

Don't Be Afraid of RF: Radio Frequency (RF) can seem like black magic to the uninitiated. Its complexity is real, but not insurmountable, even for hobbyists with limited access to specialized test equipment. With study and patience, the "magic" reveals itself as science. Don't be intimidated — intimidation quickly fades when you persist. True insight is earned by working through the hard parts.

https://youtube.com/@BalticLab

BuildYourCNC (Patrick Hood Daniel)

Source: Janceluch / Adobe Stock

Modern Techniques

Embrace Direct Microcontroller Programming: Don't be afraid to dive into subjects that seem intimidating with microcontrollers. Platforms like Arduino or dev boards aren't necessary for prototyping. All you need is the chip and a programmer. If the chip is SMD, get a simple breakout board and solder the chip on that board for breadboard use.

SMD Soldering with Solder Paste: SMD soldering with a traditional iron can be frustrating when trying to stick fine pitch packages to PCBs. The pads want the chip to position itself between them. Use a bit of solder paste sloppily placed on a few pins — the chip can be adjusted easily since the paste has surface tension.

Hot Plate for SMD Assembly: Use a hot plate to solder SMD components rather than an oven with a profile. You can watch the solder paste melt and get that shiny appearance. Hot plates provide better heat distribution. This works for small boards only, since hot plates are typically small. Then drag solder or do touch-ups with a soldering iron and flux.

USB Soldering Irons: The new USB soldering irons are far superior to traditional soldering stations. Consider a Fnirsi USB soldering iron, which heats up in seconds and shows power consumption and temperature. All the old stations I had are sold — USB irons are the future.

Leverage AI for Programming: Use LLMs (Claude, ChatGPT, Gemini, DeepSeek, Llama, Grok, etc.) for microcontroller programming and even creating circuits. All of the code I use has been largely written by LLMs with tweaks as needed. LLMs also make it very easy to understand code and what's going on under the hood.

https://youtube.com/@PatrickHoodDaniel

Cyrob (Phillipe Demerliac)

Practical Workshop Solutions

Short-Circuit Tool Creation: Create a simple short-circuiting tool for debugging. Take flexible silicone wire and two 15 AWG copper wires, strip them to points, and tin them. Use heat shrink tubing for a comfortable grip. This tool is perfect for bypassing components during troubleshooting — like testing whether a transistor or relay is the problem in a circuit.

Salvaging Video Cable Materials: Don't throw away old video cables (VGA, DVI, HDMI, DisplayPort). These cables contain excellent quality wire that's perfect for prototyping. The shielding makes great ground connections, and the individual wires are ideal for breadboard connections. You can recover high-quality copper wire, shielding, and various gauges of wire for different applications.

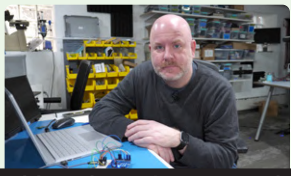
DIY Micro Solder Bath: Create a micro solder bath using a copper adapter on your soldering iron. Drill a hole to fit your iron tip and create a small well for solder. This gives you the benefits of a solder pot for tinning wires without the expense and bulk of a full solder bath. Perfect for preparing wire ends and small components.

Component Tester Protection: Protect your component testers from charged capacitors by creating a simple discharge tool. Use six 2.7 kΩ resistors in parallel on a small PCB, connected between two test probes or clips. Before testing any capacitor with your component tester, first connect this discharge tool across the capacitor's terminals to safely drain any stored charge. This prevents damage to your meter from accidentally testing charged capacitors.

TO-220 Support Mounting: Use proper socket supports for TO-220 packages instead of soldering them directly to boards. This makes testing and replacement much easier while providing better thermal management. Different pin configurations require different socket types, so match your support to your specific component.

https://youtube.com/@Cyrob-org

Source: terex / Adobe Stock



Bald Engineer (James Lewis)

Workshop Efficiency

Blue Masking Tape Magic: Keep a roll of blue masking tape or painter's tape around. The adhesive is strong enough to hold items together temporarily, but weak enough to rarely leave residue behind. It's excellent for holding through-hole parts in place when soldering circuit boards. Combine it with a black or silver marker for high-contrast labels.

Silk-Screen Documentation Blocks: Add a silk-screen block to PCB designs for notes. The size can vary based on board space available. Use these blocks to identify different variants of a prototype build. For example, if you build three copies of a board with a microcontroller, you can easily track which board has what firmware or hardware bodges.

Zona Hobby Knife Handle: The Zona Hobby Knife Soft Grip Handle is an inexpensive handle compatible with brand-name knife blades. It has robust blade retention and release mechanism, and the soft grip is a pleasure to hold while doing detailed work. It's fantastic for cleaning up 3D-printed models or cutting PCB board traces.

PanaVise Speed Handle: If you have a PanaVise Jr (PV-201), the standard knob is fine for minor adjustments but an absolute nightmare for large adjustments. The Speed Handle is a 3D-printable attachment that makes large adjustments fast and easy, while keeping tiny adjustments as fine as the standard knob.

Cleanroom Wipes for Flux: Use clean room wipes to clean up flux residue. Cleaning flux after assembly or repair requires applying a cleaner (like IPA or deionized water) and removing the flux. Get your board wet with IPA, use an ESD-safe brush to break up the residue, put the wipe down, and brush again. The cloth soaks up the residue and is far more effective than just rinsing.

https://youtube.com/@AddOhms

Max Imagination

Practical Workshop Solutions
Smart Component Selection

Choosing the Right ESP32 Board: There are five key ESP32 families: ESP32-WROOM for general purpose with great GPIO and dual-core performance, ESP32-CAM for budget-friendly camera projects, ESP32-S2/S3 for USB HID applications and fast I/O, ESP32-C3 for low power with RISC-V and BLE 5.0, and XIAO ESP32-S3 for tiny form factor with big features.

Motor Selection Strategy: Define your project's needs first: torque requirements (force needed to move or rotate loads), speed/RPM requirements, and power supply constraints. Choose motors with at least the minimum torque needed to ensure smooth operation and avoid stalling. Different motor types serve different purposes: brushed DC motors for simple, cost-effective projects; brushless DC (BLDC) motors for high efficiency and no brush wear; geared DC motors for slow speed and high torque; stepper motors for precision control; and servo motors for torque and precision in small packages.

Battery Technology Selection: Match battery voltage to your project requirements. Common voltages include 3.7 V Li-ion or LiPo cells for small, portable, low-power projects; 7.4 V or 14.1 V Li-ion or LiPo packs for motorized builds requiring more power; and 12 V or higher packs for high-power systems. Consider the C rating — it tells you how fast a battery can discharge. Use the formula mAH = (current draw in mA) × (desired run time in hours) to estimate battery capacity needs.

Arduino vs ESP32: For basic or low power projects, choose Arduino (8-bit microcontrollers). For wireless connectivity or speed requirements, choose ESP32 (32-bit microcontrollers). Arduino boards excel at simple, efficient operation for sensors, motors, and basic robotics with low power consumption and beginner-friendly operation. ESP32 boards provide high performance for complex tasks, wireless connectivity with built-in Wi-Fi and Bluetooth, and more memory and speed for data-heavy projects.

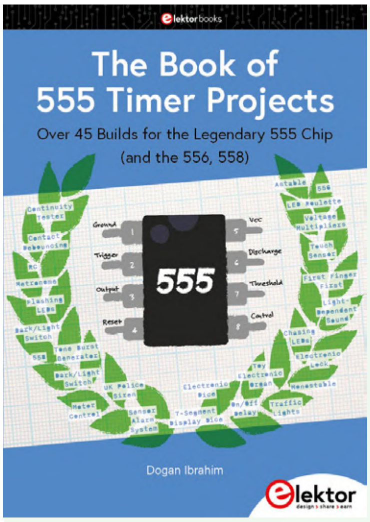
Component Organization Systems: Stop wasting time digging through parts when you could be building. Good organization saves time, reduces frustration, and speeds up prototyping. Choose your organizer style: hardware storage bins for bulk components, repurposed containers for small parts, tackle boxes for resistors and capacitors, and labeled drawers for frequently used items. Sort components by type, value, and project frequency. Use labels, colored tape, or spreadsheets to track inventory.

https://youtube.com/@MaxImagination

250386-01

14 Practical Projects with the 555 Timer

DC Motor Control and Fast Reaction Challenges



By Dogan Ibrahim (United Kingdom)

We begin with the DC Motor H-Bridge Control project, which uses two 555 timers to create an H-bridge circuit that controls the direction of a DC motor, offering a great introduction to motor control. Next, we switch to the Fastest Finger Quiz Circuit project, which tests your reaction time and that of others. With two or more players, the first person to press the button lights up the LED and triggers the buzzer, offering a creative way to see the 555 timer in action while engaging in a quiz-style challenge.

Editor's Note. *This article is an excerpt from* The Book of 555 Timer Projects*. It was formatted and lightly edited to match Elektor Magazine's conventions and page layout. The author and editor are happy to help with queries. Contact details are in the* **Questions or Comments?** *box.*

The **DC Motor H-Bridge Control** and **Fastest Finger Quiz Circuit** are a fun way to get into electronics with the 555 timer. You don't need a lot of complicated parts, and both projects are easy to set up. Whether you're racing to answer questions or learning how to control a motor, these projects give you a hands-on introduction to the basics of electronics.

DC Motor H-Bridge Control

In this project, you'll use an H-bridge circuit to control a DC motor in either direction. The H-bridge is widely used in motor speed control applications, such as in robotics. A potentiometer will be used to control the direction of the motor and a small, brushed DC motor will be used to demonstrate the setup.

The basic operation of the H-bridge is simple: when switches SW1 and SW4 are closed, the motor will rotate in one direction. When switches SW2 and SW3 are closed, the motor will rotate in the opposite direction (see **Figure 1**).

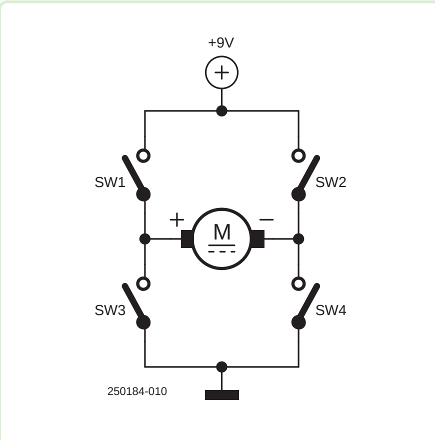


Figure 1: H-bridge basic circuit operation.

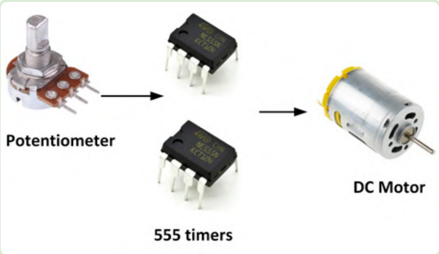


Figure 2: Block diagram of the DC Motor H-Bridge Control project.

This project uses two 555 timer ICs. **Figure 2** shows the block diagram of the project, and **Figure 3** shows the circuit diagram. When the potentiometer arm is moved in one direction, the voltage at the trigger input of the first 555 goes lower than VCC/3 and this sets its output high. As the output of the second 555 is low, the motor starts rotating in one direction. When the potentiometer arm is moved in the other direction, the voltage at the threshold of the first 555 exceeds 2/3 VCC. This resets the output of the first 555 timer and at the same time the second 555 becomes high. This puts the first 555 into current-sinking mode and the second one into current-sourcing mode.

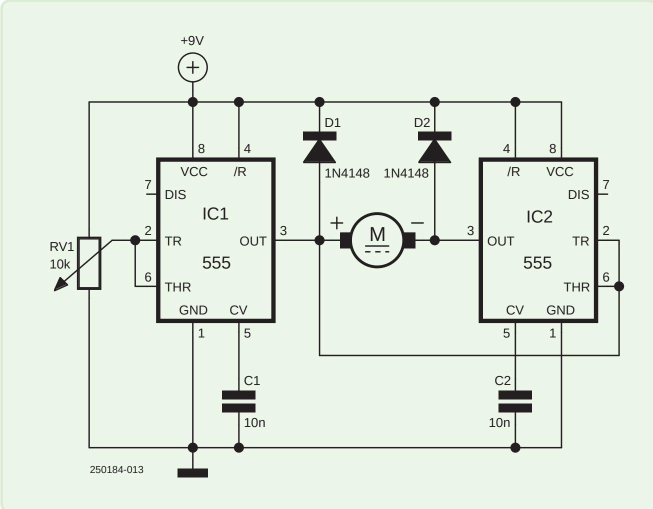


Figure 3: Circuit diagram for the DC Motor H-Bridge Control project.

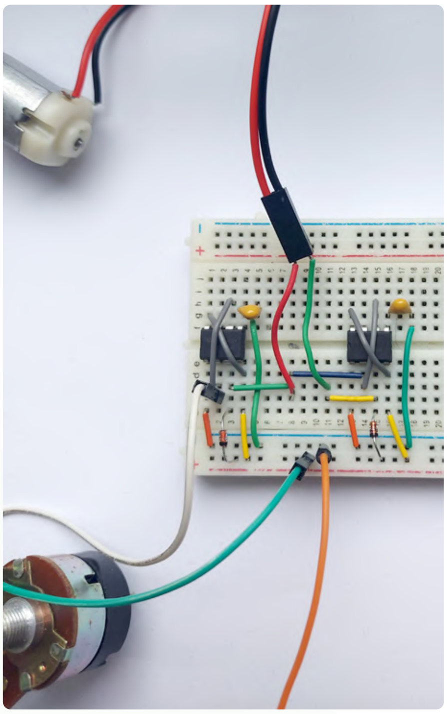


Figure 4: DC Motor H-Bridge circuit on a breadboard.

Testing

To test the project, connect the battery and rotate the potentiometer. **Figure 4** shows the circuit constructed on a small breadboard. The motor should rotate in one direction when the potentiometer is turned one way. When the potentiometer is rotated in the opposite direction, the motor should rotate the other way. This demonstrates the H-bridge circuit's ability to reverse the direction of the motor with the simple use of two 555 timers.

Fastest Finger Quiz Circuit

This quiz is an event played by two or more participants to test their knowledge. A pushbutton switch and a light are provided for each user. After the quizmaster asks a question, the person who presses the button first is given the opportunity to answer the question. When a person presses the button, an active buzzer sounds and the person's light turns on to indicate who has pressed the button first. Once someone presses the button, all other participants' buttons are disabled. The quizmaster then resets the circuit, turning off the lights and buzzer, preparing for the next question.

Version with Two Participants

In this version, it is assumed that the quiz is between two people. The project uses two pushbutton switches, two LEDs, and a buzzer. Additionally, the quizmaster uses a pushbutton switch to reset the circuit. In a real application, relays can be used, and the LEDs can be replaced with lamps and the buzzer with a loud audio device. **Figure 5** shows the block diagram of the project.

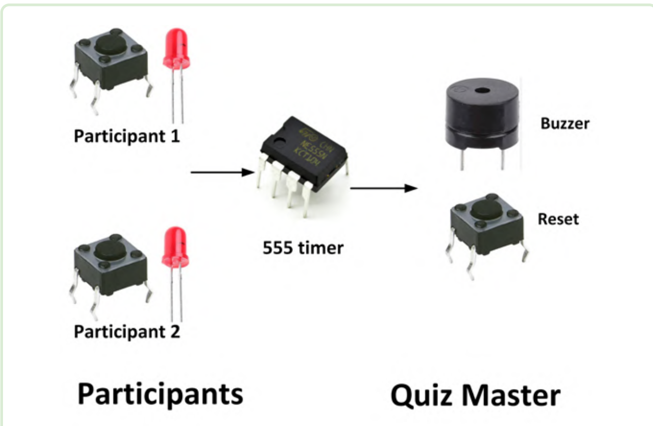


Figure 5: Block diagram for the two-player project.

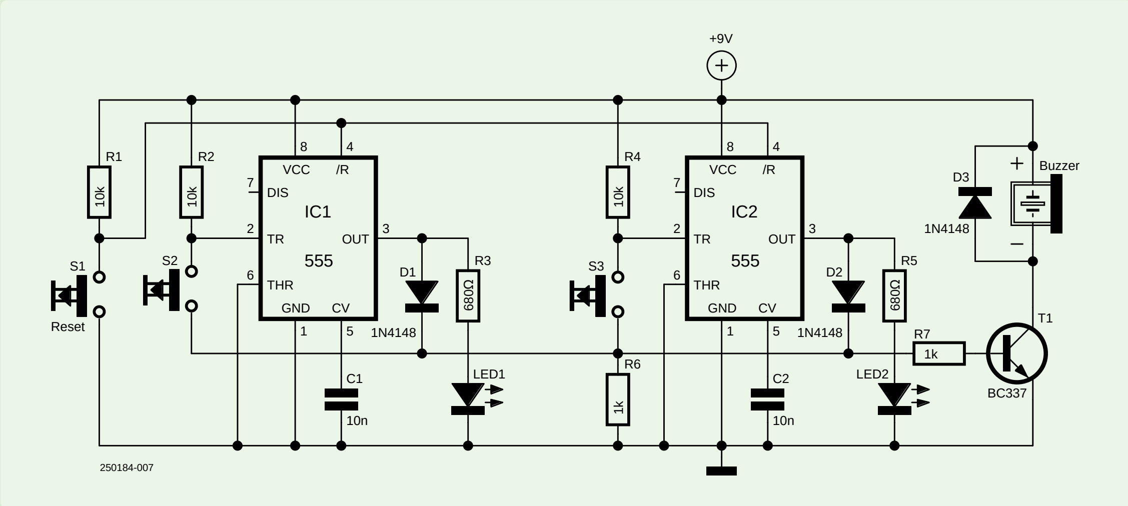


Figure 6: Circuit diagram for the two-player project.

As shown in **Figure 6** in the circuit diagram, two 555 timer ICs are used in the design. More timer chips can be used if there are more than two participants. The buttons are connected to the trigger inputs of the two timers. Pressing a button triggers a timer, turns on that participant's LED and activates the active buzzer. At the same time, diode D1 or D2 becomes forward biased and keeps the outputs high so that the circuit does not accept any more triggers. In other words, the output of that timer disables the triggering of the other timer so that only one button press is accepted by the circuit. The quizmaster presses the Reset button to restart the game.

Testing

Figure 7 shows the circuit on a breadboard. Press the buttons randomly as well as at the same time to make sure that only the first button press is accepted by the circuit.

The buzzer should sound when a button is pressed and its corresponding LED should turn on. Pressing the Reset button should reset the circuit by turning the buzzer and the LED off.

Version with Six Participants in Three Groups

This version is similar to the previous one, with the key difference being the assumption that there are three groups of people in the quiz, each consisting of two participants. The design is very similar to the two-player version, but here, three 555 timers and more buttons are used to accommodate the additional participants.

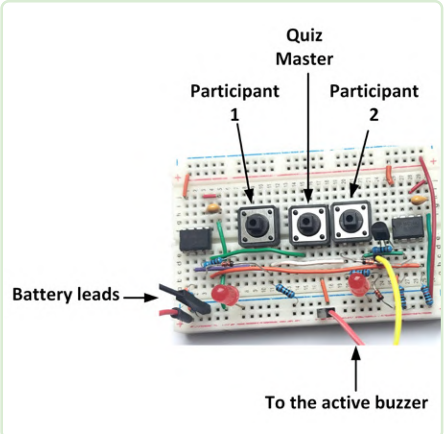


Figure 7: Breadboard setup for the two-player project.

Component List for the DC Motor Project

1× 10 kΩ potentiometer
2× 0.01 µF capacitor
2× 1N4148 diodes
2× 555 timer IC
1× 9 V battery
1× battery Clip
1× breadboard

Component List for 2-Player Circuit

3× 10 kΩ resistor
2× 1 kΩ resistor
2× 680 Ω resistor
2× 0.01 µF capacitor
2× 555 timer chip
1× BC337 transistor
2× 5 mm red LED
3× 1N4148 diode
3× pushbutton switch
1× active buzzer
1× 9 V battery
1× battery clip
1× breadboard

Component List for 3-Group Circuit

3× 680 Ω resistor
4× 10 kΩ resistor
2× 1 kΩ resistor
3× 0.01 µF capacitor
4× 1N4148 diode
3× 5 mm red LED
3× 555 timer chip
1× BC337 transistor
7× pushbutton switch
1× active buzzer
1× 9 V battery
1× battery clip
1× breadboard

The number of groups and participants can be easily adjusted, offering flexibility for different configurations. The block diagram for the project is shown in **Figure 8**, illustrating the system's layout.

Figure 9 shows the project circuit diagram. As in the previous project, the quizmaster resets the circuit by pressing the Reset button.

250184-01

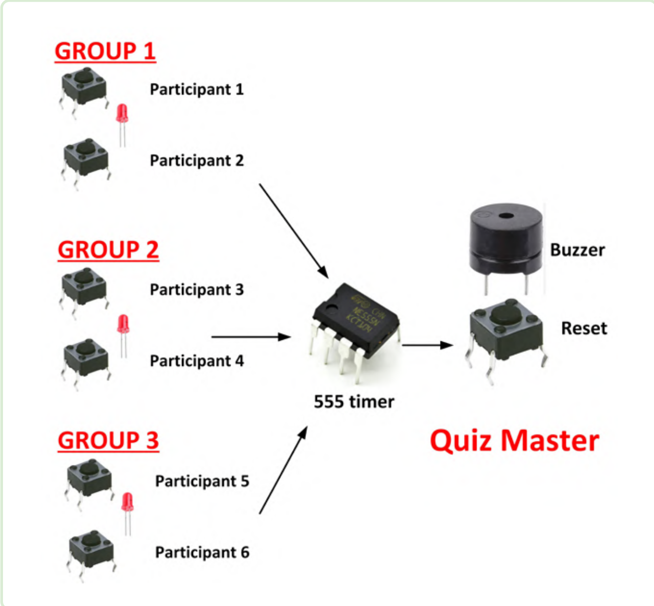


Figure 8: Block diagram for the three-group project.

Questions or Comments?

Do you have any questions or comments related to this article? Email the author at d.ibrahim@btinternet.com or Elektor at editor@elektor.com.

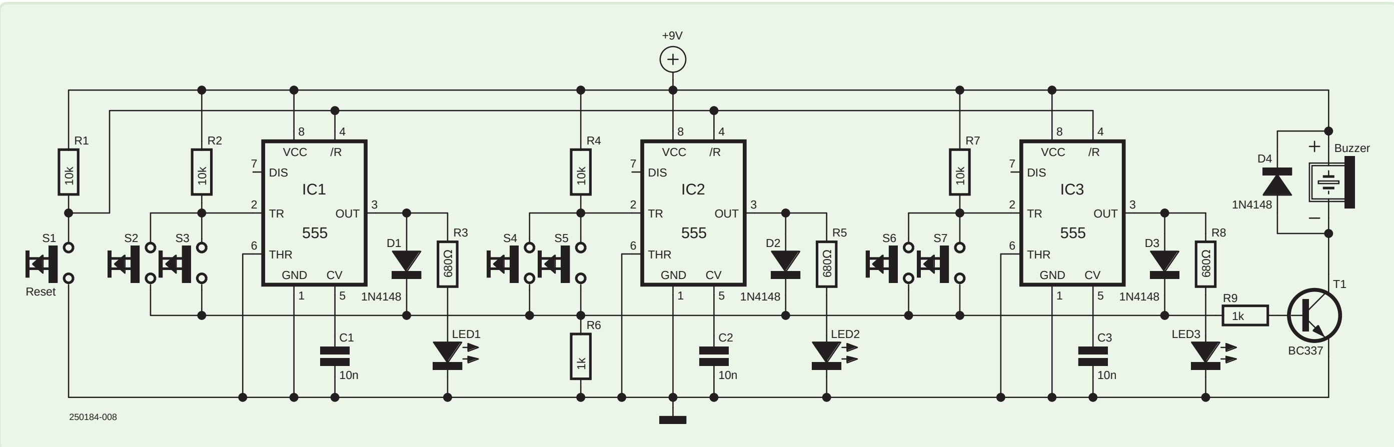


Figure 9: Circuit diagram for the three-group project.

About the Author

Prof Dr Dogan Ibrahim has BSc degree in electronic engineering, an MSc degree in automatic control engineering, and a PhD degree in digital signal processing. Dogan has worked in many industrial organizations before he returned to academic life. Prof Ibrahim is the author of over 70 technical books and published over 200 technical articles on microcontrollers, microprocessors, and related fields. He is a Chartered electrical engineer and a Fellow of the Institution of the Engineering Technology. He is also a certified Arduino professional.

Related Products

- Dogan Ibrahim, *The Book of 555 Timer Projects* (Elektor 2024)
www.elektor.com/20948
- Dogan Ibrahim, *The Book of 555 Timer Projects* (E-Book, Elektor 2024)
www.elektor.com/20949
- 555 Timer Projects (Bundle)
www.elektor.com/21117

15 Basic AC-Load-On Monitor

Save Energy with a Simple Device

By Giuseppe La Rosa (Italy)

This tiny AC-Load-On LED monitor is useful wherever there are remote small electric devices such as on terraces, in garages, in hallways, and in courtyards. It warns us when loads (up to 150 W) were left on, with a considerable saving in electricity. It doesn't require a connection to the neutral line and can be installed anywhere in the electrical system.

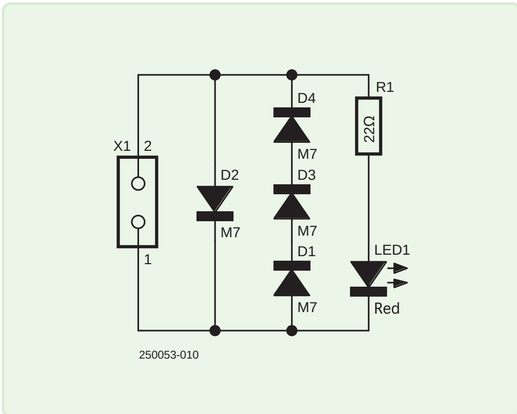


Figure 1: AC Load Monitor circuit diagram.

In almost every home, there are lamps or external electrical loads that — once they are far away — we often forget to turn off because they're out of sight. The proposed LED-based monitor can be inserted directly into the wall box that houses the switch for the load to be sensed.

Circuit

In **Figure 1**, we can see the basic circuit diagram. The operation is as follows: When the load switch is closed, the line voltage on terminal block X1 alternatively brings D2 (positive half-wave) and the series of D1, D3, and D4 (negative half-wave) to conduction, causing a negligible asymmetrical voltage drop on the AC line of about 0.7 V and 2.1 V, respectively. The latter, limited in current by resistor R1, turns on LED1.

WARNING! This circuit is connected directly to 230 V AC mains grid. Always make sure to disconnect the mains power supply before you connect the load monitor to electrical devices!

Assembly and Wiring

The PCB is double-sided with metallized holes, and the construction of the circuit, based on SMD components, is simple. **Figure 2** shows the silkscreen, indicating the components' soldering positions and the diodes' orientation, while the PCB layout for both sides can be downloaded from [1].

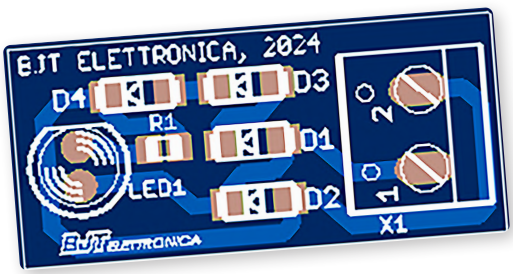


Figure 2: The tiny PCB's component-side silkscreen.

Figure 3: The prototype, installed in a blank cover plate for a wall box.

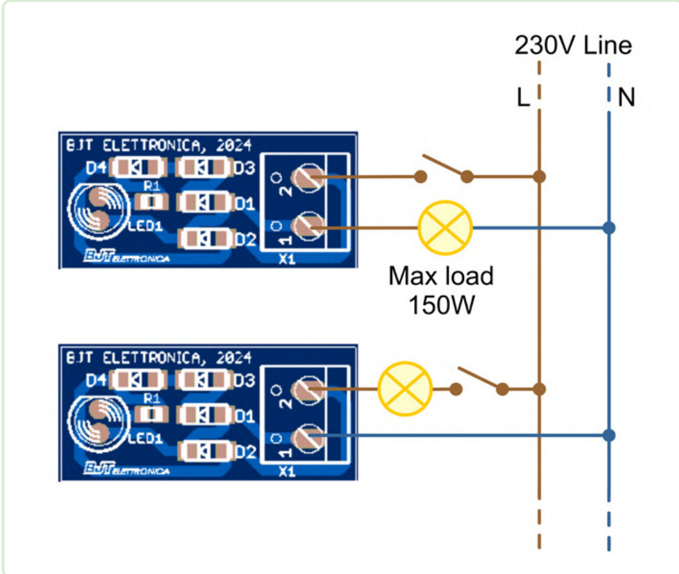


Figure 4: The simple wiring diagram of the AC Load Monitor to the existing electrical system.

Assembling SMD components requires specific tools and a certain amount of manual dexterity. The soldering iron should have a very fine tip and a power rating of not more than 12 W (or even better, use an SMD soldering station). The tin wire should have a diameter of not more than 0.5 mm and the picking up and placing of components should be done with the help of needle-nose tweezers and a magnifying glass. Note that LED1 should be placed on the component side, and terminal block X1 on the solder side of the small PCB.

Once the assembly is complete, installation can proceed. **Figure 3** shows the finished prototype, already in place. All you need to do is to drill a 5 mm hole in the wall box blank cover plate, insert the LED with the circuit board and connect it in series to the existing switch, as shown in **Figure 4**.

The series connection of this small circuit can be realized either on the live (L) or the neutral (N) lines, indifferently. The high-efficiency red LED guarantees good visibility, regardless of ambient light intensity.

Before Powering Up

This circuit may also be assembled onto a stripboard, replacing the SMD components with through-hole ones and following the schematic in Figure 1. The maximum load that can be connected to the circuit board is 150 W. A video of the project in action is available at [2].

250053-01

Questions or Comments?

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

About the Author

Passionate about electricity from an early age, Giuseppe La Rosa graduated with a degree in Electronics and Telecommunications in 2002 at I.T.I.S. "G. Ferraris" of Acireale, Sicily. Later, he began studying microcontroller systems, particularly PIC microcontrollers and the Arduino UNO open-source platform. Over the years, he has created various prototypes, many of which have been published in electronics magazines. Currently, he deals with security systems (video surveillance and anti-burglary alarms) and software for the management of points of sale.

WEB LINKS

[1] Elektor Labs downloads for this project: https://tinyurl.com/elektorlabsdl
[2] Video of the prototype in action: https://youtu.be/XqeqJa8_HLc

16 Power Banks in Parallel

A Three-Day Continuous Power Solution

By Johnny Verhoeven (Belgium)

USB power banks are increasingly being used to power small microcontroller projects and devices, but what to do if the capacity is not sufficient for a stand time of some days? Connecting multiple power banks in parallel to share the load could be a solution, but as often is the case in electronics, it is not so easy as it seems. Here's a detailed exploration of one approach to solve this power dilemma.

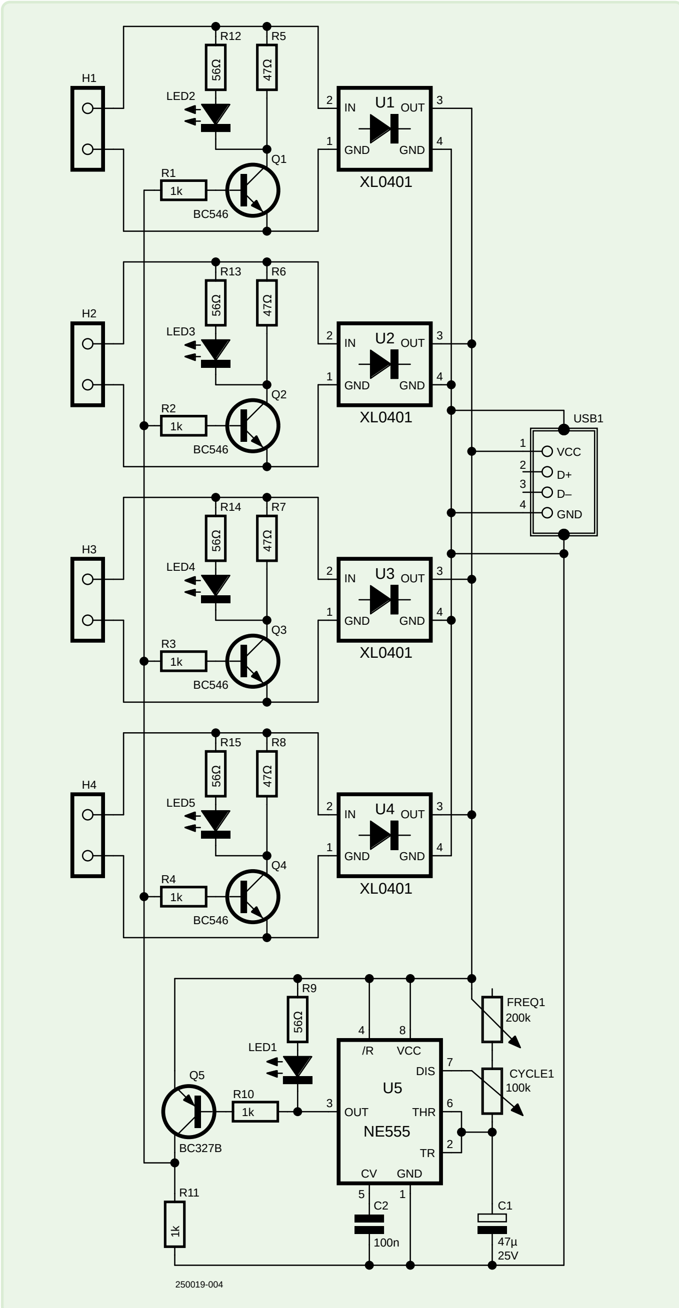


Figure 1: The schematic diagram.

Creating a haunted box — a device that produces eerie sounds and lights — can be a fun project, especially for Halloween or other spooky events. However, designing one that runs unattended for three days presents unique challenges, particularly the need for consistent power delivery. There was no single power pack in hand with enough capacity to meet this requirement. The logical solution was to connect multiple power packs in parallel to share the load and extend the runtime. Doing so introduced several technical challenges, which are described below along with the solutions employed developing this project.

Voltage Mismatch and Current Backflow

No two battery packs, even when brand new, are identical in voltage output. When connected in parallel, this voltage mismatch can cause one pack to backfeed current into another. This not only wastes energy but can damage the packs or any step-up circuits involved.

To prevent current backflow, diodes were introduced between each power pack and the circuit. A common choice like the 1N4001 diode blocks reverse current effectively but comes with a significant drawback: it induces a 0.7-V voltage drop, which can drastically reduce the output voltage, particularly for low-voltage systems.

To overcome this, a low-drop diode module, such as the XL0401, was employed. These modules have minimal voltage drop and efficiently prevent backflow without compromising the circuit's performance.

Unequal Load Sharing

With the diodes in place, another issue surfaced: uneven power delivery. The power pack with the highest output voltage became the sole provider of power, while the others switched off due to their lower voltages and no load. When the active pack was depleted, there was a brief delay before another pack took over. That loss of power caused the haunted box's Arduino microcontroller to reboot, losing the settings for things like timing and volume.

One possible solution considered was the "keep-alive circuit" [1] from *Elektor Circuit Special 2024*. This is a fairly complex solution, so a simplified circuit was designed which provided the same "keep alive" pulse to all the powerpacks, with only a slight loss of efficiency.

The Circuit

When you think "pulse" in circuitry, the first thing that comes to mind is a 555 timer IC. So, the basis of this circuit (**Figure 1**) is a standard 555 timer. R9 and LED1 indicate the pulses and can be omitted if needed. Length and frequency of the pulses can be adjusted using trim potentiometers in order to set to minimum power consumption (more about this later).

We are looking now on the top part of the circuit diagram, responsible for the first one of the powerpacks, but the next three parts of the circuit act exactly as the first one. The circuit to accommodate four power packs can be easily extended if more power packs need to be connected.

The transistor Q1 and resistor R5 form the actual load, about 100 mA, but that can be changed as needed. R12 and LED2 indicate the pulses for the "live" powerpack. When a powerpack is depleted and shuts down, the pulses stop because the LEDs get their power directly from the powerpack themselves.

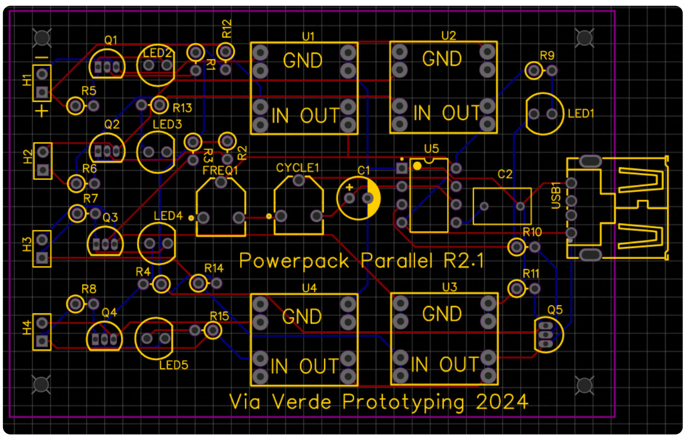


Figure 2: PCB layout of the Parallel Power Bank circuit.

Next is the low-drop diode module XL0401. This acts as a diode but without the 0.7-V voltage drop. Using an XL0401 makes sure the full 5 V from the powerpack is available at the output.

At H1 to H4, simple USB-A-male/Wire-adapters are soldered on the PCB.

Setting the Pulses

We want the extra load as low as practical. Pulses must be short and low frequency. This is done by setting the pulse length and frequency so that all connected powerpacks stay alive and don't shut down.

Do this by first setting the potentiometer FREQ1 all the way to the right. That is the highest frequency. Then set CYCLE1 all the way to the left. That is the longest pulse. This should keep all powerpacks in the on-state. If that is not the case, you can reduce the value of R5, R6, R7 and R8.

To minimize power usage in the keep-alive state, lower the frequency (turn FREQ1 to the left) until one of the packs shuts down. Now increase FREQ1 a bit, until all packs stay on. Then shorten the pulse (turn CYCLE1 to the right) until one of the packs shuts down. Now decrease CYCLE1 a bit, until all packs stay on. Verify that all packs stay on. Increase frequency and/or lengthen the pulse as needed.

The PCB

Upon achieving satisfactory results from the breadboard tests, I proceeded to design a PCB for the circuit. The PCB and its schematics were meticulously crafted using EasyEDA, and then sent to a PCB manufacturer. For a visual representation of the PCB, refer to **Figure 2**.

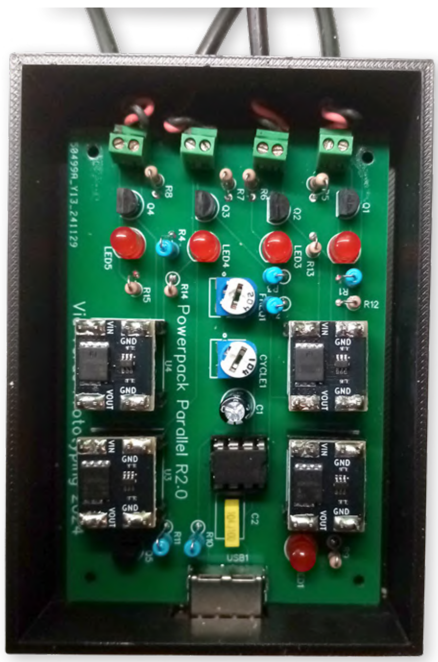


Figure 3: The PCB in its 3D printed enclosure.

Figure 4: How it is connected.

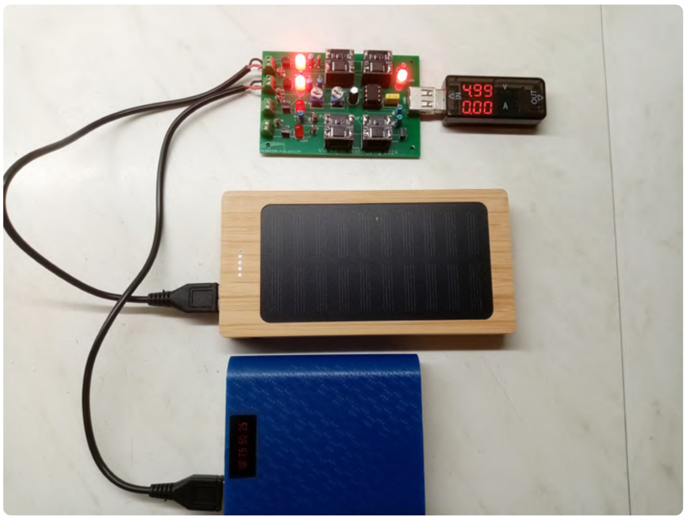


Figure 5: Testing with two powerpacks.

Finally, all the components are soldered in place and then encased in a 3D-printed enclosure as shown in **Figure 3**. How it is all connected is shown in **Figure 4**. **Figure 5** shows a test with two powerpacks.

250019-01

Questions or Comments?

Do you have technical questions or comments to submit about this article? Feel free to write to the author at johnny.verhoeven@yahoo.com or to the Elektor Editorial Team at editor@elektor.com.

Related Product

- **Joy-IT UM25C USB Measuring Device**
  www.elektor.com/18599

About the Author

Johnny Verhoeven studied electronics and holds a bachelor's degree in IT. He also holds several degrees in the art of music. Johnny is an overall creative maker who, since his recent retirement, has taken up electronics again. You can find more about this on his Facebook page, "Via Verde Prototypes," at *facebook.com/profile.php?id=100083476892160*.

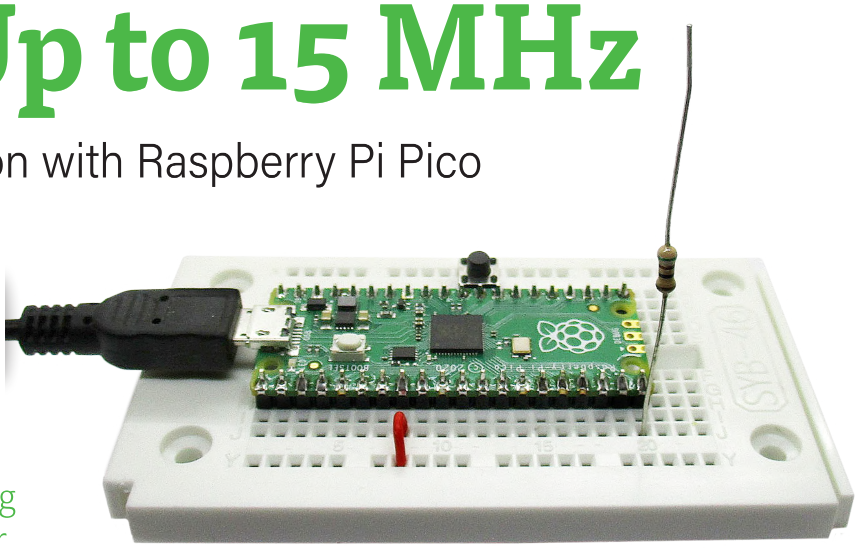
WEB LINK

[1] Johnny Verhoeven, "Powering Low-Draw Devices With Power Banks," Elektor Circuit Special 2024: https://www.elektormagazine.com/230637-01

17 VFO Up to 15 MHz

An Implementation with Raspberry Pi Pico

By Burkhard Kainka (Germany)



This tunable oscillator serves as an auxiliary oscillator for shortwave reception. It can be used to listen to signals without their own carrier using a normal AM shortwave radio receiver. It can then be used to receive SSB, CW, or digital signals such as FT8 if the frequency is tuned precisely to the carrier. A Raspberry Pi Pico, whose programmable input/output (PIO) is programmed using assembly language, is sufficient to generate the RF signal.

A short antenna in the form of a resistor at the output of GPIO15 transmits the signal. The RF signal is generated by a PIO assembler program *vfo.pio* [1] simply by switching the port pin on and off with a counting loop for delay. The program runs at the maximum clock rate and only needs one clock cycle per command. Additional counting loops are the only way to make things slower. This delay can be controlled from the outside.

The assembler program uses each given cycle period twice, because the actual bottleneck is the transfer loop in the main program. Thus, the maximum possible frequency is doubled. One delay loop works with `x`, the other with `y`. The number of cycles is shown after the comment sign `;`.

```
.program vfo
    set pindirs,  1
.wrap_target
    pull    ;1
    out x, 32    ;1
    mov y, x     ;1
loop2:
    jmp y-- loop2    ;1+x
    set pins, 1 [4]  ;1+4
    set pins, 0      ;1
    nop      ;1
    nop      ;1
    nop      ;1
loop3:
    jmp x-- loop3    ;1+x
    set pins, 1 [4]  ;1+4
    set pins, 0      ;1
.wrap           ; 10+x
```

Main Program

The processor clock speed was increased to 180 MHz in the C++ program in **Listing 1**. Because a run takes `10+x` clock cycles, x = 8 would generate exactly 10 MHz. With x = 9, you would get a noninteger 9.473684 MHz. To set any frequency in between, you have to switch between 8 and 9 extremely quickly and accurately. It's hard to believe that you can do this and that a clean signal will come out at the end.

The desired frequency in Hertz can be sent with a prefixed "f" via the serial monitor of the Arduino IDE at 9600 baud (**Figure 1**). For even easier operation, a VB6 program originally developed for the Elektor SDR-Shield [2] was adapted. The software also allows fine-tuning for SSB reception or other applications (**Figure 2**). With a simple trick, the tuning range was extended up to 30 MHz. All frequencies above 15 MHz are divided by three, so that in the upper range, you work with the third harmonic.

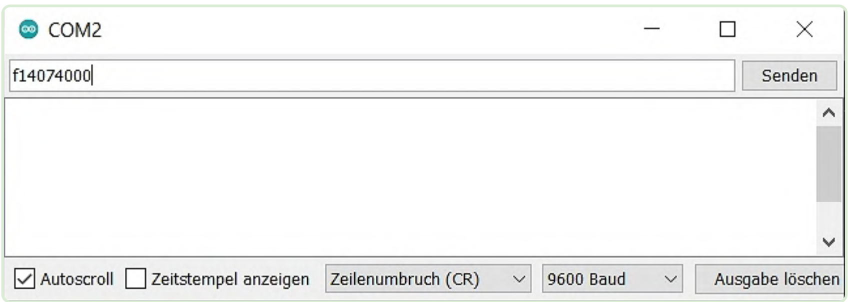


Figure 1: The frequency in Hz is entered with a leading "f" via the serial monitor of the Arduino IDE at 9600 baud.

Listing 1: C++ Program.

```
//PicoVFO bis 15 MHz
#include "vfo2.pio.h"
#include "pico/stdlib.h"
#include "pico/multicore.h"

uint32_t clk_khz = 180000;
volatile uint32_t f=7074000;
volatile uint32_t periods;

void setup(void){
  set_sys_clock_khz(clk_khz,true);
  Serial.begin(9600);
}

void set_freq(void){
  uint64_t ratio = (uint64_t)clk_khz*1000LL *(1<<24)/(uint64_t)f;
  periods=(uint32_t)(ratio/1-(10<<24));
}

void flush_input(void)
{
  while (Serial.available() > 0)
  Serial.read();
}

void loop(){
  if (Serial.available()) {
    int ch = Serial.read();
    f = Serial.parseInt();
    Serial.read();
    if((ch == 102)&& (f>100000)){
      {set_freq();}
    }
  }
}

void setup1(void){
  uint16_t t;
  uint32_t delta;

  PIO pio = pio0;
  gpio_init(0);
  gpio_set_dir(0, GPIO_OUT);
  pio_gpio_init(pio0, 15);
  uint offset = pio_add_program(pio0, &vfo_program);
  pio_sm_set_consecutive_pindirs(pio0, 0, 15, 1, true);
  pio_sm_config c = vfo_program_get_default_config(offset);
  sm_config_set_set_pins(&c, 15, 1);
  pio_sm_init(pio0, 0, offset, &c);
  pio_sm_set_enabled(pio0, 0, true);
  set_freq();
  //periods=10<<24;
  for(;;){
    t = (periods+delta) >> 24;
    pio_sm_put_blocking(pio0, 0, t);
    delta += periods-(t << 24);
  }
}

void loop1(){}
```

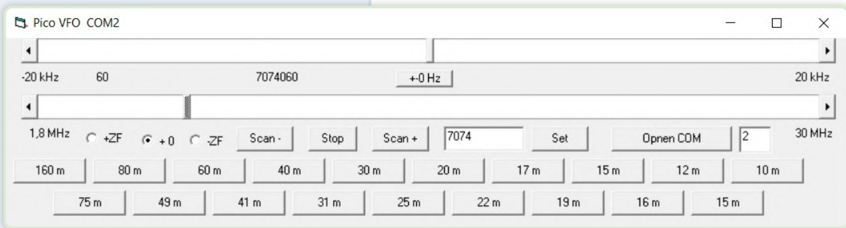


Figure 2: The VB software allows fine-tuning for SSB reception or other applications.

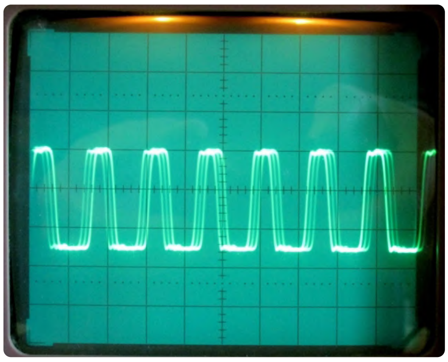


Figure 3: The oscillogram shows a distinct phase jitter with weak secondary emissions.

Operation

A signal of 14,074 kHz sounds clean in the receiver. The oscillogram shows an obvious phase jitter, which actually leads to weak secondary emissions at greater distances (**Figure 3**).

WSPR places particularly high demands on the stability of the oscillator signal. However, even these signals could be reliably received.

Translated by Jörg Starkmuth — 250214-01

Questions or Comments?

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

Related Product

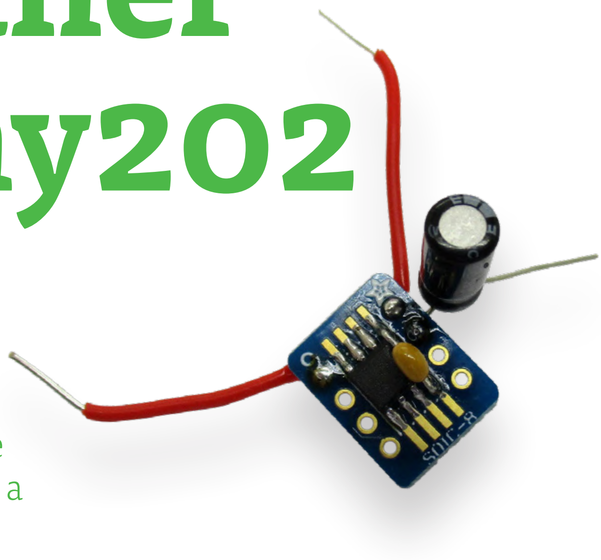
- **Burkhard Kainka, *SDR Hands-on Book* (Elektor 2019)**
  Book: www.elektor.com/18914
  Ebook/PDF: www.elektor.com/18915

WEB LINKS

[1] Software download:
https://www.elektormagazine.com/labs/elektor-articles-software-downloads

[2] Burkhard Kainka, "Elektor SDR Shield 2.0 (3)," Elektor 11-12/2018:
https://www.elektormagazine.com/magazine/elektor-52/42080

18 Violin Tuner with ATtiny202



By Burkhard Kainka (Germany)

One port pin is sufficient to output a square wave signal. The frequencies of the notes G, D, A, and E are generated here via a phase accumulator.

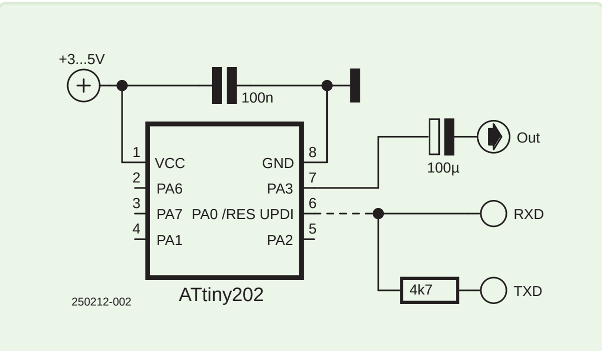


Figure 1: ATtiny202 circuit with UPDI programming interface and external components.

The ATtiny202 is small, inexpensive, and powerful. No special programmer is required for programming via the UPDI pin; all you need is a USB/serial converter and a resistor (**Figure 1**, see also the DDS generator in this issue [1]). Getting started is easy with the Arduino IDE and the megaTinyCore board extension.

Oscillator

A numerically controlled oscillator (NCO) can be constructed similarly to a DDS generator [1]. However, the Tiny202 does not have a D/A converter. Therefore, in the simplest case, a square wave is generated, which means that we do not need a table containing the waveform. The signal shape is relatively unimportant, because a violin does not generate a sine tone either, but has numerous overtones. The controller's internal RC oscillator runs at 20 MHz. The frequency is maintained with an accuracy of 1%, which is perfectly adequate for this purpose.

The program (**Listing 1**) [2] uses an interrupt function that is called by the TCA0 timer at approximately 65.5 kHz. Here, the frequency in Hertz is added to the phase accumulation register. Its most significant bit is output directly to PortA.3, where the square wave signal with the desired frequency is generated. You can connect headphones or a speaker with an electrolytic capacitor and, if necessary, a suitable series resistor, or even a power amplifier. The program generates the notes G, D, A, and E in succession for 5 s each.

Translated by Jörg Starkmuth — 250212-01

Listing 1: C Program.

```
//NCO202, Violin Tuner
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#define F_CPU 20000000
uint16_t phase= 0;
uint16_t freq = 1000; // 1 kHz

ISR(TCA0_OVF_vect) {
  PORTA.OUT  = 8*(phase>>15);
  phase += freq;
  TCA0.SINGLE.INTFLAGS = TCA_SINGLE_OVF_bm;
}
```

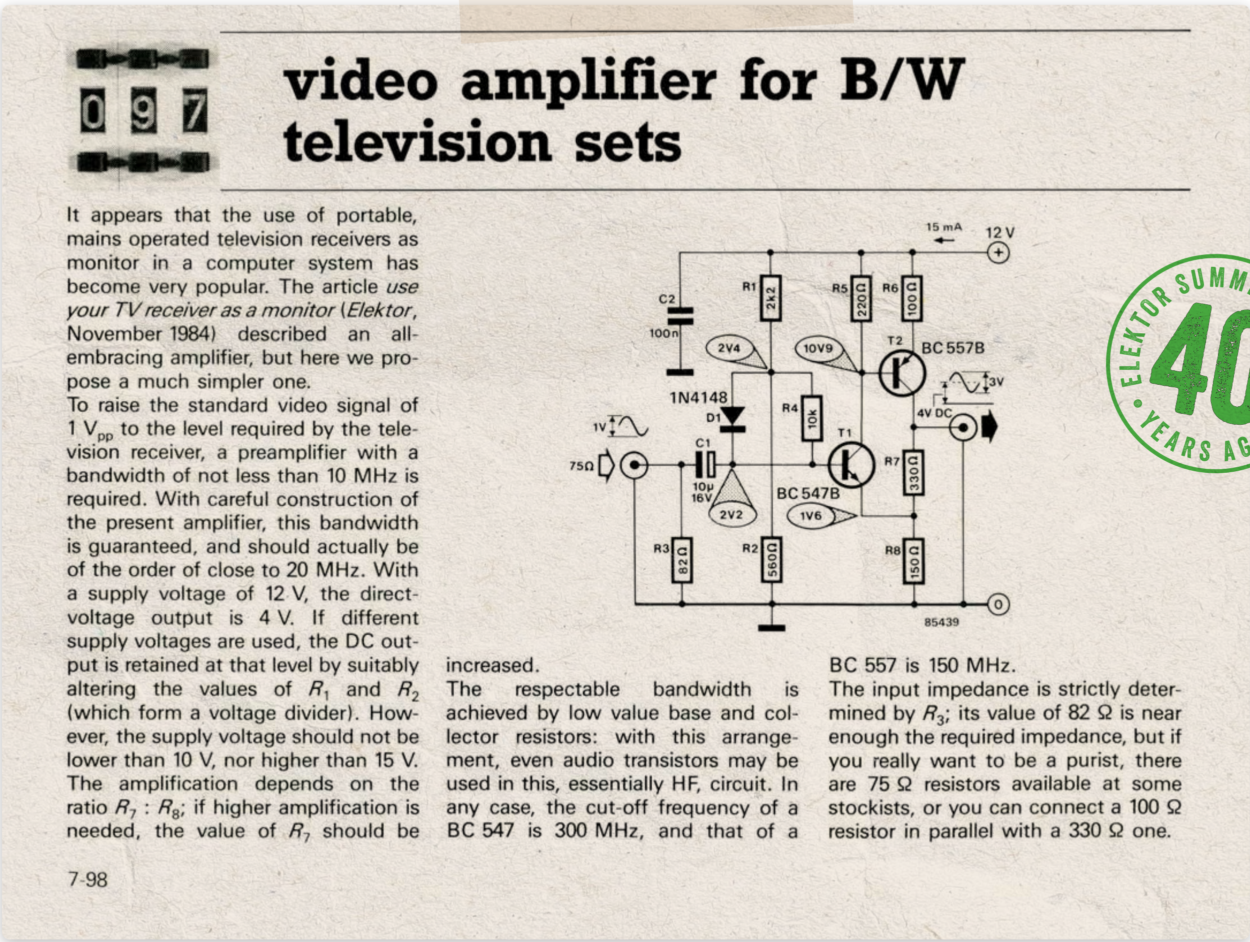
```
int main(void) {
  sei();
  _PROTECTED_WRITE (CLKCTRL_MCLKCTRLB, 0); // 20 MHz
  PORTA.DIR |= PIN3_bm;
  TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV1_gc;
  TCA0.SINGLE.PER = 305; // 65536 Hz
  TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;
  TCA0.SINGLE.CTRLA |= TCA_SINGLE_ENABLE_bm;
  while (1) {
    freq=196;   //G
    _delay_ms(5000);
    freq=294;   //D
    _delay_ms(5000);
    freq=440;  //A
    _delay_ms(5000);
    freq=659;  //E
    _delay_ms(5000);
  }
}
```


Questions or Comments?

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

WEB LINKS

[1] Burkhard Kainka, "DDS Generator with ATtiny," Elektor Circuit Special 2025: https://www.elektormagazine.com/250211-01
[2] Software download: https://www.elektormagazine.com/labs/elektor-articles-software-downloads



097

video amplifier for B/W television sets

It appears that the use of portable, mains operated television receivers as monitor in a computer system has become very popular. The article *use your TV receiver as a monitor* (*Elektor*, November 1984) described an all-embracing amplifier, but here we propose a much simpler one.

To raise the standard video signal of 1 V_{pp} to the level required by the television receiver, a preamplifier with a bandwidth of not less than 10 MHz is required. With careful construction of the present amplifier, this bandwidth is guaranteed, and should actually be of the order of close to 20 MHz. With a supply voltage of 12 V, the direct-voltage output is 4 V. If different supply voltages are used, the DC output is retained at that level by suitably altering the values of R_1 and R_2 (which form a voltage divider). However, the supply voltage should not be lower than 10 V, nor higher than 15 V. The amplification depends on the ratio R_7 : R_8; if higher amplification is needed, the value of R_7 should be increased.

The respectable bandwidth is achieved by low value base and collector resistors: with this arrangement, even audio transistors may be used in this, essentially HF, circuit. In any case, the cut-off frequency of a BC 547 is 300 MHz, and that of a BC 557 is 150 MHz.

The input impedance is strictly determined by R_3; its value of 82 Ω is near enough the required impedance, but if you really want to be a purist, there are 75 Ω resistors available at some stockists, or you can connect a 100 Ω resistor in parallel with a 330 Ω one.

7-98

19 Capacitance Meter

20 pF to 600 nF

By Burkhard Kainka (Germany)

Need a capacitance meter? This very simple capacitance meter is not particularly accurate, but does not require any additional components because it uses the controller's internal pull-up resistor.

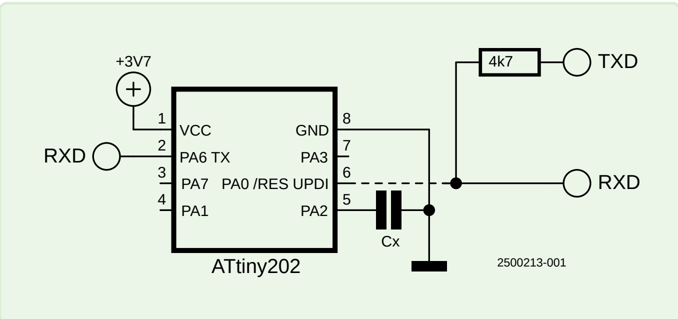


Figure 1: Circuit diagram of the capacitance meter.

The object to be measured is simply placed between pin 5 (PA2) and GND, as shown in **Figure 1**. For accuracy, a stabilized operating voltage of 3.3 V should be used.

Listing 1 shows the program, which can be downloaded from Elektor Labs [1]. The Arduino IDE and the megaTinyCore board extension from Spence Konde were used (for programming see also [2]). In the setup function, the pin is set to output and the internal pull-up resistor is activated. In the loop function, we configure the pin to input, and the line `while(!(VPORTA.IN & 4)){n++;}` then measures the charging time of the capacitor, which depends on the value of the internal pull-up resistor and the capacitance.

The factor 20 for the conversion to pF and the quotient 50 for the conversion to nF have been determined experimentally and can be adjusted for individual calibration purposes. Reference capacitors or a precise capacitance measuring device can be used for this purpose.

Listing 1: C Program.

```
#include <Arduino.h>
int n;

void setup () {
     PORTA.DIRSET = 0x04;    //Output PA2
     PORTA.PIN2CTRL = 0x08;  //Pullup PA2
     Serial.begin(9600);
 }

void loop () {
     n=0;
     PORTA.DIRCLR = 0x04;
     while(!(VPORTA.IN & 4)){n++;}
     PORTA.DIRSET = 0x04;
     if (n>1){
         if (n<500){
             Serial.print((n-1)*20);
             Serial.println (" pF");
         }
         else{
             Serial.print(n/50);
             Serial.println (" nF");
         }
         delay(500);
     }
 }
```

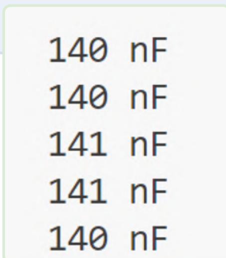


Figure 2: The output of Listing 1.

Tests

The output appears line by line (**Figure 2**). A film capacitor with a nominal value of 150 nF was displayed as 140/141 nF, which is still within the usual tolerance of 10%.

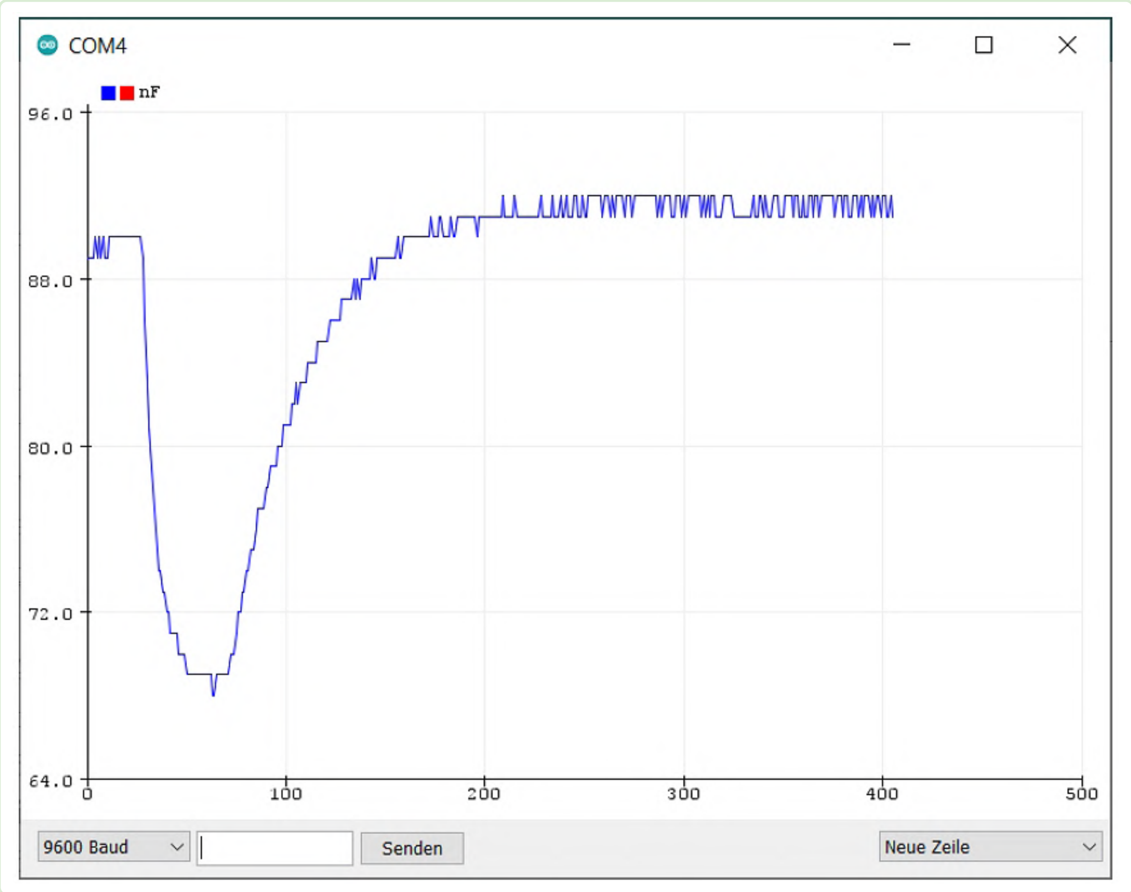


Figure 3: Capacitance change in the Arduino plotter: Heating between two fingers caused the capacitance to drop below 70 nF and slowly rise above 90 nF again, as it cooled down.

High-capacity ceramic capacitors are known to have a high temperature coefficient. This was demonstrated for a relatively small 100-nF disc capacitor using the Arduino plotter (**Figure 3**). Heating between two fingers caused the capacitance to drop below 70 nF and then slowly rise above 90 nF again as it cooled.

If a better resolution down to 1 pF is required, an external resistor of 1 MΩ can be used instead of the internal pull-up resistor. The port pin must then be operated as an input without a pull-up.

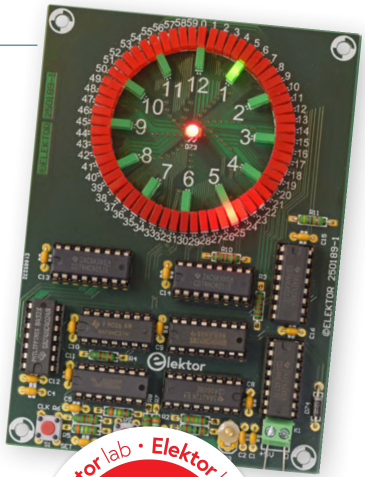
Translated by Jörg Starkmuth — 250213-01

Questions or Comments?

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

WEB LINKS

[1] Software download: https://www.elektormagazine.com/labs/elektor-articles-software-downloads
[2] Burkhard Kainka, "DDS Generator with ATtiny," Elektor Circuit Special 2025: https://www.elektormagazine.com/250211-01



20 Quasi-Analog Clockwork Mk II

Two LED Rings for Hours and Minutes

By Ton Giesberts (Elektor)
Original project by P. Hogenkamp

Once upon a time, a "Quasi-Analogue Clockwork" was published in 1995 in Elektor. This tribute to mechanical clockworks used a circle of LEDs to display the time. 30 years later with the "Quasi-Analog Clockwork" a remake was introduced. And now – in the same year – its little sibling, the "MkII" makes its debut. Read to learn what's new.

The MkII [1] is based on the recent remake [2] of the original Quasi-Analogue Clockwork [3], but with several improvements. It uses only 73 LEDs instead of 144, yet provides a one-minute resolution instead of five-minute increments. It includes a second pushbutton for faster time setting and has a significantly smaller PCB (80 × 113 mm). Like its ancestors, it relies entirely on standard logic. Enough enhancements to warrant a third version, wouldn't you say?

The Circuit

The updated circuit, shown in **Figure 1**, uses 60 red LEDs (in an outer circle) to indicate minutes and 12 green LEDs (in an inner circle) for hours, replacing the original 144 3-mm round LEDs in a single ring. Flat 2 × 5 mm flangeless LEDs allow a placement in a tight circle — less than 52 mm in outer diameter.

Figure 1 also includes the 14-stage binary ripple counter with oscillator CD4060 (IC1), which serves as the 32,768 Hz reference oscillator and initial divider. The 2 Hz output from IC1 acts as the clock for the 12-stage ripple counter 74HC4040 (IC2), which creates a one-minute pulse (output IC3A). This pulse then drives a 7-stage binary ripple counter (IC5, 74HC4024) that generates an hourly pulse (output IC3B). The one-hour pulse is the clock signal for two 74HC4017 (IC8 and IC9). This combination of two 5-stage Johnson counters allows sequentially driving 12 green LEDs.

More Details

IC1 functions as the reference clock, using a 32,768 Hz resonator in a Pierce oscillator configuration. Trimmer C3 allows frequency adjustment. The easiest way to verify the frequency is by measuring the 8 Hz signal at output CT11 (pin 1). For an exact clock, this frequency should be exactly 8 Hz, within a few ppm. The crystal used has a tolerance of ±20 ppm. The prototype is equipped with the type X32K768L104 from AEL Crystals.

IC3A resets IC2 to divide its clock input by 120, equals 8 + 16 + 32 + 64. When outputs CT3, CT4, CT5 and CT6 are high the output of IC3A is high and IC2 is reset via IC4C and IC4D. This creates the one-minute pulse. The clock can be set by pushing S1, which results

Features

Supply voltage:	5 V
Supply current:	<10 mA
Time display:	60 minutes in a circle, 12 hours in 2nd inner circle
Display minutes:	60 red 2×5 mm flangeless LEDs
Display hours:	12 green 2×5 mm flangeless LEDs
Seconds indication:	0.5 Hz blinking LED in center
Resolution (display):	1 minute
Technology:	Eight HC-logic series ICs, one 4000-logic series IC
Reference signal:	32,768 Hz quartz oscillator, adjustable
Set clock:	2 pushbuttons, 1 minutes step and oscillator for faster setting
Programming needed:	not at all

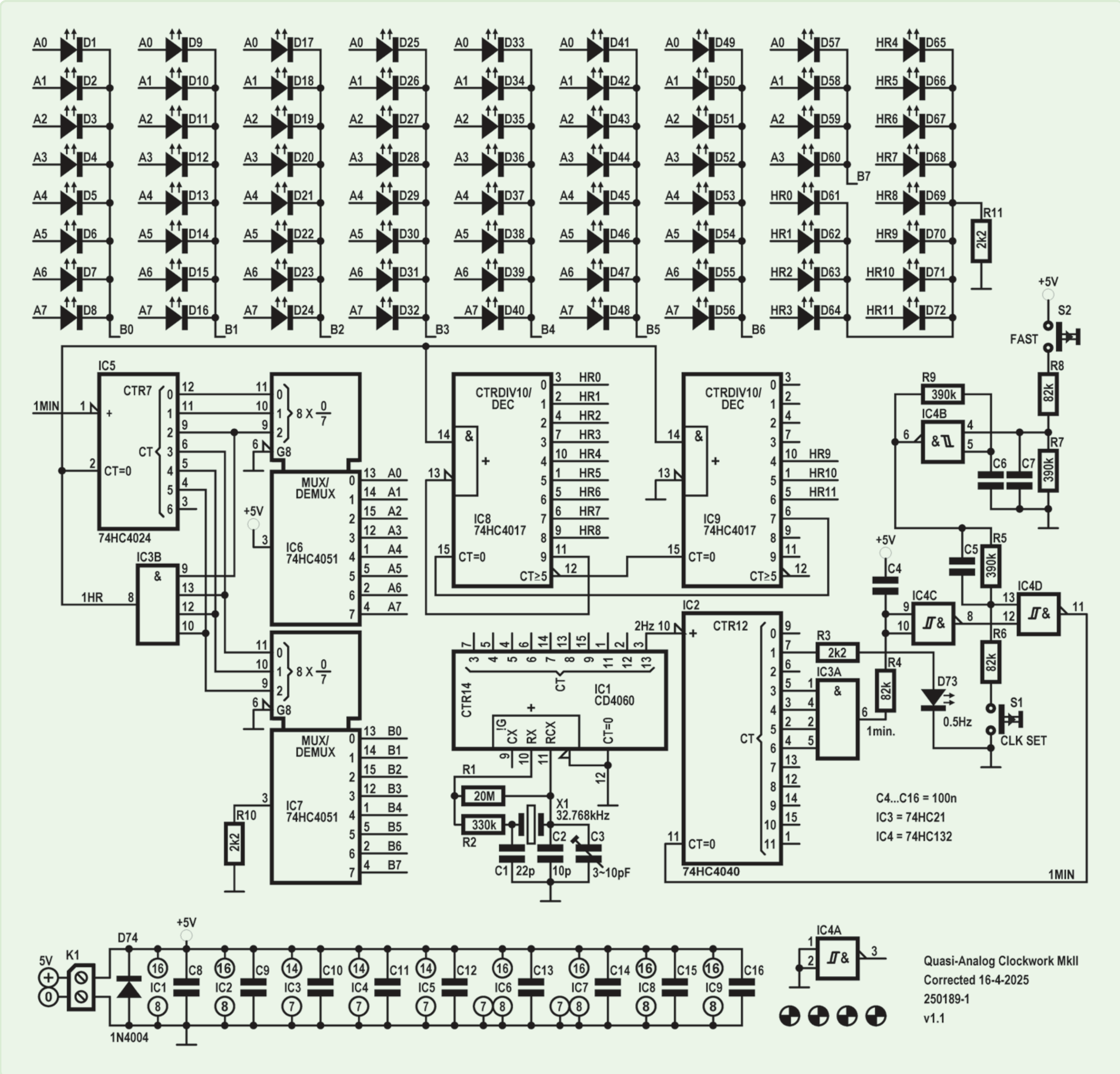


Figure 1: Schematic of Quasi-Analog Clockwork MkII (250189-1 v1.1)

in an additional reset pulse through debounce network R6/R5/C5. The Schmitt-Trigger input of IC4D prevents glitches. As IC4 is a quad NAND gate and hence inverts, an additional inverter is needed for the output of IC3A pass through to the reset un-inverted, hence IC4C.

Setting the clock after power-up may require up to 719 pulses via S1 — not very convenient. Therefore, the second push button S2 offers a faster alternative. It activates oscillator IC4B, which then drives the debounce network of IC4D. If the oscillator is inactive, the output remains high, IC4D stays low, and S1 remains usable. The oscillator frequency is around 30 Hz but varies depending on the Schmitt-Trigger's threshold. In the prototype, setting a full 12 hours takes 24 seconds. Adjust R9 to change the speed — lowering its value increases speed. Use S1 for the final minutes.

The circuit around IC5 works similarly to the one around IC2. To create a one-hour pulse, IC5 must divide the one-minute pulse at the clock input by 60 (4 + 8 +16 + 32). When outputs CT2, CT3, CT4 and CT5 are high, the output of IC3B is briefly high, resetting IC5 and generating the one-hour pulse.

The first six outputs of IC5 control analog switches IC6 and IC7 (both 74HC4051,

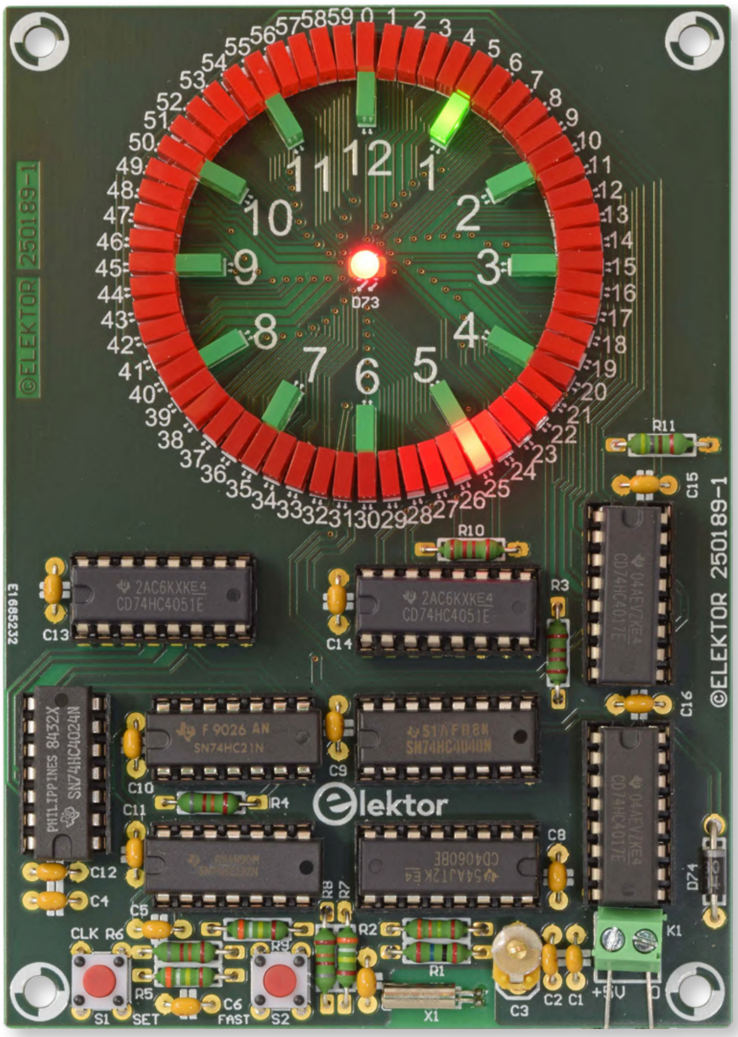


Figure 2: Quasi-Analog Clockwork with all parts mounted. Time is 1:25.

8-channel analog multiplexer/demultiplexer). The eight independent inputs/outputs of IC6 are connected to the anodes of the LEDs in eight groups of eight (signals A0 to A7), while the eight independent inputs/outputs of IC7 are connected to the common cathodes of the beforementioned groups of eight LEDs (signals B0 to B7). The common input/output of IC6 is connected to the +5 V and the common input/output of IC7 is connected to ground via R10, which sets the current of the red LEDs. Only one LED is lit at any moment in this 8 × 8 matrix. The three LSB outputs of IC5 (CT0, CT1 and CT2) drive the three digital select inputs of IC6 and the next three outputs of IC5 (CT3, CT4 and CT5) drive the three digital select inputs of IC7. When IC5 is reset, LED D1 (0 minutes) lights up. After 59 counts, LED D60 lights up, indicating 59 minutes.

The one-hour pulse is the clock signal for IC8 and IC9. Only a single output of a 4017 is active at the same time (1 of 10). The active low carry output pin 12 of IC8 is used to suppress the initial five counts of IC9. As long as pin 12 of IC8 is active, IC9 remains reset and its output 0 (pin 3) remains active during this time. IC9 advances the first time after reset when output 6 (HR6) of IC8 becomes active. So, output 1 (pin 2) of IC9 is equivalent to HR6 of IC8 – both are active at the same time. Outputs 4, 5 and 6 of IC9 are the signals HR9, HR10 and HR11 (pins 10, 1, and 5) respectively. Output 9 of IC8 is connected to the active low clock input (pin 13, high-to-low edge-triggered) to prevent counting any further until output 7 of IC9 resets IC8, which then immediately resets IC9. Each of the 12 outputs of IC8 and IC9 drives a single green LED. Output 0 of IC8 is active at 12 o'clock (LED D61).

A quick note about the seconds indicator LED D73: In the original 1995 version, its counterpart D163 was flashing at quite a different frequency: 50 Hz/2^8 ≈ 0.195 Hz. Like the remake from 2025, LED D73 is flashing at a frequency of exactly 0.5 Hz, so 1 s on and 1 s off.

PCB

Despite housing nine ICs, the PCB is more compact at 80 × 113 mm. See how densely it is populated in **Figure 2**. To reduce a parasitic capacitance between the ground plane and the crystal, there's a void in the ground plane where the crystal is. In its predecessor, this was the most likely cause of the oscillator of IC1 not starting at power-up when the crystal was placed flat against the PCB. Because its leads are thin, it's best to secure crystal X1 with a drop of glue. Depending on how the PCB is placed inside an enclosure or in a stand, the screw terminal block K1 for the 5 V power supply can also be mounted on the back. The labels 0 and +5 V are also printed at the bottom. D74, a 1N4004 next to K1 protects against wrong polarity. When placing IC sockets and ICs, look closely at the correct orientation – especially at IC1, which is rotated 180° compared to the other ICs!

Construction

In the SMD era, this advice still applies: start placing with the lowest-profile components. Suggested order:

- All resistors.
- Diode D74 (1N4004).
- All IC sockets (observe the correct orientation of IC1).
- Capacitors.
- Switches S1 and S2.
- LEDs D1...D60 (2×5 mm, red).
- LEDs D61...D72 (2×5 mm, green).
- LED D73 (3 mm, round).
- Trimmer capacitor C3.
- Screw terminal block K1.
- 32768 Hz Crystal X1.

Fitting and Soldering the LEDs

Very important: Place the **cathode** (short lead) of **red LEDs** D1...D60 **outward**!

Solder one lead of all the red LEDs while pushing each lead to shift the LED outward.

Keep the LED perpendicular to the PCB. The holes are a little bigger than the diameter of the leads. After soldering, look if the LEDs line up perfectly and are equally spaced. Correct the ones that are not. Then solder the other 60 leads. The pads of the rectangular LEDs are small and thinner solder, like 0.35 mm, is advised to be used here.

Next place the **cathode** (short lead) of **green LEDs** D61...D70 **inward!**

Perform the same handling as described for the red LEDs. Solder one lead of all the green LEDs and check if they are placed correctly. Then solder the other leads.

To match the brightness of red and green LEDs, in my case the current through the green LEDs was increased to 6 mA by lowering the value of R11 from the standard 2.2 kΩ to 470 Ω. Depending on the LEDs used, you may want to change this value. The current of the red LEDs is set by R10 and is about 1.4 mA.

Now your clockwork is complete. Enjoy!

250189-01

Questions or Comments?

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

About the Author

Ton Giesberts started working at Elektuur (now Elektor) after his studies, when we were looking for someone with an affinity for audio. Over the years, he has worked mainly on audio projects. Analog design has always been his preference. Of course, projects in other fields of electronics are also part of the job. One of Ton's mottos is: "If you want to have it done better, do it yourself." For example, for a PCB design for an audio project with distortion figures on the order of 0.001%, a good layout is crucial!

Component List

Resistors
(all 250 mW, 2.5 × 6.8 mm)
R1 = 20 M
R2 = 330 k
R3, R10 = 2.2 k
R4, R6, R8 = 82 k
R5, R7, R9 = 390 k
R11 = 470 Ω

Capacitors
C1 = 22 p, 50 V, 5 %, ceramic, C0G/NP0, pitch 5 mm
C2 = 10 p, 50 V, 5 %, ceramic, C0G/NP0, pitch 5 mm
C3 = 3~10 pF, trimmer capacitor, 150 V, e.g. BFC280823109 Vishay
C4...C16 = 100 n, 50 V, ceramic X7R, pitch 5 mm

Semiconductors
D1...D60 = LED, red, flat, 2×5 mm, flangeless
D61...D72 = LED, green, flat, 2×5 mm, flangeless
D73 = LED, super red, round, 3 mm
D74 = 1N4004, DO-41 (DO-204AL)
IC1 = CD4060, DIP-16
IC2 = 74HC4040, DIP-16
IC3 = 74HC21, DIP-14
IC4 = 74HC132, DIP-14
IC5 = 74HC4024, DIP-14
IC6, IC7 = 74HC4051, DIP-16
IC8, IC9 = 74HC4017, DIP-16

Miscellaneous
K1 = 2-way PCB terminal block, pitch 3.5 mm
S1, S2 = 6 mm tactile push button
X1 = 32.768 kHz crystal, 20 ppm, Cload 12.5 pF, 8×3 mm cylinder package, e.g. X32K768L104 AEL Crystals
IC1, IC2, IC6...IC9 = IC socket, DIP16
IC3...IC5 = IC socket, DIP14
PCB 250189-1 v1.1

WEB LINKS

[1] Quasi-Analog Clockwork MkII, Elektor Labs: https://www.elektormagazine.com/labs/quasi-analog-clockwork-mkii-250189
[2] Ton Giesberts, "Quasi-Analog Clockwork," Elektor 3-4/2025: https://www.elektormagazine.com/magazine/elektor-409/63743
[3] P. Hogenkamp, "Quasi-analogue clockwork," Elektor 1/1995: https://www.elektormagazine.com/magazine/elektor-199501/33259

21 You Can Do Anything You Want

(with the Arduino Ecosystem at Your Side)

By The Arduino Team

From beginner builds to industrial applications, discover how Arduino boards are powering innovation at every level.

Arduino is an open-source platform made to help people bring ideas to life — from students building their first robots to professionals prototyping intelligent systems. Don't believe us? Check out these four projects to see the breadth of what's possible ... and keep exploring on Arduino Project Hub: a growing online library of open-source examples created and shared by people around the world.

Full-Sized Photovoltaic System — Smart Energy with Arduino Opta

This is an ambitious project developed by Arduino community member *arduinojacky*, for a photovoltaic installation that tracks and adjusts solar panel angles for maximum energy output. The field-tested system uses time, date, and geographical location data to optimize panel positioning throughout the day and improve the efficiency of renewable energy deployments thanks to rigorous and reliable automation.

Arduino Opta Lite.

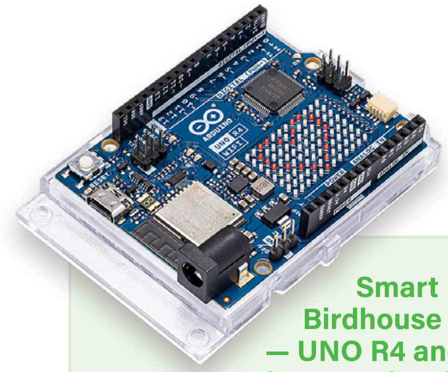
How It Works

The system utilizes a DS1307 real-time clock module to calculate the sun's position throughout the day, incorporating network time synchronization using NTP servers to maintain accurate timekeeping. After determining the optimal azimuth and elevation angles, actuators controlled by an Arduino Opta Lite — which handles all logic operations and edge computing — adjust panel tilt and rotation. The system employs potentiometers to monitor actuator positions, ensuring precise alignment. Safety features are integrated for protection during adverse weather conditions: during storms, for example, panels are automatically repositioned to a safe orientation.

Cloud connectivity, also supported by Opta, allows for remote monitoring. By combining real-time data processing, precise motor control, and environmental responsiveness, the system offers an efficient and reliable method for solar energy optimization through solar tracking systems.

Explore the full project:
https://tinyurl.com/full-sized-pv-system

Maximize solar energy capture with a smart system.



Smart Birdhouse — UNO R4 and Arducam Take Flight

What if your garden birdhouse could snap photos of feathered visitors and light up to welcome them? That's the idea behind this charming automation project powered by the Arduino UNO R4 WiFi and an Arducam camera module. Thanks to motion detection, the birdhouse captures images when a bird arrives and uploads them to the cloud. The user can then remotely trigger interior lighting — perfect for observation or photography. With its clever use of Arduino Cloud for remote interaction, this project can serve as a delightful entry point into embedded IoT design. Featuring just a few components, it's ideal for hobbyists looking to explore wireless control and real-time data communication in a whimsical and hands-on way.

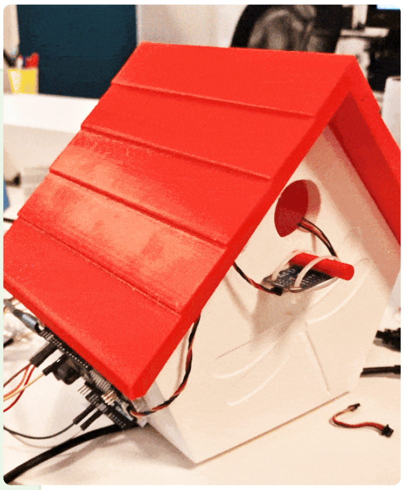
How It Works

Utilizing a Modulino Movement as a motion sensor, the system detects when a bird lands on the perch, triggering the camera to capture an image. This image is then processed and uploaded to a Replit server, which assembles the image and generates a URL. The URL is subsequently displayed on the Arduino Cloud dashboard, allowing users to view the captured image remotely.

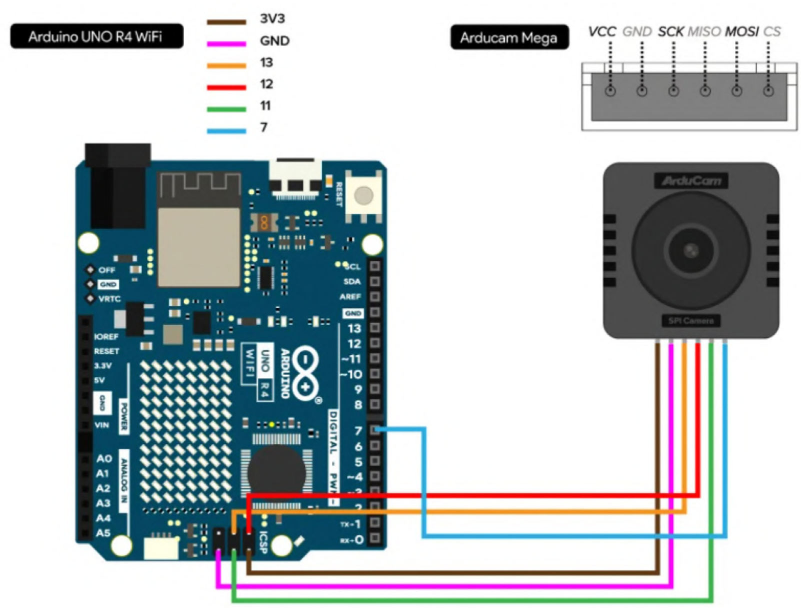
In addition, the Smart Birdhouse features controllable lighting using Modulino Pixels. Users can remotely adjust the lighting color and state through the Arduino Cloud dashboard. The system employs two cloud variables: `lampColor` (a CloudColor variable) to set the desired color, and `ledOn` (a boolean variable) to toggle the lights on or off.

All in all, the combination of hardware and cloud services creates an engaging IoT application and demonstrates how real-time data acquisition, processing, sensing, and remote control can be seamlessly integrated.

Explore the full project:
https://tinyurl.com/smart-birdhouse



Build your Smart Birdhouse with wood or 3D-printed pieces!



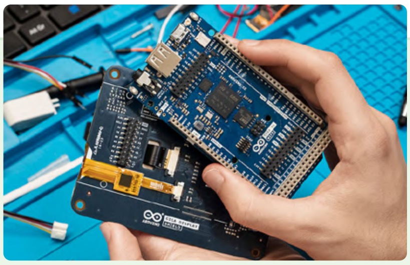
Connect the Arducam to the UNO R4 WiFi as shown.

Wavetable Synthesizer — A GIGA Sound Playground

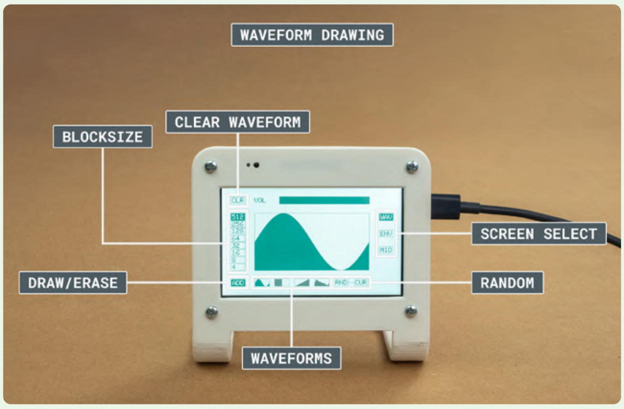
Step into the world of audio synthesis with this fully interactive polyphonic wavetable synthesizer, built using the Arduino GIGA R1 WiFi and the GIGA Display Shield. The project combines creative coding, responsive touch controls, and immersive sound design into a single, elegant setup to provide users with a hands-on experience in digital sound synthesis. With its dual-core STM32H7 microcontroller, the GIGA handles real-time audio generation with ease. Musicians and engineers alike can experiment with waveform blending, note sequencing, and tactile control, while the touchscreen interface makes interaction smooth and intuitive.

How It Works

At the core of the synthesizer is a wavetable engine that employs various waveform look-up tables, including sine, square, sawtooth, and organ pipe waves. These waveforms can be selected and modified to create diverse sound textures. The system supports 16-voice polyphony, allowing multiple notes to be played simultaneously, and features an ADSR (Attack, Decay, Sustain, Release) envelope generator for dynamic sound shaping. Users can interact with the synthesizer through the GIGA Display Shield, which provides a graphical interface for waveform selection and parameter adjustment.



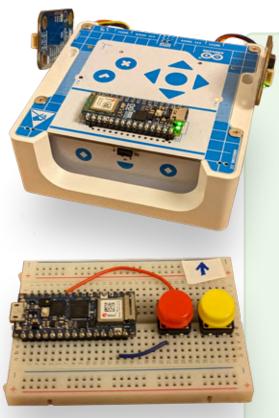
The GIGA R1 WiFi and GIGA Display Shield.



The main waveform editing screen.

The project is open-source and licensed under the Creative Commons Attribution-NonCommercial 4.0 International License, encouraging enthusiasts and developers to explore and expand upon the design. It exemplifies how Arduino's powerful boards can serve as the brain of sophisticated multimedia applications — all while remaining open-source and fully customizable.

Explore the full project:
https://tinyurl.com/wavetable-synthesiser



Arduino Alvik and its Nano RP2040 Connect-based remote.

Alvik Remote Control — Robotics Made Simple

Developed by community member *c_sarnataro_arduino*, this project pairs the compact and versatile Arduino Alvik robot with a custom-built Bluetooth Low Energy (BLE) remote controller coded in MicroPython — an intuitive, readable language perfect for learners aged 12 and up.

How It Works

Instead of traditional buttons or joysticks, this system employs the built-in Inertial Measurement Unit (IMU) of the Arduino Nano RP2040 Connect to detect tilt motions. These motions are then translated into directional commands for the Alvik, allowing intuitive control: tilting forward moves the robot forward, tilting right turns it right, and so on.

Both the controller and the Alvik are programmed using MicroPython, facilitated by the Arduino Lab for MicroPython environment, and cloning the necessary code from the provided GitHub repository. Additionally, the controller features a buzzer that plays a melody when a designated button is pressed, enhancing user interaction.

The controller app, running on a mobile phone or another microcontroller, lets users navigate the robot in real-time, offering a hands-on way to explore motion, control theory, and wireless communication.

With this project, students can go from wiring and scripting to autonomous movement and testing in a single classroom session, combining motion sensing and wireless communication to create an engaging and educational robotics experience.

Explore the full project:
https://tinyurl.com/alvik-remote-control

250477-01

About Arduino

Arduino (www.arduino.cc) is an open-source company dedicated to the design of hardware, software, and educational resources. Founded as an open-source project in 2005, Arduino has offices in Italy, Sweden, Switzerland, and the USA. Arduino supports makers, innovators, educators, and enterprises around the world in the creation of projects, courses, digital products, and services.

22 NEON Lamp Dice

By Clemens Valens (Elektor)

LED-based dice are common, but their light is cold. Not so for this electronic die, which displays its value with the warm glow of neon lamps — perfect for playing games on cold, dark winter evenings.

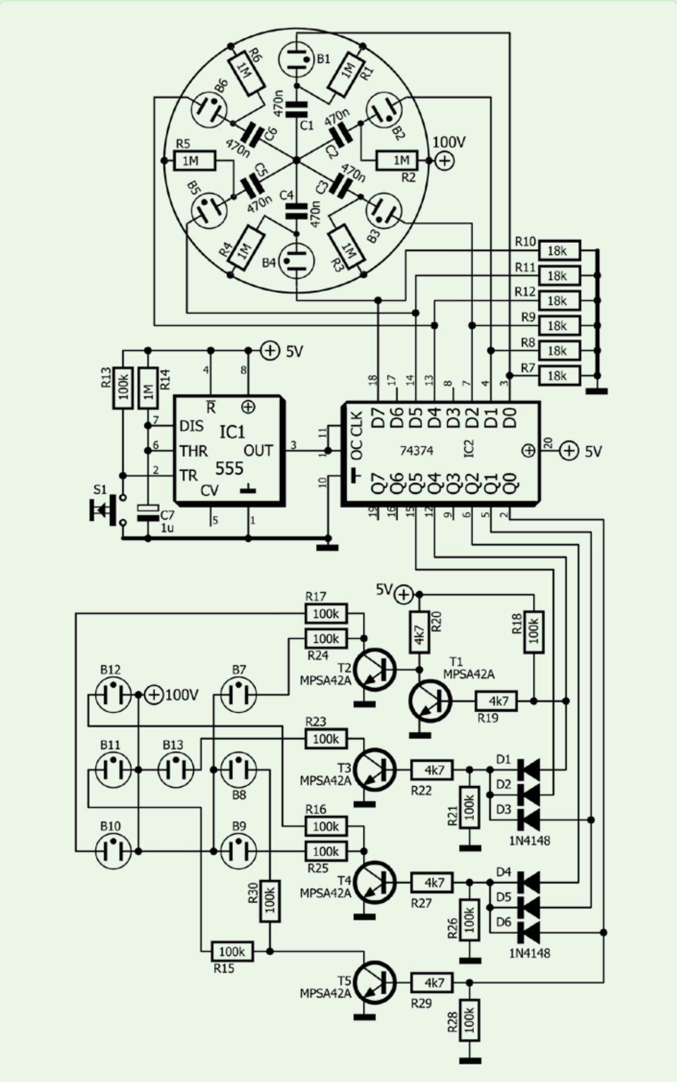


Figure 1: The schematic of the neon lamp dice. The ring counter is easy to spot. The sampler is just below it, while the die display is in the bottom half. T1 and T2 control the pips for the value 2.

A neon lamp is a curious device. It behaves like an open circuit when not conducting. When it starts conducting, the voltage across it drops significantly and remains fairly constant over a wide current range. This negative-resistance property allows a neon lamp to be used in, for example, relaxation oscillators. The circuit shown here is based on a semi-random ring counter that was popular in the sixties. [1]

Random Blinking

The counter operates as follows (see **Figure 1**): The supply voltage, about 100 V, is sufficient to trigger the neon lamps B1 to B6. When this happens for one of the lamps, the voltage across the bulb drops to around 60 V. This voltage drop is transmitted through the capacitor connected in series with the lamp to the central capacitor node, causing the voltage at this node to drop as well. Now the supply voltages of the other lamps are pulled below their breakdown threshold via the other capacitors. As a result, none of the other bulbs can switch on.

The voltage at the central node then gradually increases as the capacitor is discharged through the conducting neon lamp. Once it becomes high enough for one of the other lamps to switch on, the process repeats. Due to component tolerances, it is uncertain which lamp will activate next. Consequently, the blinking sequence is (semi-)random.

When pushbutton S1 is pressed, IC2 samples the current state of the ring counter. As only one lamp in the ring counter can be on at any time, only the output corresponding to the lit lamp will go high.

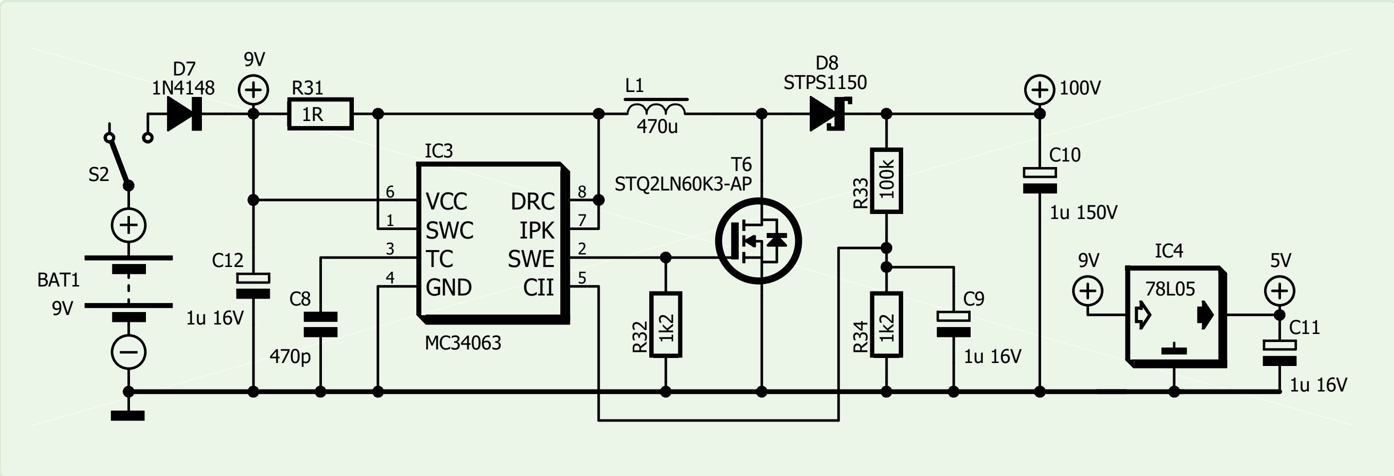


Figure 2: The (low-current) high-voltage power supply required to drive the neon lamps is based on a cheap MC34063 step-up converter. The 5 V supply is on the right.

The bit value at the outputs is translated into a die-like display using OR-ing diodes and transistors.

The MPSA42 transistors driving the pips are common high-voltage types that can withstand up to 300 V between the collector and the emitter. T1 can be a low-voltage type, but for the sake of sparing a line on the bill of materials we used a high-voltage one.

The 100k resistors in series with the neon lights limit the current through the pips to about 0.4 mA, so a six requires 2.5 mA. The total high-voltage current draw is less than 3 mA.

Neon Lamps

The neon lamps B7 to B13 are the pips of the dice. They come in pairs, except for the centre pip B13. Note that the pips for the value 2 (B7 and B10) should be on for all values higher than 1. Therefore, if the value is 1, T1 switches B7 and B10 off instead of using OR-ing diodes to switch them on for the five other cases. Because of this, the sampled output of value 2 is not needed.

IC1, a monostable multivibrator, was added to block multiple, fast key presses. Its output signal is also used to switch the pips off for about one second by driving the output control (OC) input of IC2 low; so it is clear when a fresh value is being displayed.

Power Supply

Figure 2 shows the 100 V power supply required for the neon lamps. It is created from a 9 V battery by step-up converter IC3 used in a classic configuration. The MC34063 has a switching transistor built-in, but it doesn't support high voltages, which is why T6 was added, a 600 V N-MOSFET.

Diode D7 provides reverse polarity protection. Capacitor C8 sets the switching frequency of the converter, around 75 kHz in this case.

Component List

Resistors

(THT, 150 V, 0.25 W)
R1...R6, R14 = 1 MΩ
R7...R12 = 18 kΩ
R13, R15...R18, R21, R23...R26, R28, R30, R33 = 100 kΩ
R32, R34 = 1.2 kΩ
R19, R20, R22, R27, R29 = 4.7 kΩ
R31 = 1 Ω

Capacitors

C1...C6 = 470 nF, 50 V, 5 mm pitch
C7, C9, C11, C12 = 1 µF, 16 V, 2 mm pitch
C8 = 470 pF, 50 V, 5 mm pitch
C10 = 1 µF, 150 V, 2.5 mm pitch

Inductors

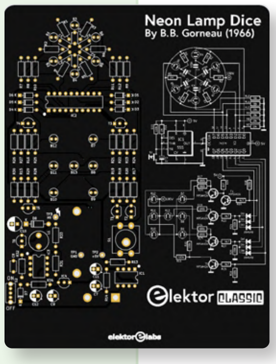
L1 = 470 µH

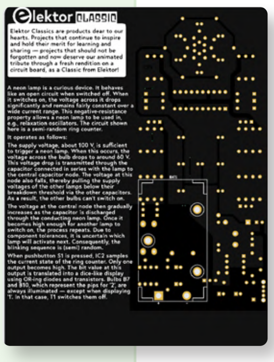
Semiconductors

D1...D7 = 1N4148
D8 = STPS1150
IC1 = NE555
IC2 = 74HC374
IC3 = MC34063
IC4 = 78L05
T1, T2, T3, T4, T5 = MPSA42
T6 = STQ2LN60K3-AP

Miscellaneous

K1 = PP3 9V battery holder
NE1...NE13 = neon light
S1 = Pushbutton, 12 mm by 12 mm
S2 = Miniature slide switch





Inductor L1 is charged when T6 is switched on by IC3. When T6 is switched off, the charge in L1 is transferred via D8 into C10. R33 and R34 form the voltage divider that determines the output voltage according to the equation V_{OUT} = (R33/R34 + 1) × 1.25 (i.e., 105 V).

This high-voltage supply can only deliver a few milliamps, just enough to drive the neon bulbs. The battery has to deliver about 50 mA max. and therefore should enjoy a long life. The user is safe too, also because IC3 features an overcurrent protection (R31). It is, of course, not recommended to touch the high voltage, but if you do, it won't do any harm.

IC4, a classic 5 V linear regulator, provides the 5 V for IC1 and IC2.

Assembling the circuit on its printed circuit board is not difficult. A good way is to start with the high-voltage supply and make sure it is working before continuing. The next step would be to assemble the ring counter and try it. If everything is fine at this point, then mount the remaining parts. Keep in mind that the battery holder is supposed to be mounted last and on the rear side of the board.

250130-01

Questions or Comments?

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

Related Product

- **Neon Lamp Dice**
  www.elektor.com/21197

WEB LINK

[1] B. B. Gorneau, "Willekeurige knipperaar," Elektuur 1/1966: https://www.elektormagazine.nl/magazine/elektor-196601/63770

ELEKTOR SUMMER CIRCUITS 40 YEARS AGO

elektor july/august 1985

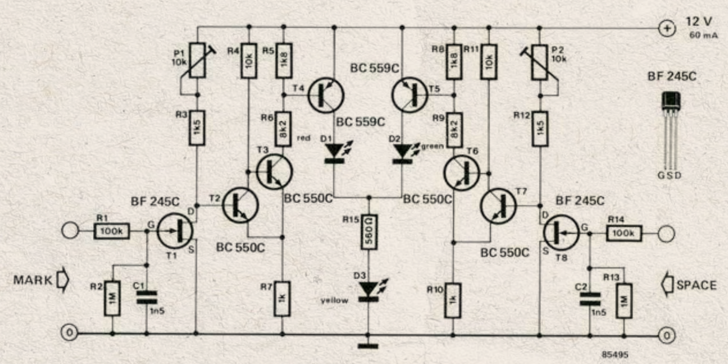
035

RTTY calibration indicator

To calibrate an RTTY (radio teletype) decoder correctly in accordance with the marks and spaces, an oscilloscope is needed. The mark and space signals are applied to the X and Y inputs of the instrument respectively, when, on correct calibration, the screen of the oscilloscope displays the well-known RTTY cross.

If an oscilloscope is not available, the circuit shown here can be used. It consists of two amplifiers with high-impedance input, T_1 and T_8, that are followed by driver stages $T_2 \ldots T_4$ and $T_5 \ldots T_7$. The driver stages control three LEDs, $D_1 \ldots D_3$ direct. Diode D_1 (red) is the mark indicator, D_2 (green) is the space indicator, and D_3 (amber) indicates whether the decoder has been calibrated symmetrically.

Preset potentiometers P_1 and P_2 determine the amplification of the field-effect transistors. Proper setting of these components enables the indicator to be matched with the filter outputs of any RTTY decoder.



After the indicator has been coupled to the RTTY decoder, that unit can be calibrated as follows:

- tune the short-wave receiver to the marks; the BFO knob must be adjusted until the red and amber LEDs both flash brightly;
- the RTTY decoder is then adjusted to the correct frequency deviation, indicated by the flashing of the green LED. If the amber LED lights continuously, the decoder has been calibrated correctly. Otherwise, the above procedure should be repeated carefully.

23 Inspiring Hardware Designs for Your ESPs

By The Espressif Systems Team

Espressif Systems empowers developers with cutting-edge wireless SoCs and open-source tools. The following projects highlight how Espressif engineers and community members push the boundaries of embedded design. These creations showcase the versatility and power of the Espressif platform in versatile applications.

ESP-Knob: Universal Smart Rotary Controller

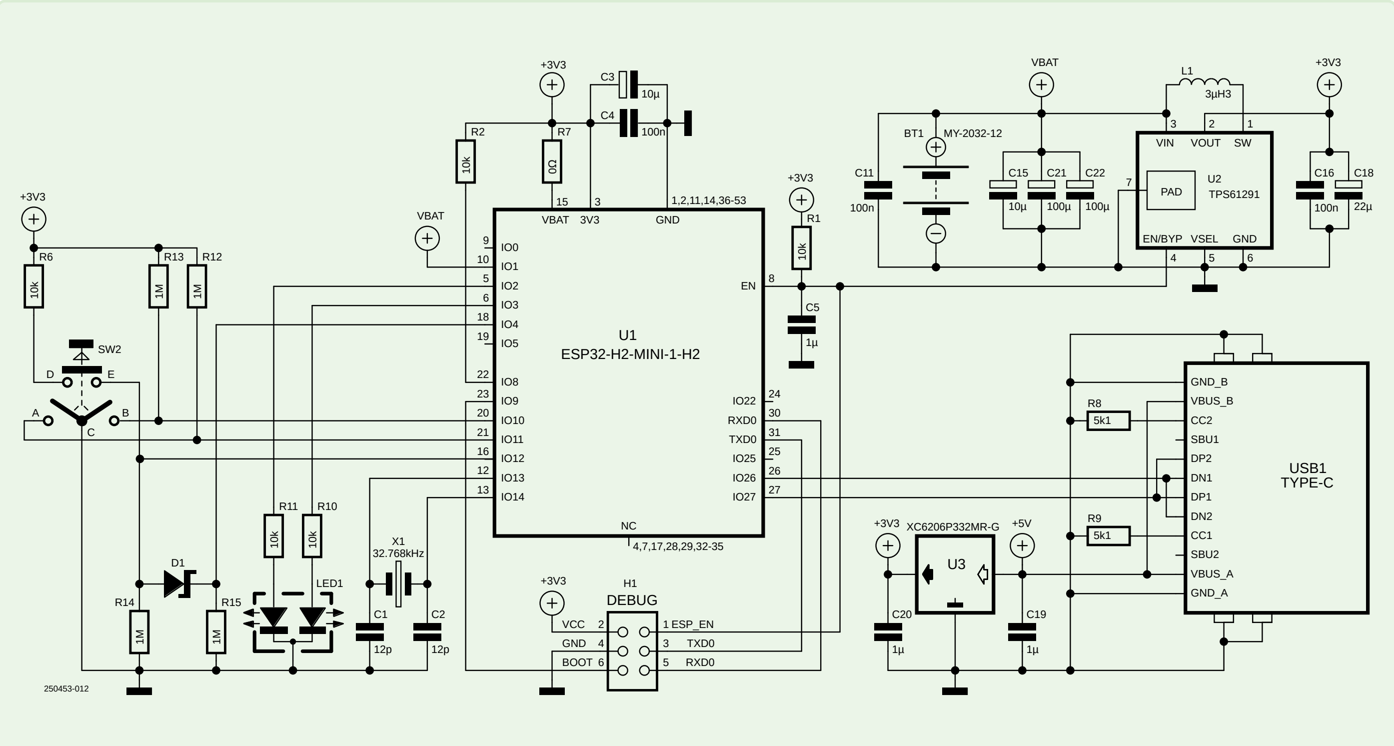
The ESP-Knob is a sleek, multifunctional rotary controller designed around the ESP32-H2 SoC. It's intended to work across various smart home ecosystems, including Matter, BLE-Mesh, Zigbee, and BLE-HID, acting as a cross-platform interface for lights, scenes, and appliances.

Key Features:

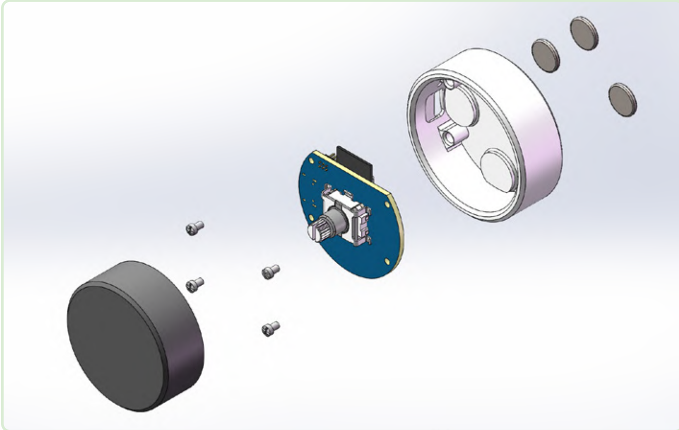
- BLE, Zigbee, Matter, and BLE-Mesh support
- BLE-HID for PC or mobile device interaction
- Press and rotation sensing
- USB-C or button-cell-powered operation
- Magnetic base for flexible placement

Hardware Design

The ESP-Knob is built around the ESP32-H2 SoC, selected for its integrated 802.15.4 and Bluetooth LE capabilities. It features a push-rotary encoder mounted on a compact circular PCB. A sleek

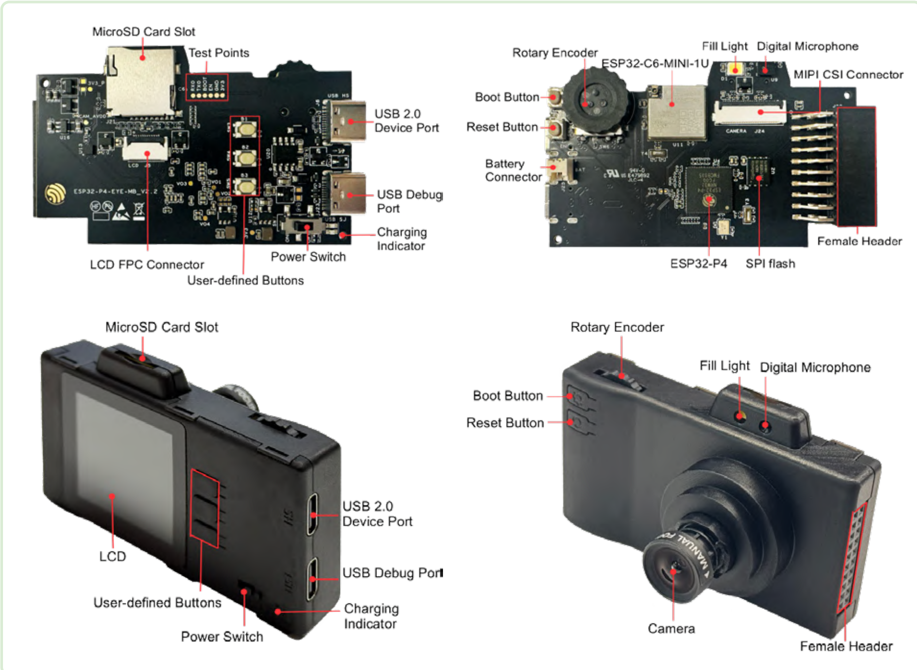


enclosure provides both tactile feedback and a premium aesthetic, while a strong magnetic base allows for vertical or horizontal mounting on metal surfaces. Power is supplied either via a CR2032 coin-cell battery for wireless operation or through a USB Type-C connector, which also supports firmware flashing and debugging. The board includes an RG LED ring for visual feedback and onboard circuitry to support low-power operation across multiple wireless protocols. For more information, visit *https://developer.espressif.com/blog/2025/03/esp32h2-knob.*



ESP32-P4-EYE: Vision Board for Edge AI

The ESP32-P4-EYE is a compact development board featuring the new ESP32-P4 SoC. It's built for AI applications that require real-time image processing, high-speed peripheral access, and efficient edge inference—ideal for smart vision and human-machine interfaces.

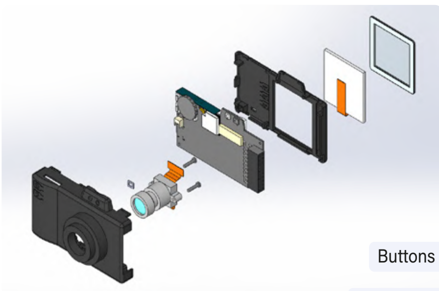


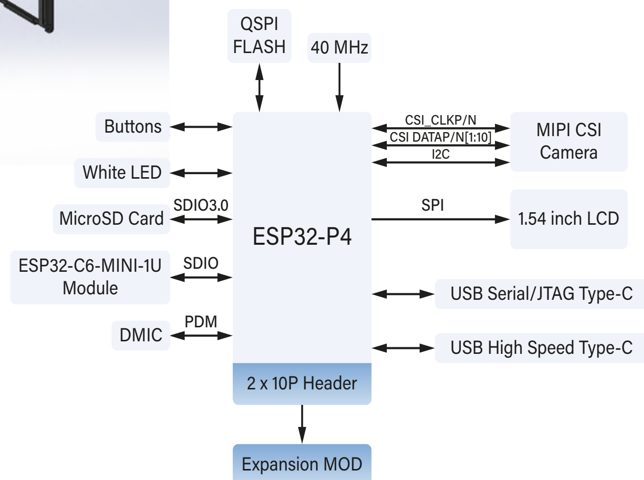
Key Features:

- Dual-core RISC-V processor (400 MHz)
- MIPI CSI/DSI, H.264 encoding, USB 2.0 OTG
- Built-in camera, display, mic, and speaker
- Supports up to 32 MB PSRAM
- Companion ESP32-C6 module for wireless (Wi-Fi, BLE, Zigbee, Thread)

Hardware Design

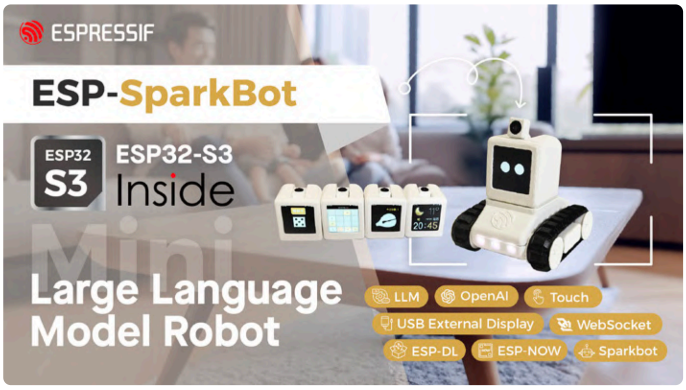
The ESP32-P4-EYE is a densely integrated board leveraging the ESP32-P4's high-speed peripheral interfaces and processing power. It includes a GC032A camera (MIPI-CSI), an ILI9488 480 × 320 display (MIPI-DSI), MEMS microphone, speaker, and MicroSD card slot—all mounted on a 47.6 × 39.6 mm board. The ESP32-P4 is paired with up to 32 MB of PSRAM to support image buffering and AI inference. A separate ESP32-C6-MINI-1U module provides wireless connectivity (Wi-Fi 6, BLE, Zigbee, Thread), making it suitable for low-latency streaming and over-the-air model updates. USB-OTG and serial ports enable development and debugging, rounding out this board as a full-featured vision-AI platform. For more information, visit *https://tinyurl.com/esp32-p4-eye-user-guide.*





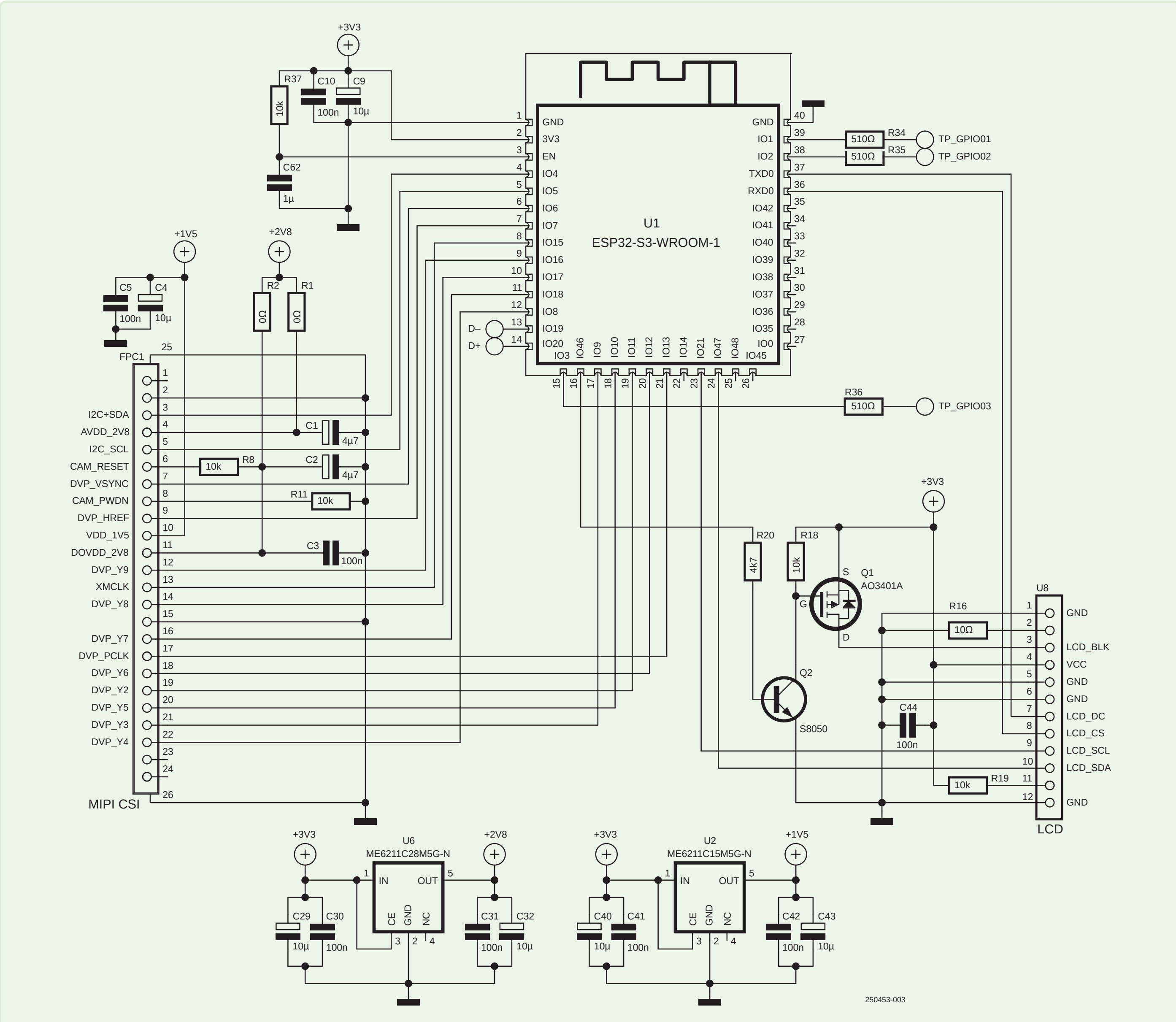
ESP-SparkBot: AI-Powered Desktop Companion

The ESP-SparkBot is an innovative, AI-powered robot built around the ESP32-S3 SoC, designed to showcase intelligent interaction, edge AI capabilities, and modular design. It combines speech interaction, facial recognition, local AI inference, and playful functions to act as a personalized desktop assistant or reconnaissance bot.

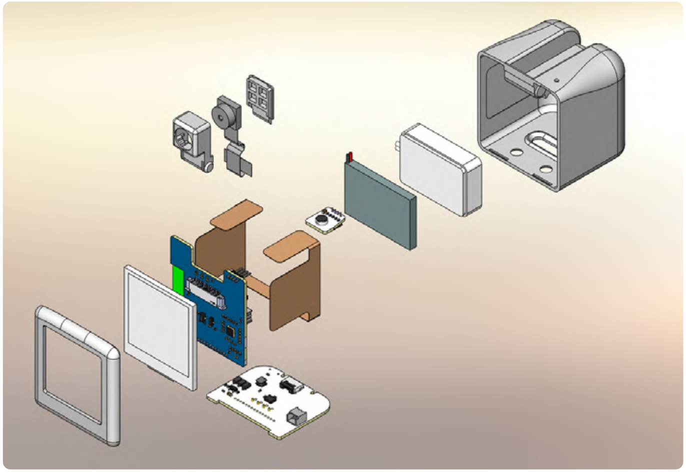


Key Features:

- Weather/time voice assistant
- LLM-powered AI chatbot
- Face recognition and 3D dice simulation
- Real-time camera streaming and USB display mirroring
- Modular design that converts into a smart RC vehicle



To view the full schematic, visit the link https://developer.espressif.com/blog/2025/04/esp32-s3-sparkbot..



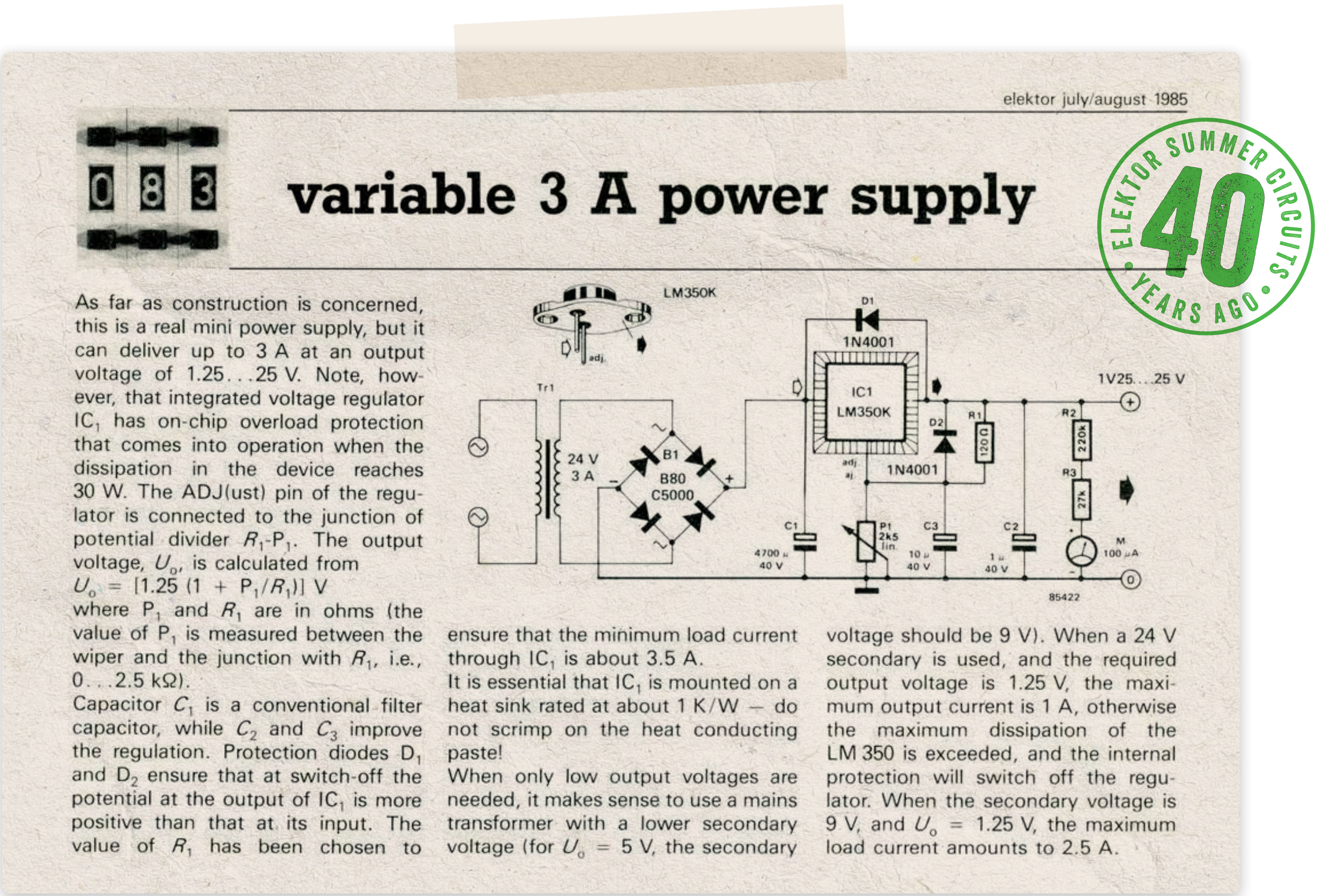
Hardware Design

The ESP-SparkBot is engineered with a three-board modular architecture. The head unit features a 1.28" round LCD, 2-MP camera, audio amplifier, and speaker, all controlled by an ESP32-S3 SoC that supports USB-OTG and LCD/CAM interfaces. A separate mic board connects via a 4-pin header, while the base board houses a 6-axis IMU, battery charger, Type-C interface, motor driver, and DC-DC converter. Magnetic pogo pins allow for tool-free assembly and power/data transfer between the base and head. The design supports peripheral expansion and motion capabilities, enabling the robot to serve as both a desktop assistant and a mobile reconnaissance unit. For more information, visit *https://developer.espressif.com/blog/2025/04/esp32-s3-sparkbot*.

250453-01

About Espressif Systems

Espressif Systems (www.espressif.com) is a public, multinational, fabless semiconductor company established in 2008. We develop cutting-edge, low-power wireless communication chipsets, with offices in China, Singapore, India, Czechia, and Brazil. Our AIoT solutions are green, versatile, and cost-effective. We have a closed-loop development cycle for core technologies, including Wi-Fi & Bluetooth LE & IEEE 802.15.4 protocols, RF, RISC-V MCUs, AI algorithms, operating systems, toolchains, AIoT frameworks, and Cloud services.



elektor july/august 1985

083

variable 3 A power supply

As far as construction is concerned, this is a real mini power supply, but it can deliver up to 3 A at an output voltage of 1.25...25 V. Note, however, that integrated voltage regulator IC_1 has on-chip overload protection that comes into operation when the dissipation in the device reaches 30 W. The ADJ(ust) pin of the regulator is connected to the junction of potential divider R_1-P_1. The output voltage, U_o, is calculated from

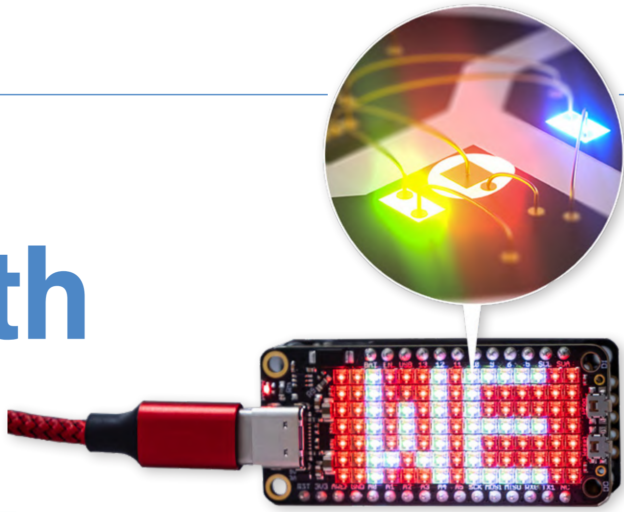
$U_o = [1.25\ (1 + P_1/R_1)]$ V

where P_1 and R_1 are in ohms (the value of P_1 is measured between the wiper and the junction with R_1, i.e., 0...2.5 kΩ).

Capacitor C_1 is a conventional filter capacitor, while C_2 and C_3 improve the regulation. Protection diodes D_1 and D_2 ensure that at switch-off the potential at the output of IC_1 is more positive than that at its input. The value of R_1 has been chosen to ensure that the minimum load current through IC_1 is about 3.5 A.

It is essential that IC_1 is mounted on a heat sink rated at about 1 K/W — do not scrimp on the heat conducting paste!

When only low output voltages are needed, it makes sense to use a mains transformer with a lower secondary voltage (for U_o = 5 V, the secondary voltage should be 9 V). When a 24 V secondary is used, and the required output voltage is 1.25 V, the maximum output current is 1 A, otherwise the maximum dissipation of the LM 350 is exceeded, and the internal protection will switch off the regulator. When the secondary voltage is 9 V, and U_o = 1.25 V, the maximum load current amounts to 2.5 A.



RGB LEDs with Integrated Control Circuit

Light with Precision: ICLEDs Set Standards

By Carlos Roberto Hernández Gómez (Würth Elektronik eiSos)

Additive color mixing with red, green and blue LEDs grouped into pixels has opened new possibilities for displays and lighting designs. Since the first modules with an integrated circuit (IC) appeared around ten years ago, the solutions have not only become more compact, but also more efficient and more precisely controllable.

Signal control systems, full-color matrix displays, audio and gaming devices, ambient lighting systems or displays on charging stations are increasingly relying on adjacent red, green and blue LEDs as pixels instead of the pixels of an LCD. ICLEDs — also known as addressable, smart or pixel LEDs — are light-emitting diode packages in which an integrated circuit (IC) is included. This driver uses pulse width modulation (PWM) for individual control of the red, green and blue chips in the package and therefore enabling sophisticated and precise lighting solutions with different colors. Because each pixel color can be individually regulated with brightness from 0 to 100 percent and digitally dimmed, more than 16 million different color and brightness values can be generated with the integrated RGB LEDs. And as fewer components are required, display solutions with intelligent LEDs are not only quicker to implement but also more energy-efficient than conventional LEDs. Digital control increases the flexibility of what can be displayed. For example, illuminated traffic signs could display any text or symbols instead of being limited to a few preset images.

Quality Is Crucial

Anyone entering a market at a later stage must set themselves apart with clear differentiating features. Würth Elektronik therefore not only offers a special service but also relies on the highest quality standards. With its Horticulture LEDs, the manufacturer attracted attention by offering not only the LEDs themselves, but also solutions for the associated control. The company also has its own research department dedicated to developing light recipes to optimize plant growth. For its ICLEDs [1], which are available in four designs, Würth Elektronik placed particular emphasis on high-quality processing, such as gold plating for chip LED versions and silver plating for PLCC derivatives to improve solderability. Comparative measurements with pin-compatible competitor components show a 40 percent higher illuminance. An annoying problem during assembly has also been tackled: In contrast to comparable solutions previously available on the market, Würth Elektronik's intelligent LEDs have a Moisture Sensitivity Level of MSL3 making them less sensitive to moisture — compared to industry standard MSL5 or MSL5a. This means that they can be used in the SMT line for a week. To qualify the LEDs for various applications, the manufacturer provides photobiological test reports in accordance with EN 62471:2008 and IEC 62471:2006 for all versions of the product group. These are relevant, for example, when the intelligent LEDs are used in toys. Although the product was not developed specifically for the automotive market, Würth Elektronik has also methodically carried out the tests and qualifications in accordance with AEC Q102-003. This provides customers with objectively comparable specifications.

Reproducible Colors

Good quality, high performance and features such as IPx7 protection for some versions are strong arguments in favor of WE-ICLEDs. But the crucial question is their practical effect in the application. This is precisely where Würth Elektronik comes in to open a new chapter in the use of smart LEDs. Smart LEDs offer new possibilities, however if the effort to use them is great, their potential will not be exploited in practice. Color deviations between different production batches in particular are a frequent challenge in ICLED applications. If the control of ICLEDs makes it possible to choose from 16 million color and brightness values, it should also be possible to reliably and reproducibly apply a specific color tone uniformly to a group of LEDs.

Digital Control

The typical application circuit of an ICLED system is shown in **Figure 1**. The data signals are generated by a microcontroller (MCU) whose data signal line is connected to the DIN pin of the first ICLED. The remaining components are connected to each other in a daisy chain connection. **Figure 2** shows that each time a data packet is received, the data is reduced by 24 bits by the integrated control circuit (IC) and the remaining bits are forwarded to the next LED in the chain. If a reset code is recognized at the DIN pin of the first IC LED, this means that the transmission of a new data packet begins.

This data determines the pulse width modulation for each individual red, green and blue LED. This means that the dimming is not controlled via the current intensity — as this would influence the color value — but by switching the current on and off quickly and imperceptibly. If the duty cycles that correspond to specific color mixing of each LED are known, any color value and brightness level can be generated. However, this was precisely the problem: application developers could not easily determine these values and did not receive any help from manufacturers.

Color Fidelity Thanks to ICLED Color Calculator

With the ICLED Color Calculator [2], Würth Elektronik has introduced a new function in its REDEXPERT online platform that significantly expands the possibilities of lighting and signal solutions with its ICLEDs. For the first time, developers can use this service

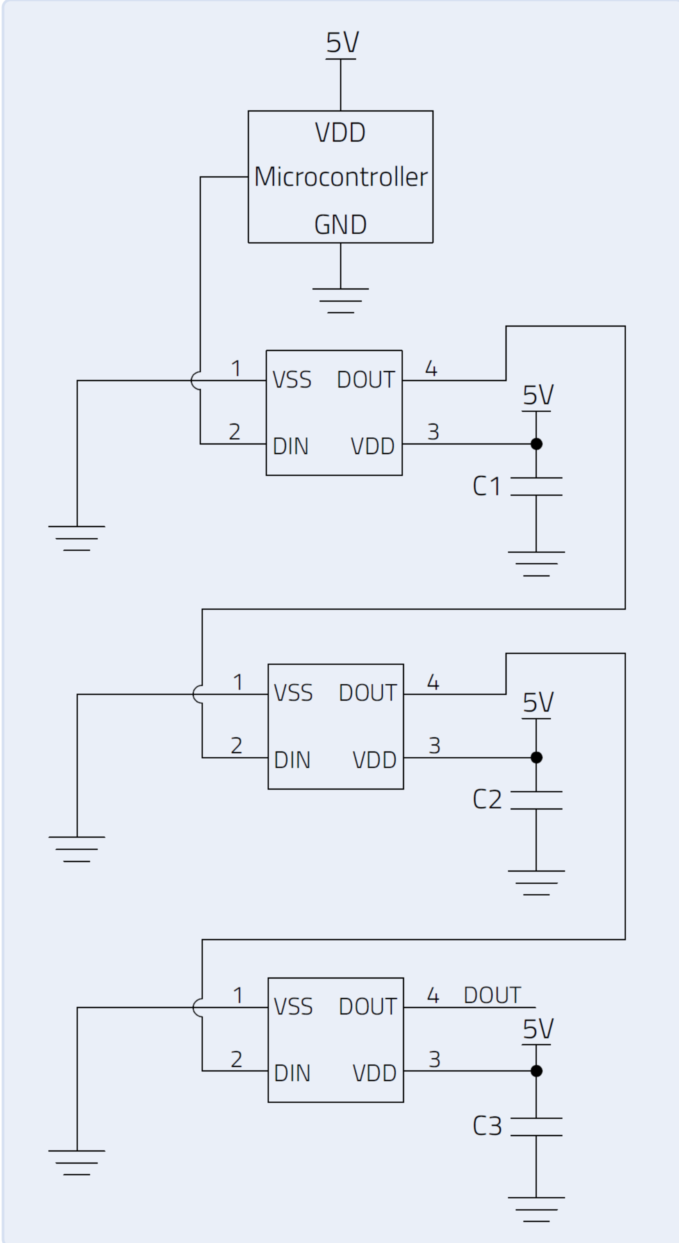


Figure 1: Typical daisy-chain circuit of an ICLED application.

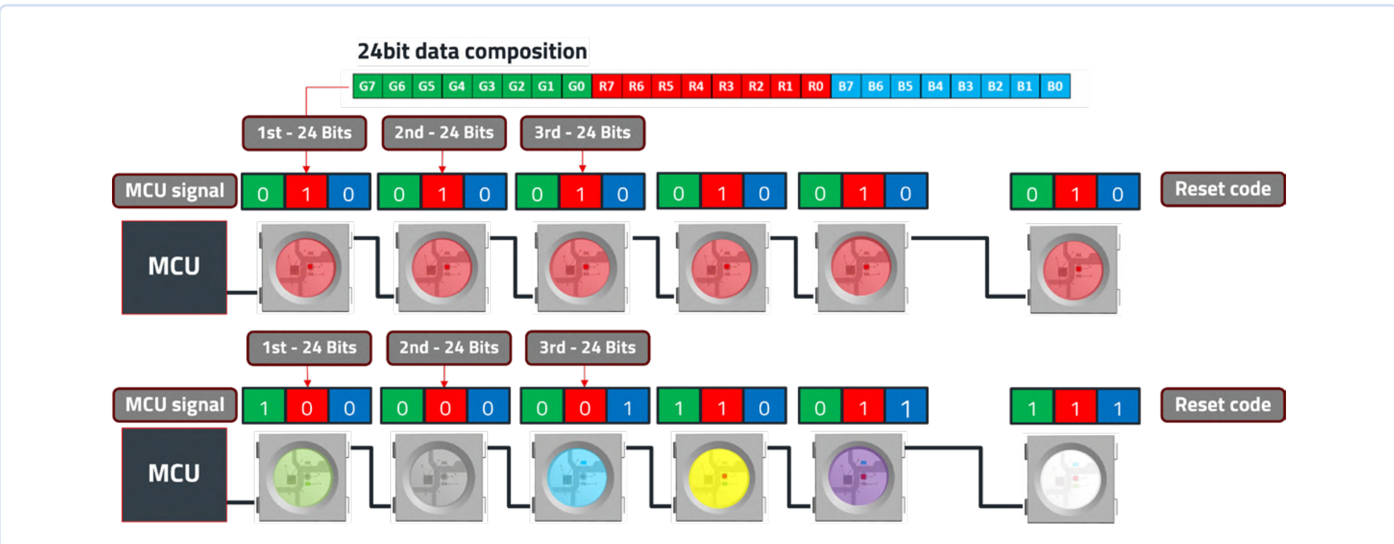


Figure 2: Principle of data transmission to a row of ICLEDs.

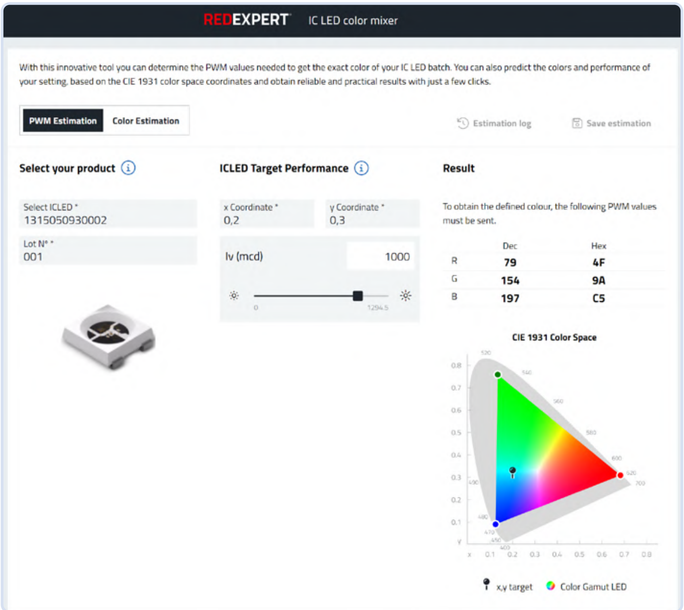


Figure 3: Select component, bin and color, and the REDEXPERT ICLED Color Calculator outputs the values for PWM control.

to easily determine the exact PWM values required to display a specific color value. On the intuitive user interface of the REDEXPERT ICLED Color Calculator (**Figure 3**), customers select the desired ICLED type as well as color and brightness in the CIE1931 standard color system ("xy color space"). This automatically generates the appropriate digital value for the pulse width modulation of the RGB LEDs installed in the ICLED component, based on the properties of their binning. These values can be transferred directly into the programming of the ICLEDs.

This is made possible by the careful sorting of the ICLEDs during production. There are numerous bins to ensure that only LEDs with uniform color temperature and brightness properties are offered per batch. Thanks to the precision of its quality control and binning, Würth Elektronik's innovative development tool opens new possibilities for sophisticated LED applications, such as high-quality LED matrix displays and applications that need to be perfectly color-matched to a corporate design.

Dual-Wire ICLEDs for Moving Images

Particularly in the field of LED matrix displays, the possibility of a true-color display is creating a desire for high frame rates to show moving images. This can now be realized with an ICLED variant that has seldom been offered until now and which Würth Elektronik now has in its product range: Dual-Wire ICLEDs.

This technology enables up to 13 times faster data transmission compared to single-wire ICLEDs. They work with fast PWM rates at 20 kHz to produce flicker-free images. Another advantage of the dual-wire ICLEDs is their integrated sleep mode, which reduces power consumption to around 1 µA per ICLED when they are not in operation. This ensures significantly more energy-efficient use, especially in battery-operated systems. In addition, the dual-wire technology offers a freely adjustable clock frequency of the data signal of up to 15 MHz, which enables flexible and fast data transmission. This combination of low power consumption, high adaptability and fast data transmission makes the dual-wire ICLEDs particularly attractive for modern smart lighting applications.

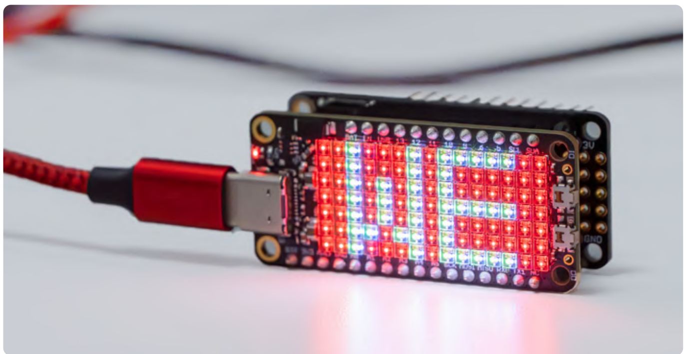


Figure 4: The ICLED Featherwing enables fast prototyping.

Of course, analog LEDs remain a lower cost alternative — around 20 to 30 percent lower than ICLEDs — and will not be displaced from simple applications. However, ICLEDs [3] offer clear advantages for signaling or optically demanding applications. Applications can be realized much more simply and economically in terms of the components and cables required. The precise controllability of ICLEDs also opens new fields of application. There is great potential in adaptive lighting, for example office lighting that automatically adjusts to weather conditions or the time of day. A color change triggered by a sensor can take on the role of a differentiated proximity alarm as an alternative signal for the hearing impaired is another example. Color-coded signals could also help to provide more information at a glance in health monitoring systems or diagnostic tools. An ICLED Featherwing is available for prototyping or for the rapid implementation of application ideas, which can be combined with various sensor and wireless Featherwings from Würth Elektronik to create an application (**Figure 4**).

250481-01

About the Author

Carlos Roberto Hernández Gómez began his academic journey in Mechatronics at the Universidad Popular Autónoma del Estado de Puebla, graduating in 2021. He advanced his education through a dual study program, combining efforts at Hochschule Aalen and Würth Elektronik, to earn a master's degree in Applied Photonics by 2023. His professional career at Würth Elektronik, initiated in 2020, has been focused on optoelectronics, particularly in IC LED technology. In 2023, he was appointed Product Manager for Optoelectronics, where he now leads the development of the IC LED product series, leveraging his unique dual-study experience to foster innovation and growth in this field.

WEB LINKS

[1] Product details: https://www.we-online.com/en/components/products/WL-ICLED
[2] REDEXPERT Tool: https://redexpert.we-online.com/we-redexpert/en/#/ic-led-color-mixer-embedded
[3] AppNote, Understanding parameters in ICLED datasheets: https://www.we-online.com/ANO009

24 Experiment: Towards a Mixed-Signal Theremin?

Blending Modern Time-of-Flight Sensors With the Timeless XR2206 Analog Generator

By Jean-François Simon (Elektor)

Earlier this year, I discovered with some amazement the time-of-flight (ToF) sensors from STMicroelectronics' VL53L series. For this Elektor Circuit Special project, I have chosen two VL53L4CD time-of-flight sensors to be the base of a gesture-controlled "pseudo-Theremin," combined with an XR2206 analog function generator. Let's discover how modern, digital sensors can interface nicely with classic analog circuits!

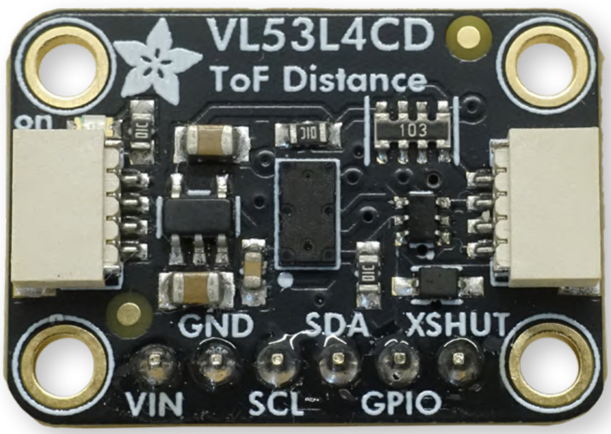


Figure 1: The VL53L4CD sensor on its breakout board.

STMicroelectronics's VL53L series, particularly the advanced VL53L8CX, are a good base for different kinds of touchless projects. Thanks to its multi-zone detection and wide field of view, the VL53L8CX can recognize human gestures. However, in this article, I'll focus on a simpler model, the VL53L4CD (**Figure 1**). For those unfamiliar with the term "time-of-flight," here's a brief explanation: these are distance sensors that measure the straight-line distance to an object by sending out a pulse of infrared light. A portion of the pulse is reflected by the object and returned to the sensor, which then measures the round-trip time. Considering the extremely short durations involved, as light travels about 3 mm in just 10 ps, this requires exceptionally fast electronics to achieve good resolution.

A Pseudo-Theremin

In this project, we'll build a kind of pseudo-Theremin, which, unlike the original instrument based on radio waves [1], uses two ToF sensors mounted on a fixed base. With a classic Theremin featuring two antennas, the pitch of the note is controlled by moving the right hand closer to or farther from the vertical antenna, while the volume is adjusted by varying the distance of the left hand from the horizontal antenna. Instead of antennas, two ToF sensors are used here, and both the frequency of the output signal and its volume (amplitude) are controlled by moving the hand closer to or farther from the corresponding sensor. Since these sensors are digital and use an I^2C interface, a microcontroller is required. The entire "instrument" could easily fit into an Arduino or ESP32 board equipped with a DAC, or output MIDI signals if discrete notes are preferred.

That's not the route I took! For the sheer fun of tinkering, I've chosen to use a microcontroller strictly as an I^2C-to-analog converter and leave sound generation to a good old XR2206 [2] (**Figure 2**). While no longer manufactured by Exar (its original developer), many fully functional clones from Asia are readily available.

The Distance Sensor

Among ST's VL53L series, I selected the VL53L4CD [3], a single-zone sensor optimized for short distances (up to approximately 1200 mm)

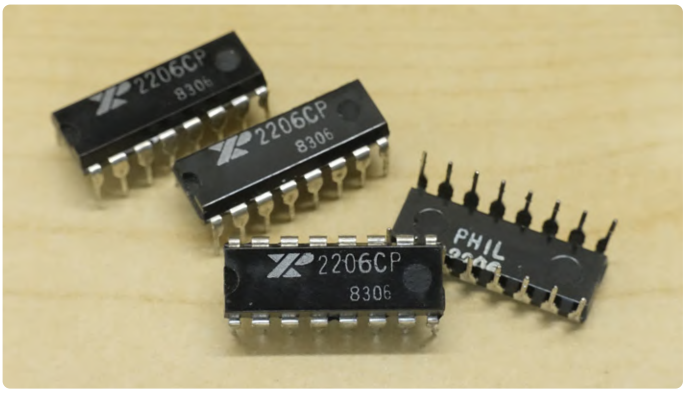


Figure 2: These old, but brand new XR2206s were generously donated by Mark B. (Thank you!)

with an 18° field of view. ST provides a library of readymade C functions to simplify setup, and the STM32duino open-source community on GitHub offers ready-to-use Arduino examples [4]. Because the sensor comes in a tiny BGA package and operates at 2.8-V logic levels, Adafruit wisely produced a breakout board [5] that includes a voltage regulator and level shifter, making it easy to use with any 3.3 V or 5 V microcontroller.

Circuit Diagram

The Theremin is 105 years old and the XR2206 is about 53 if I'm not mistaken. Continuing the nostalgic theme, I used an Arduino Uno R3, whose design has not changed since its launch 13 years ago (except that I use the SMD version, for no specific reason). The schematic (**Figure 3**) is very simple and should be easily adaptable to other platforms. It includes two VL53L4CD modules connected to the Arduino's SDA and SCL I²C lines. Since both sensors initially share the same I²C address, they must be individually powered on or off via their Shutdown pins to reprogram them with unique addresses, as we'll see later. Each sensor also features a GPIO that can trigger an interrupt under programmable conditions, but I'll not use it here.

The XR2206 is powered via pins 4 (12 V) and 12 (GND), with capacitors on the 12 V pin and on the internal voltage reference at pin 10. The internal oscillator generates a frequency which is of the form of 1/RC, where C is a capacitor connected between pins 5 and 6 (C1 in this circuit), and R is the resistor path between pin 7 and ground (R2 and R7). You can optionally connect an alternate resistor on pin 8 and activate it via pin 9, but these are left disconnected in this design.

The sine wave output appears on pin 2, with resistor R1 reducing distortion. Here, I found that a value of 200 Ω gives acceptable results. To generate a triangle wave, insert a switch between pins 13 and 14 to open this loop. Fine symmetry adjustment is available on pins 15 and 16, but is unused here. A square wave is also available via the open collector output on pin 11; to visualize or use this signal, add pull-up resistor R3. Resistor R6 limits the maximum output amplitude. 22 kΩ limits it to approximately 1.3 V peak, making it suitable for the line-level inputs on your amplifier or powered speakers. As we are using the XR2206 in single supply mode, it's needed to bias the output signal to half the supply voltage, which is done by the resistor divider R4/R5. Of course, any DC bias is unwanted at the output, so capacitor C7 blocks it before it reaches your amplifier.

Musical Range

The output frequency depends on the resistance seen by pin 7 of the XR2206, which is sourcing current, up to a maximum of 3 mA. The Arduino reads the distance from the "pitch" ToF sensor and outputs a PWM signal. This is filtered by R8 and C5 and drives the gate of

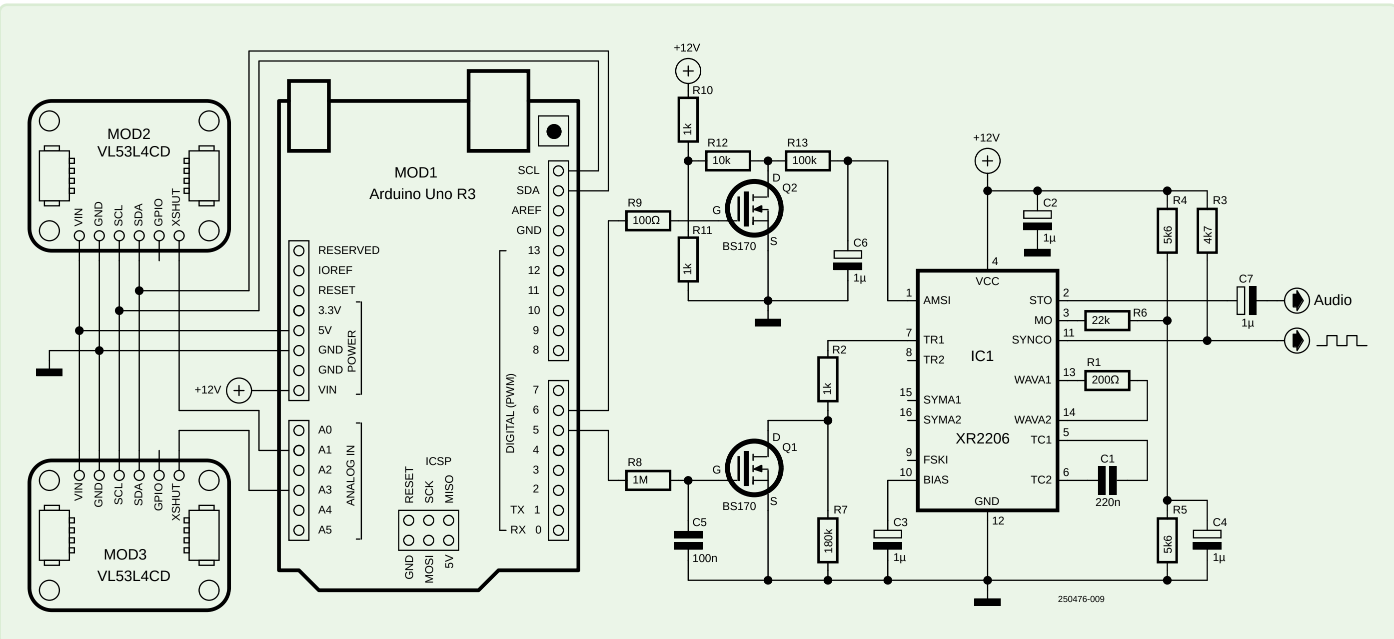


Figure 3: The schematic is quite simple.

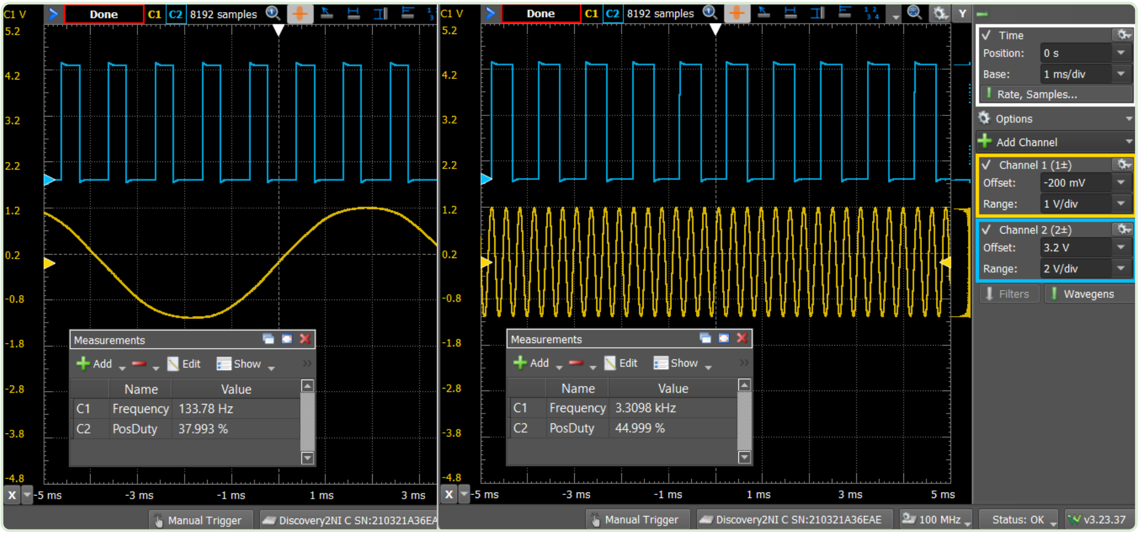


Figure 4: Problem: a small duty cycle variation modifies the output frequency too much.

transistor Q1, which partially shunts R7 to vary frequency (**Figure 4**), with a maximum set by R2. To diverge from classic Theremins which typically output signals from about 60 Hz to 3 kHz, I mimicked the range of a piano (27.5 Hz to 4.186 kHz). A 1-kΩ resistor (R2) and 220-nF capacitor (C1) work well for this.

Amplitude Control

The Volume control uses a similar approach (PWM, transistor, RC filter made of R13 and C6) applied to pin 1 of the XR2206. However, on the XR2206 the output volume peaks when V_{pin1} = 0 V or VCC, and is minimized when V_{pin1} = VCC/2, hence the actual arrangement of components is a bit different. The VCC/2 voltage is provided by the R10/R11 divider.

Software

The Arduino program benefits from STM32duino's API, particularly the functions `InitSensor()`, `CheckForDataReady()`, and `GetResult()`. The code in my Arduino sketch was intentionally kept simple, hoping it will inspire you to integrate other digital sensors into analog circuits. The beginning of the sketch details pin assignments, I²C addresses, and measurement ranges.

The code uses arrays like `sensors[]` to handle two sensors. If you're considering even more sensors, consider switching to a C `struct`. According to ST's documentation, all sensors must be turned off using `begin()` and `VL53L4CD_Off()` before initialization. Then use `InitSensor()` on each sensor in turn with a unique I²C address per sensor.

When that's done, the sensors are further configured using `SetRangeTiming(timing_budget_ms, inter_measurement_ms)`. The first parameter sets how long the sensor can spend on a measurement; the second sets the interval between readings. For better measurement accuracy, increase the timing budget; for faster updates, shorten the interval. I used 20 ms and 0 ms, respectively, but other values are worth experimenting with. This corresponds to 50 measurements per second (i.e., a refresh rate of 50 Hz). For applications requiring even greater responsiveness, the timing budget can be reduced to 10 ms (the minimum value allowed by the manufacturer). This gives the sensor a maximum speed of 100 Hz, but at the cost of reduced accuracy. It's also possible to combine a short timing budget with a long interval to lower power consumption.

The `loop()` function is shown in **Listing 1**. It uses `VL53L4CD_CheckForDataReady()` to check for new data, resets the sensor's internal interrupt (this is mandatory, even though we're not using the interrupt pin), and retrieves the results via `VL53L4CD_GetResult()`. The result report includes not only the distance in mm but also signal quality and status info. Read the guide at [6] for more detail.

To keep the program as flexible as possible, the measurement range (in mm) and PWM duty cycle bounds (in %) are configurable. The `map()` function is used to fit the measured values within these bounds. When playing on a proper Theremin, pitch is increased by bringing the right hand closer to the vertical antenna, and volume is decreased by bringing the left hand closer to the loop antenna. This is mimicked with the ToF sensors, but it's also possible to change that behavior by writing any combination of `True` or `False` in the `pwm_invert[]` array. Finally, the PWM output is generated by `analogWrite()` to control the XR2206. The full code is available on the Elektor Labs project page [7].

Construction and Testing

This project is mostly about having fun and encouraging creativity. Since the final "musical" instrument's quality is questionable, I don't provide a ready-made PCB for it. However, the circuit is easily assembled on a solderless breadboard (**Figure 5**) or on solderable perfboard. Make yourself a few adapters, using leftover parts of male and female header pins, to allow several wires to connect to the same Arduino pin. This is useful for the SDA, SCL, 5 V, GND lines for example and avoids wasting rows on the breadboard.

You can experiment with component values, in particular RC filters, using decade resistor boxes. I use those on a PCB where the resistor value is selected using jumpers, they're not perfect but still extremely helpful. To see what's going on, an oscilloscope is highly recommended. I used a PC-based Analog Discovery from Digilent for measurements, but any oscilloscope will do at these low frequencies. All that remains is to connect the output to an amplified speaker's line input and have fun. Be careful with the volume to avoid heart attacks!

In practice, producing actual "music" is very difficult. The proposed Q1/Q2 transistor implementation and their respective RC filters do not optimally use the full control range. For example, PWM values below 25% or above 50% on the frequency channel have no effect. You've read that right, 75% of the range is wasted. Even though the

Listing 1: The loop() function from the Arduino program.

```
void loop()
{
    for (int i = 0; i < NUM_SENSORS; ++i)
    {
        if (!sensors[i]->VL53L4CD_CheckForDataReady(&new_data_ready[i]) && new_data_ready[i])
        {
            int dist;
            sensors[i]->VL53L4CD_ClearInterrupt();
            sensors[i]->VL53L4CD_GetResult(&results[i]);
            if (results[i].range_status == 0)
            {
                dist = constrain(results[i].distance_mm, range_min[i], range_max[i]);
            }
            else
            {
                dist = range_max[i];
            }
            int pwm_val = map(
                dist,
                range_min[i], range_max[i],
                pwm_invert[i] ? pwm_max[i] : pwm_min[i],
                pwm_invert[i] ? pwm_min[i] : pwm_max[i]);
            analogWrite(pwm_pins[i], pwm_val);
         }
    }
}
```

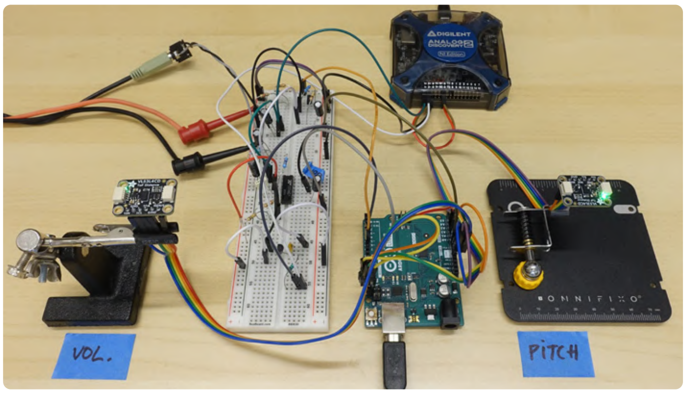


Figure 5: A view of the prototype, still on a breadboard.

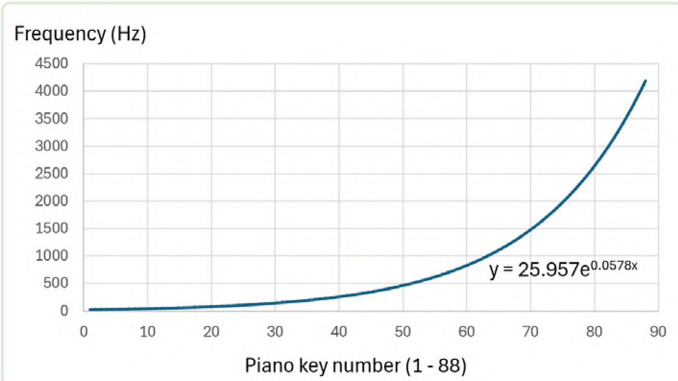


Figure 6: Piano frequencies follow an exponential curve, not a linear slope.

problem can be circumvented by software using the `map()` function, this significantly reduces the resolution of frequency control. Volume control fares better with 70% of the PWM range being usable. Due to time constraints, I had to stop development, but improvements are clearly possible! A faster PWM frequency could also be easier to filter out, and a DAC resolution greater than 8 bits could be implemented if necessary. Feel free to send me suggestions via email or on Elektor Labs.

Another issue is the linear mapping of distance to frequency, which makes playing the instrument even more difficult. On a piano, when sweeping across all keys the generated frequencies follow an exponential curve instead (**Figure 6**). Feel free to modify the code to implement such mapping or another, even better one. A polished version of the circuit would look great on an Arduino Uno-format prototyping shield (**Figure 7**). The Arduino can be powered by 12 V on its barrel jack input, which can also power the XR2206 on the shield through the VIN pin. Connectors could be added for the sensors, as well as a 3.5-mm jack for audio output.

Of course, this modest project doesn't do full justice to the capabilities of the sensors in the VL53L range. I encourage you to explore the rest of them at [8] and open their datasheets. They offer rich features like configurable interrupts when the measured distance enters or leaves a programmable window. Multi-zone sensors can even recognize gestures, though they require more complex software.

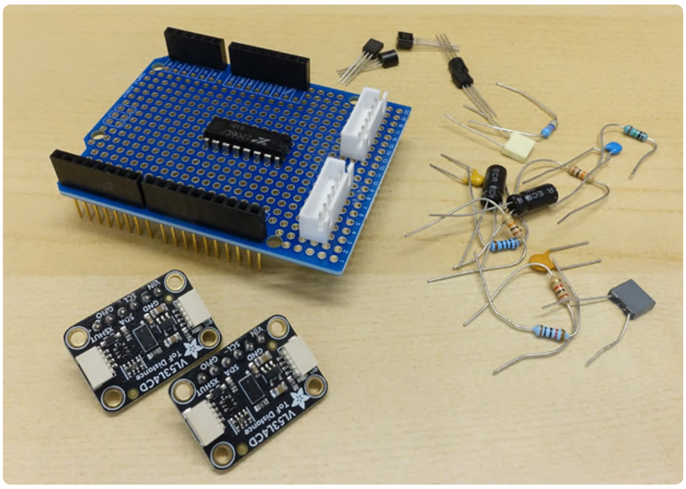


Figure 7: A future, improved version could fit on an Arduino shield like this one.

Specifically for this project, the VL53L4CD's field of view may be a bit too narrow, and it's hard to keep the hand almost perfectly aligned above the sensor. Perhaps you know of a better alternative sensor? I also recommend a video I found while preparing this article: musician Emmanuel Presselin built a much more advanced Theremin using a similar sensor. His version is genuinely pleasant to hear [9] and has many more bells and whistles. And for those interested in analog synths based on XR2206, ICL8038, and others, check out the works of Thomas Henry [10]. Happy soldering!

250476-01

Component List

Resistors (1/4 W)
R1 = 200 Ω
R2, R10, R11 = 1 kΩ
R3 = 4.7 kΩ
R4, R5 = 5.6 kΩ
R6 = 22 kΩ
R7 = 220 kΩ
R8 = 1 MΩ
R9 = 100 Ω
R12 = 10 kΩ
R13 = 100 kΩ

Capacitors
C1 = 220 nF
C2, C3, C4, C6, C7 = 1 µF, electrolytic
C5 = 100 nF

Semiconductors
IC1 = XR2206
Q1, Q2 = BS170

Miscellaneous
MOD1 = Arduino Uno R3, or similar
MOD2, MOD3 = VL53L4CD breakout board, Adafruit 5396 or similar

Questions or Comments?

Do you have questions or comments about this article? Email the author at jean-francois.simon@elektor.com, or contact Elektor at editor@elektor.com.

About the Author

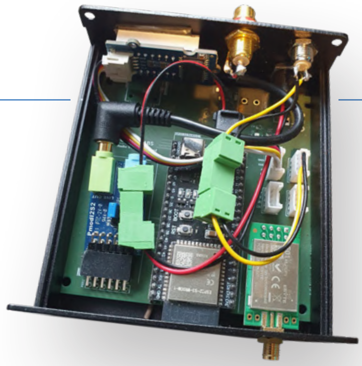
Jean-Francois Simon (Engineer, Elektor) has a longstanding passion for electronics and enjoys topics as varied as circuit design, test and measurement, prototyping, playing with SDRs, and more. He likes to create, modify and improve his tools and other systems. He has an engineering background and also enjoys mechanics, machining, and all things technical. Follow him on X at *x.com/JFS_Elektor*.

Related Products

- **Dogan and Achmed Ibrahim, *Practical Audio DSP Projects with the ESP32* (Elektor, 2023)**
  www.elektor.com/20558
- **Mixer Geek Theremin+ Musical Instrument**
  www.elektor.com/21140

WEB LINKS

[1] Theremin on Wikipedia: https://en.wikipedia.org/wiki/Theremin
[2] XR2206 datasheet, SparkFun: https://cdn.sparkfun.com/assets/8/a/b/3/9/XR2206.pdf
[3] VL53L4CD datasheet, STMicroelectronics: https://www.st.com/resource/en/datasheet/vl53l4cd.pdf
[4] STM32duino library for the VL53L4CD on GitHub: https://github.com/stm32duino/VL53L4CD
[5] Adafruit's breakout board: https://www.adafruit.com/product/5396
[6] ST driver guide: https://tinyurl.com/msjx7nvk
[7] Project's page on Elektor Labs: https://www.elektormagazine.com/labs/a-sound-generator-using-time-of-flight-sensors
[8] ST Time-of-Flight sensors: https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html
[9] Emmanuel Presselin's time-of-flight Theremin (YouTube Video): https://www.youtube.com/watch?v=AIj81NnNKzA
[10] Thomas Henry's DIY synth works, Synth DIY Wiki: https://sdiy.info/wiki/Thomas_Henry



25 ESP32 Audio Transceiver Board (Part 1)

SD Card WAV File Player Demo

By Saad Imtiaz and Jens Nickel (Elektor)

ESP32 controllers are well-like solutions used in thousands of IoT, smart home, and other remote control projects. The ESP32 has also a good I2S interface, which can be used to output and input digital audio signals. In combination with an I2S DAC, an I2S ADC, and an SD card, many kinds of audio projects are possible. In this project, we integrate all this, together with a nRF24 transceiver for transmitting and receiving (audio) data, on a solder-friendly PCB. In the first part of this article series, we present the block diagram, circuit, and demo software.

The popular ESP32 controllers are affordable, fast and easy to program — for example, with the Arduino IDE. Equipped with Wi-Fi, they are used in thousands of IoT, smart home, and other remote control projects, and we can hardly count the Elektor projects of that kind. However, ESP32 controllers have also built-in a good I2S interface, which can output and input audio streams digitally. (Refer to **I2S Interface** text box.) What you additionally need for playing audio is an I2S DAC that takes the I2S data from the ESP32 and transforms it into an analog signal. For the other way round, sampling audio, you need an audio ADC with an I2S interface. For this project, we put an ESP32-S3 based development module together with an I2S ADC/DAC breakout board on a carrier PCB. We also integrated an option to plug on a Nordic nRF24 wireless module to transmit and receive digital (audio) data. An SD card, several extension connectors and different power supply options complete the project.

Use Cases

This board is mainly dedicated for audio applications, but you also can use it to sample/output/store/transmit/receive other signals and data. Here are some use cases, and we will implement some of these in upcoming articles.

- Audio player playing files from SD card (see demo software below)
- Audio recorder
- Audio processor (sampling audio, do something with the data, and outputting it)
- Audio wireless sender/receiver via Wi-Fi or ESP-NOW
- Audio wireless sender/receiver via Nordic nRF24
- Datalogger: sensors connected via SPI/I²C, data are stored on SD card
- Wireless sensor node: transmitting sensor values via Wi-Fi or nRF24
- Generating or measuring LF signals

Main Blocks

You can see the blocks and connectors of the ESP32 Audio Transceiver in **Figure 1**.

I2S Interface

The I2S interface and protocol for transmitting digital audio data (has nothing to do with I²C) is well documented in the Internet [9], so we don't have to waste too many sentences about the basics. Audio data are transferred bitwise on one wire, where another wire is used as the bit clock for this transmission. Another wire is used to separate data for the left audio channel from data for the right audio channel. So, you need at least three wires between the microcontroller and the I2S DAC or ADC (besides VCC and GND to supply these chips).

To ensure that the microcontroller and the DAC or ADC are clocking synchronously, there is another connection foreseen, called "Master Clock". Many sources on the Internet and also my own experience tell that, using an ESP32, the ESP32 should act as the clock master and the DAC or ADC as the slave, and not vice versa. Therefore, the I2S2 Pmod module (hosting a DAC and an ADC) we use in this project must be configured as slave, with a jumper connection at the right place.

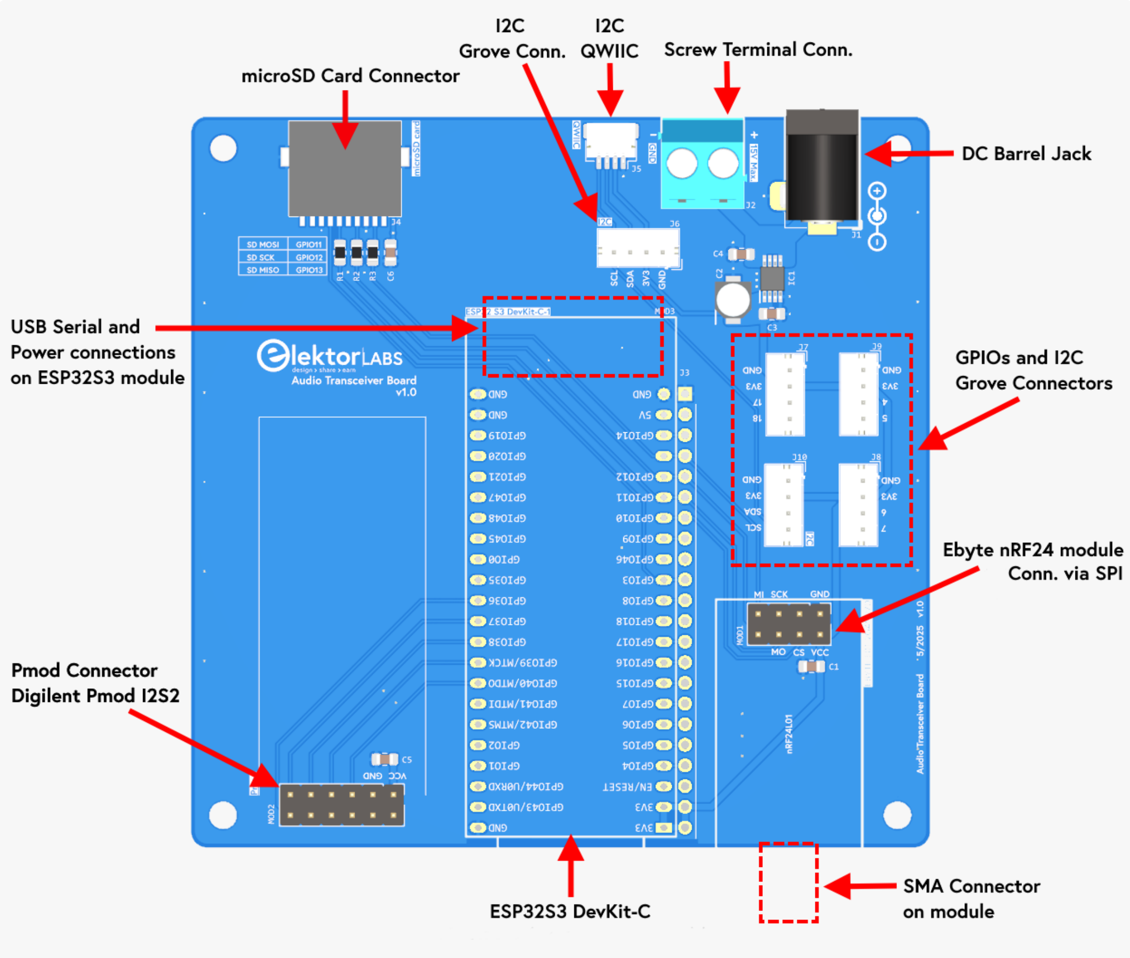


Figure 1: The blocks and connectors of the ESP32 Audio Transceiver.

ESP32-S3-DevKit-C

The brain of the project is the ESP32-S3-DevKit-C Board with 44 pins [1]; it will be plugged in the center of our PCB. Basically, the DevKit-C board integrates an ESP32-S3-WROOM module (with a 2-core ESP32-S3 processor), maker-friendly DIL extension connectors and two USB connectors for programming and power supply (see **Figure 2**). Please note that there are several pin-compatible S3 DevKit-C variants on the market. Some are hosting a WROOM-module with a printed PCB antenna, some are a WROOM-module with a connector for an external antenna. The latter makes it possible to put our carrier board together with the DevKit in a metal housing, and still use the Wi-Fi functions of the ESP32.

As there are S3-DevKit-C-variants with a width of 0.9" and 1", we made our carrier board suitable for both, with an extra pin-header. If you use a 0.9" wide ESP32-S3-DevKit, the outer pin-header (J3) can be also used as an extension connector.

An S3-DevKit-C with PCB antenna is available for about €8; it can be plugged and swapped easily. The only thing you have to solder are two rows of headers. However, we are already thinking on a second version of our PCB, with the S3-WROOM module directly soldered on it, which will save space and even more costs.

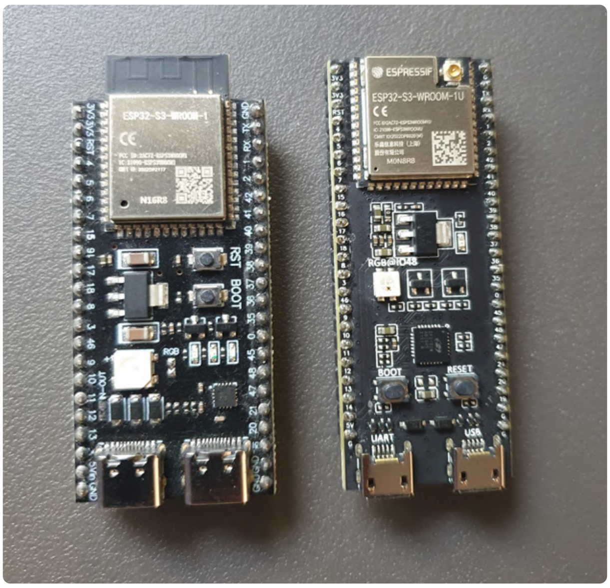


Figure 2: The DevKit-C board integrates an ESP32-S3-WROOM SoC and two USB-connectors for programming and power supply. On the left a version with PCB antenna and 1" width. On the right the version from the Elektor Store, with 0.9" width and connector for an external Wi-Fi antenna.

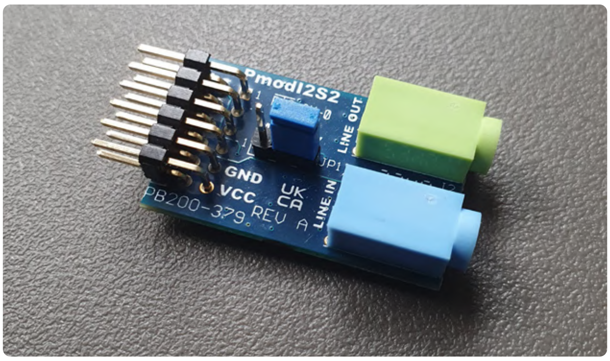


Figure 3: The Digilent I2S2 Pmod module is hosting a Cirrus CS4344 DAC and a Cirrus CS5343 ADC, both connected to stereo audio jacks.

Figure 4: The nRF24 module from Ebyte [4] has additionally to the transceiver chip an SMA antenna connector and an antenna amplifier integrated.

I2S DAC/ADC

There are many I2S breakout boards for sampling and playing audio on the market. At the end, we decided for a Digilent I2S2 Pmod module [2], hosting a Cirrus CS4344 DAC (up to 192 kHz/24 bit) and a Cirrus CS5343 ADC (up to 96 kHz/24 bit), both connected to stereo audio jacks (**Figure 3**). Basically, it is two breakout-boards in one. At the 2x6 connector there are two independent I2S interfaces accessible, one for outputting and one for inputting audio (see textbox I2S interface). The VCC and GND pins for the power supply are also double. It is a bit unusual that this 2x6 connector stands horizontally, but that is part of the Pmod connector specification from Digilent. To plug this module on our PCB, you have to solder an angled 2x6 receptable.

The I2S2 Pmod module is by far not the cheapest I2S BoB you can buy, but the audio quality is quite good and it works excellent with the ESP32 as I2S master. There is also no configuration needed before you can start playing or sampling audio.

On our board, the 2x6 Pmod connector is routed to GPIO35 to GPIO40, plus GPIO47 and GPIO48. Of course, nothing hinders you to connect here other Pmod modules (current sensors, audio amplifiers, ...) or to make your own small extension board with a 2x6 connector in the usual 2.54 mm pitch.

nRF24 Transceiver

Besides the Wi-Fi and ESP-NOW capabilities of the ESP32, we wanted to offer a third option, especially tending to very low latency when sending/receiving a continuous audio stream. Nordic's proprietary wireless protocol for their nRF24 transceiver modules is very fast. You can send packets from sender to receiver in less than 1 ms. The nRF24 chips [3] are also used in many IoT projects, as there are quite cheap breakout boards integrating this chip, and furthermore, there are good (Arduino) libraries. The modules integrating an nRF24, an antenna amplifier and an antenna connector are especially worth the money. Our experiences show that the range inside buildings is better than Wi-Fi. One advantage is that the chips are sending in their own 1-MHz-wide channels and not exactly in the Wi-Fi channels, where you always have a lot of traffic. A little backdraft is the maximum payload size of 32 bytes and the maximum data rate in real applications, which is far below the nominal data rate of 2 Mbit/s. We will come in the next article back to this.

The nRF24 is accessed via an SPI interface plus two digital lines (Chip Enable and Interrupt). There is a kind of standard for the 2x4 connector, so you can use a nRF24 module of your choice. We recommend using the E01-ML01DP5 from Ebyte [4], which has an SMA antenna connector and an antenna amplifier integrated (**Figure 4**). Our carrier PCB is prepared for this module, the antenna connector will sit right at the edge of our board, to fit perfectly in a plastic or metal housing.

As the 2x4 connector on our board is routed to the SPI interface of the ESP32, nothing prevents you to connect here any other SPI modules — for example, a sensor.

SD Card Socket

The board includes a microSD card socket, which provides convenient local storage for logged data, configuration files, or audio content. It connects to the ESP32-S3 via the SPI interface (MOSI, MISO, SCK, and an extra CS line), making it easy to use with the widely supported Arduino *SD* or *SdFat* libraries. The socket is mounted on the top side of the board and oriented for easy access. Decoupling capacitors are placed close to the power pins for stability during write operations. Since the SD card shares the SPI bus with other peripherals like the nRF24 module, proper chip-select handling in software is essential to avoid conflicts.

Extension Connectors

The board features multiple extension connectors (J7 to J10) that expose unused GPIOs from the ESP32-S3, allowing easy connection of additional sensors or other modules. These connectors are suitable for general-purpose use, but also ideal for user interface elements such as displays (e.g., OLED via I²C), push buttons, LEDs, or IR receivers. Each connector includes power and ground pins to simplify wiring.

Power Supply Options

The board can be powered using either a standard DC barrel jack or a 5.08 mm pitch screw terminal, giving flexibility depending on your setup. A TLV76733 linear regulator steps down the input voltage (up to 16 V) to 3.3 V, which powers the ESP32-S3 and connected peripherals. This LDO can supply up to 1 A and offers good precision (±1%), low quiescent current, and protection features like thermal shutdown and current limiting. Its high PSRR (70 dB at 1 kHz) helps keep the system stable even with noisy inputs, and the low dropout voltage ensures efficient operation even when the input voltage gets close to 3.3 V.

Another option to power the board is via one of the USB connectors of the S3-DeKit-C module. It is also possible to power the board via the barrel jack and connect it to the PC via USB at the same time — for example, to send or receive serial data.

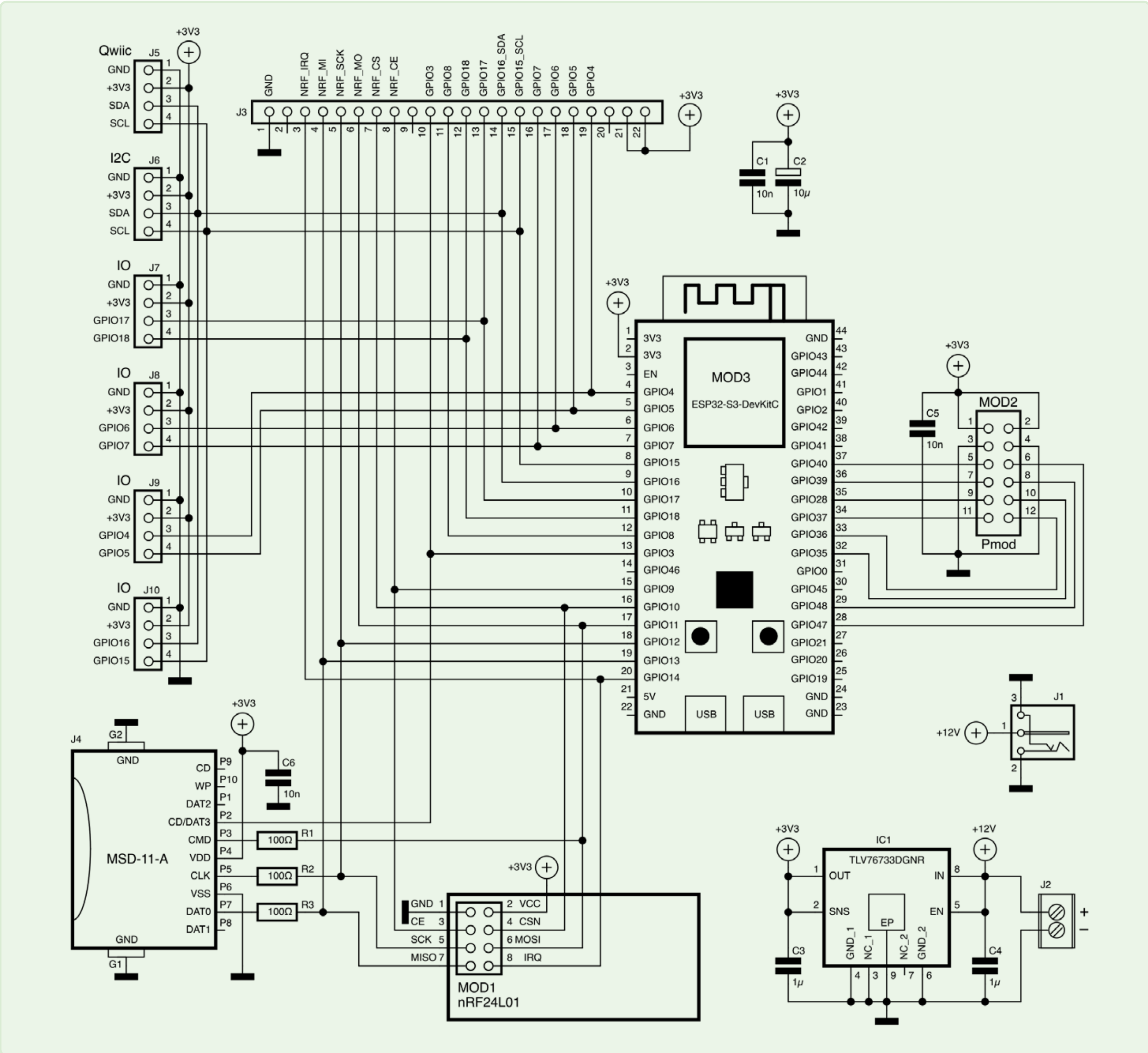


Figure 5: The provided schematic outlines a microcontroller-based system centered around the ESP32-S3-DevKit-C development board.

The Schematic

As we already presented the modules to be connected, the schematic is easy to understand. The circuit diagram (**Figure 5**) outlines a microcontroller-based system centered around the ESP32-S3-DevKit-C development board. Power is supplied via a 12 V input at connector J1, which is regulated down to 3.3 V using a low-dropout voltage regulator (IC1). This regulated 3.3 V line powers the ESP32-S3 board and all other low-voltage components in the circuit, with several decoupling capacitors (C1 to C4) providing noise suppression and stability. The ESP32-S3 serves as the main processing unit and connects to various peripherals through its GPIOs.

Wireless communication is enabled using the nRF24L01 module (MOD1), which interfaces with the ESP32 via SPI lines: MISO, MOSI, SCK, CSN, CE, and IRQ. These connections are protected with 100 Ω resistors (R1 to R3) and a bypass capacitor (C6) for power stabilization. For storage, the design includes an MSD-11-A microSD socket (J4) connected via standard SDIO/SPI lines, with additional lines for card detect and write protect functionality.

The system supports peripheral expansion through multiple GPIO headers (J7 to J10), each providing power, ground, and signal lines to external display modules, buttons, or devices. An I²C interface is exposed through Grove (J6) and Qwiic-compatible (J5) connectors, allowing for straightforward integration of digital sensors and modules. Furthermore, a Pmod connector (MOD2) is included, offering an interface for Digilent-compatible peripheral modules.

The PCB

The PCB layout is organized around the ESP32-S3-DevKitC (see **Figure 6**). The connectors are placed on the top side of the PCB, including the power jack, screw terminal, Qwiic connector, and SD card slot. The Pmod connector is located at the lower side of the board to allow compatibility with various Pmod modules. The nRF24 module is positioned on the right edge, and four connectors (J7 to J10) are placed in the middle-right section to provide additional GPIO access for external modules. I²C tracks are short and narrow, suitable for typical communication speeds, and power traces are wider to handle current distribution, with a ground plane covering most of the board.

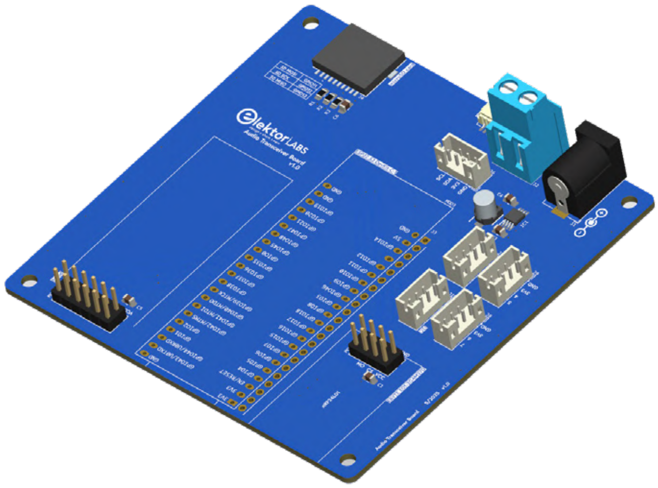


Figure 6: The PCB layout is organized around the ESP32-S3-DevKitC.

Espressif I2S API

The ESP32-S3 has two I2S peripheral blocks, which we can use to output and input audio at the same time. On the PCB we have used eight connections for the two interfaces (besides 2x VCC and GND). In general, you can flexibly choose the GPIOs of the ESP32 for the I2S lines.

For outputting and inputting data via I2S, the DMA controller is used, so the processor core is free for other tasks, when the data are transmitted or received. Of course, a DMA buffer is needed. With the ESP32 I2S API provided by Espressif, you can select the number of buffers (at least two) and the number of music samples in each buffer. More on this in one of the next articles.

The API for controlling the I2S interfaces is of course also subject to development. For our purposes, a legacy version [10] fits well, as it is a bit easier to understand than the latest version of the API. A lot of good Arduino audio libraries hide the API commands from the application programmer, in most cases with some wrapper functions (for example [8]). Although these audio libraries are very powerful and flexible, we thought it would be good that you know how the I2S API works. For convenient use, we grouped all commands to start the ESP32 I2S driver in one function (OK, our own wrapper). Here is the code to start the driver for playing audio, you can find it in the file *Stream_I2S.h*:

```
// these pin definitions are located in BoardPin.h
#define I2S_MCK_OUT 36
#define I2S_WS_OUT  35
#define I2S_SCK_OUT 48
#define I2S_SD_OUT  47

#define I2S_PORT   I2S_NUM_0

uint8_t audiochannelcount = 1;
uint16_t samplerate = 32000;
uint8_t samplebuffer_count = 4;
uint8_t samplebuffer_length = 32;

void I2S_Start()
{
  i2s_channel_fmt_t audiochannelformat =  I2S_CHANNEL_FMT_ONLY_LEFT;

  if (audiochannelcount == 2)
  {
    audiochannelformat = I2S_CHANNEL_FMT_RIGHT_LEFT;
  }
```

```
  i2s_comm_format_t communicationformat =  i2s_comm_format_t(I2S_COMM_FORMAT_I2S |
                                                              I2S_COMM_FORMAT_I2S_MSB);

  i2s_config_t i2s_config_out = {
  .mode = i2s_mode_t(I2S_MODE_MASTER | I2S_MODE_TX),
  .sample_rate = samplerate,
  .bits_per_sample = i2s_bits_per_sample_t(16),
  .channel_format = audiochannelformat,
  .communication_format = communicationformat,
  .intr_alloc_flags = 0,
  .dma_buf_count = samplebuffer_count,
  .dma_buf_len = samplebuffer_length,
  .use_apll = true,
  .tx_desc_auto_clear = true, // silence on underflow
  .fixed_mclk = samplerate * 256
  };

  i2s_driver_install(I2S_PORT, &i2s_config_out, 0, NULL);

  const i2s_pin_config_t pin_config_out = {
    .mck_io_num = I2S_MCK_OUT,
    .bck_io_num = I2S_SCK_OUT,
    .ws_io_num = I2S_WS_OUT,
    .data_out_num = I2S_SD_OUT,
    .data_in_num = -1
  };

  i2s_set_pin(I2S_PORT, &pin_config_out);
  i2s_start(I2S_PORT);
}
```

You don't have to understand everything in detail for now, but you can see that it is not difficult to change for example the sample-rate or the GPIO pins used for the I2S interface.

Here is the API driver command to send 128 bytes of data via the I2S interface to the DAC:

```
i2s_write(I2S_PORT, &wBuffer, 128, &bytesWritten, portMAX_DELAY);
```

With this, you'll send 128 bytes of audio to the DAC, which must be located in `wBuffer`. You can fill it with audio data from the SD card, a generated sine or rectangle audio signal or any other sources. `wBuffer` can be set up as an array of simple bytes (`uint8_t wBuffer[128];`) or an array of music samples of 16 bit width, in that case you have to use signed integers (`int16_t wBuffer[64];`). The `i2s_write` will work with both sorts of buffers, but the total number of bytes must be 128, as stated in the third parameter of the function.

A quite convenient feature is that `i2s_write` with `portMAX_DELAY` as fifth parameter is blocking, until space is available in the DMA buffers. After filling the buffer again, the function returns, and you can do other things, while the data are output without needing the processor. You just have to make sure that you do not waste too much time until calling `i2s_write` again with the next data packet, before the buffers run empty. That makes it very simple to output data continuously!

For the other way round — sampling data via the ADC — there is a convenient to use `i2s_read` function:

```
esp_err_t result = i2s_read(I2S_PORT, &rBuffer, 128, &bytesIn, portMAX_DELAY);
```

We will use this function in a later article.

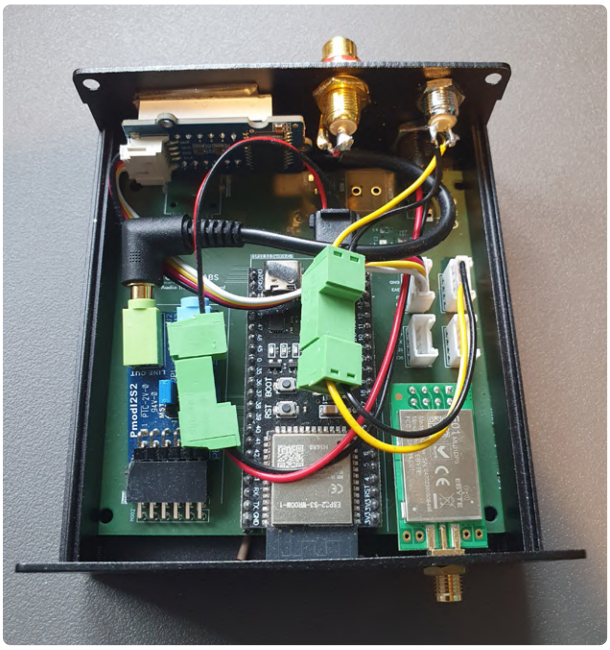


Figure 7: The SMA connector on the nRF24 module is positioned so that a hole can be drilled in the faceplate to mount the antenna externally — this way, the module remains usable even inside the metal housing.

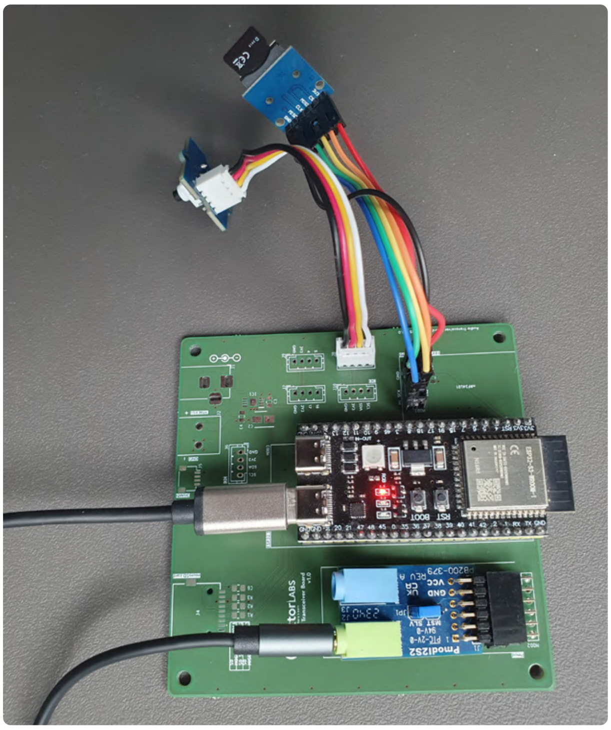


Figure 8: The WAV player prototype with a button to jump to the next song. For the prototype, we did not use the SD card socket on the PCB, but an extra-small SD card module, which is connected via SPI to the ESP32.

The Housing

The PCB fits into a 100 × 97 × 40 mm aluminum enclosure with a sliding lid [5]. It mounts directly into the enclosure rails, and the lid makes it easy to access the internals when needed. Metal enclosures are useful for mechanical protection and shielding, but they also block high-frequency signals like Wi-Fi and Bluetooth. The metal acts as a Faraday cage, so wireless communication with the PCB antenna version of the ESP32 DevKit won't work — unless you provide an opening. For this reason, the ESP32-S3 antenna area is placed at the bottom edge of the board, allowing a small cutout to be made in the enclosure for signal access. If you use an ESP32-S3-DevKit with an antenna connector — for example, the one from the Elektor store — you can mount an external antenna at the front or rear plate of the metal housing.

The SMA connector on the nRF24 module is positioned so that a hole can be drilled in the faceplate to mount the antenna externally (see **Figure 7**).

Demo Software

As said, in the next editions, we will present some applications and further developments. For example, in upcoming "Wireless" edition of *Elektor* in September 2025, we will present software for transmitting and receiving audio. But we did not want to end this article without some first demo software.

The demo software is a simple WAV audio file player. You can download it from the Labs page for the Audio Transceiver Board [6]. At startup, the ESP32 reads the files that are stored on an SD card (no subdirectories) and puts all the names of the WAV files in a small list. After that, it plays the first audio file from that list. With a press of a button, which is connected to GPIO7 of the ESP32 (at J8), the current track is stopped and the next track comes on turn. **Figure 8** shows a prototype with the button. For the prototype, we did not use the SD card socket on the PCB; instead, it is an extra small SD card module, which is connected via SPI to the ESP32. As a connector for the SPI signals, we used the 2x4 "nRF24 socket."

If you connect the ESP32 via USB to a PC, you can send commands from a terminal program. A "b" lets the software jump to the next track; an "a" to the previous one. By the way, as our software cannot play MP3 tracks (yet), you can use a free converter like [7] to convert your favorite MP3 song(s) to a WAV file.

The demo code is modular. Maybe the most interesting software module is *Stream_I2S.h*, which handles writing the music bytes to the I2S interface. (See the **ESP32 I2S API** text box.) We intentionally did not use one of the powerful big audio libraries — for example, the *Arduino Audio Tools* by Phil Schatzmann [8]. They can be flexibly used for many hardware platforms, they have a tremendous amount of features, and they are very well documented. On the other hand, they hide the raw music data a bit from the user, and we wanted to show how easy it is to work with the bytes representing the audio signal.

Besides our simple *Stream_I2S* library, there is a small library *SDCardFiles.h*, which is based directly on the Arduino standard libraries for SD card access. A very small *Button.h* lib handles the button presses.

The most important loop of the main program is:

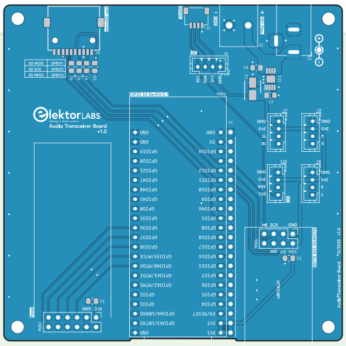
```
uint8_t bBuffer[128];
...

while (SDCardFiles_DataLeftToRead())
{
      SDCardFiles_ReadInBuffer(bBuffer, 128);
      I2S_WriteFromBuffer(bBuffer, 128);

...

}
```

Component List



Resistors
R1, R2, R3 = 100 Ω

Capacitors
C1, C5, C6 = 10 nF
C2 = 10 µF
C3, C4 = 1 µF

Semiconductors
IC1 = TLV76733DGNR (SOP65P490X110-9N)

Modules and Connectors
J1 = Barrel Jack
J2 = TB010-508-02BE terminal block
J3 = Pinheader 1x22, 2.54 mm pitch
J4 = MSD-11-A (MSD11A) SD card socket
J5 = Qwiic Connector
J6 = Grove Connector
J7, J8, J9, J10 = Grove Connector
MOD1 = nRF24L01 + Vertical 2.54 mm (2x4) connector
MOD2 = Pmod I2S2 + Right Angle 2.54 mm (2x6) connector
MOD3 = ESP32-S3-DevKit-C + 2x pinheader 1x22, 2.54 mm pitch

First, we read 128 bytes from the SD card; after that, we write these 128 bytes to the I2S interface of the ESP32. After that, we have enough time to check if a button was pressed or a character was received via the serial interface.

Please consider this software just as a demonstration of how easy it is to work with audio data. Of course, a real player should be completed with a display and better input elements like a rotary encoder.

In the next part of this series, we will try to send and receive audio data wirelessly. As an appetizer, refer to Figure 7, which shows a prototype of the wireless receiver. Stay tuned!

250384-01

Questions or Comments?

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

About Saad Imtiaz

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.

About Jens Nickel

Jens Nickel studied physics and was further educated as an editor for professional tech/science magazines. In 2005, he started working for Elektor, and for the past several years, he has served as the Editor in Chief of the magazine. Jens has liked programming since the early days of the C64. No wonder that he is also fascinated by the flexible and powerful options of good microcontroller platforms. Over the last few years, the ESP32 has become his favorite controller for personal projects. Jens's hobbies include music videos, digital audio and studio control projects.

WEB LINKS

[1] ESP32-S3-DevKitC-1, Espressif: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32s3/esp32-s3-devkitc-1/index.html
[2] Pmod I2S2: Stereo Audio Input and Output, Digilent: https://digilent.com/shop/pmod-i2s2-stereo-audio-input-and-output/
[3] nRF24 Series, Nordic: https://www.nordicsemi.com/Products/nRF24-series
[4] E01-ML01DP5, nRF24L01 module by Ebyte: https://www.cdebyte.com/products/E01-ML01DP5
[5] Empty Housing Aluminium - 100 x 97 x 40 mm, Amazon: https://www.amazon.de/dp/B098TNTGFP?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1
[6] Software-Download: https://www.elektormagazine.com/labs/esp32-audio-and-iot-board
[7] WAV Converter, Restream.io: https://restream.io/tools/wav-converter
[8] Arduino Audio Tools, GitHub: https://github.com/pschatzmann/arduino-audio-tools
[9] I2S, Wikipedia: https://en.wikipedia.org/wiki/I%C2%B2S
[10] API Reference I2S (V4.2), Espressif : https://docs.espressif.com/projects/esp-idf/en/v4.2/esp32/api-reference/peripherals/i2s.html

Elektor infographic

Circuits and Circuit Design

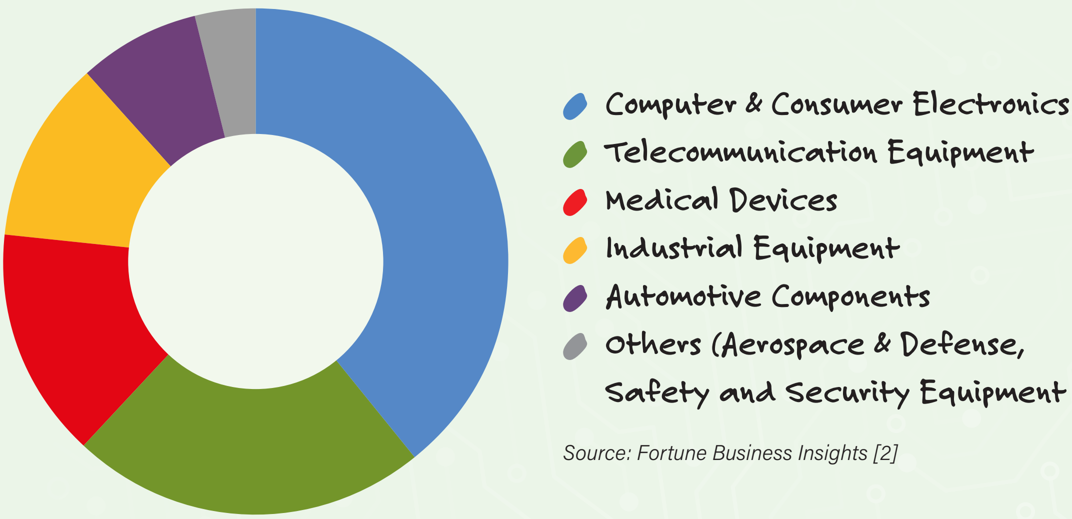
PCB Design Market

Leading Companies

- Cadence
- Siemens
- Zuken
- Altium
- WestDev (Pulsonix)
- Novarm Limited (DipTrace)
- Autodesk
- Synopsys
- KiCad

Source: Research Nester [1]

Global PCB Design Software Market Share



Source: Fortune Business Insights [2]

A Greener Future for PCBs

Key trends in sustainable electronics show growing environmental concerns and a strong push for new ideas. As energy and water use in the semiconductor industry are expected to grow at 12% and 8% compound annual growth rate (CAGR) between 2025 and 2035, managing these resources well is becoming more important [3]. With integrated circuits (ICs) being the third most traded product worldwide, there's a huge chance to make electronics more eco-friendly.

The printed circuit board (PCB) sector stands out in this shift, with PCB and contract manufacturers expecting the most growth in sustainability efforts by 2025, according to IPC International [4]. Being environmentally responsible is now a focus, a key to change and new ways of working. Even with challenges like tight budgets and complex rules, 95% of those surveyed expressed confidence in reaching their sustainability goals, with PCB makers among the most optimistic subgroups.

Semiconductor manufacturing energy usage forecast to reach

736 billion kWh

by **2035**
at a **CAGR** of
12%

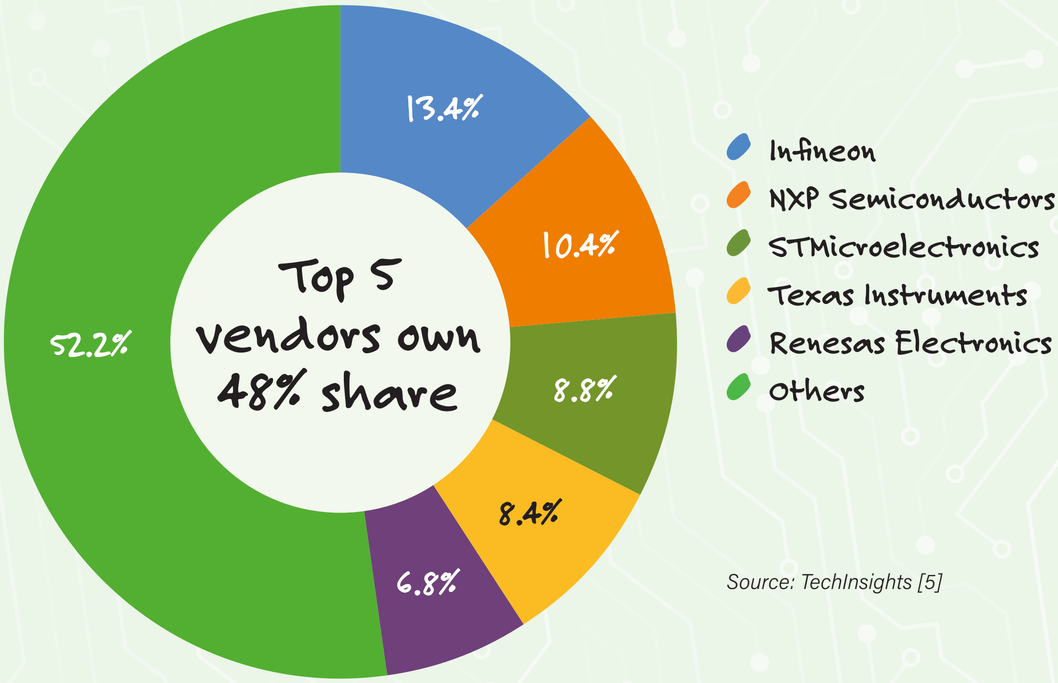
Semicondutor Water Usage

Source: IDTechEx [3]

Top Automotive Semiconductor Suppliers in 2024

Infineon Technologies has claimed the top spot among the world's leading automotive semiconductor vendors in 2024, according to the latest report by TechInsights [5]. NXP Semiconductors follows in second place, with STMicroelectronics in third. Texas Instruments ranks fourth, while Renesas Electronics rounds out the top five. The rest of the top ten includes onsemi, Micron, Qualcomm, Bosch, and Analog Devices (ADI).

Regionally, Infineon secured the top position in several key markets: it led in Europe with a 14.1% market share, in China with 13.9%, and in South Korea with 17.7%. In North America, Infineon rose to second place with a 10.4% share, just behind NXP Semiconductors. In Japan, it maintained a strong second place with 13.2%, following Renesas, which remains the market leader there.



Source: TechInsights [5]

250253-01

WEB LINKS

[1] Research Nester, "PCB Design Software Market," March 2025: https://www.researchnester.com/reports/pcb-design-software-market-size/92

[2] Fortune Business Insights, "PCB Design Software Market," March 2025: https://www.fortunebusinessinsights.com/pcb-design-software-market-107433

[3] IDTechEx, "Sustainable Electronics and Semiconductor Manufacturing 2025-2035," Jan 2025: https://www.idtechex.com/en/research-report/sustainable-electronics-and-semiconductor-manufacturing-2025/1065

[4] Nick Flaherty, "IPC sees more sustainability in board manufacturing," eeNews Europe, March 2025: https://www.eenewseurope.com/en/ipc-sees-more-sustainability-in-board-manufacturing/

[5] TechInsights, "2024 Automotive Semiconductor Vendor Share," April 2025: https://library.techinsights.com/public/sectioned-blog-viewer/2f3647d4-f374-48c0-a658-368d96066439

26 Small Audio Mixer

A Simple and Versatile Scalable Design

By Thierry Clinquart (Belgium)

Audio mixers are commercially available, but this design allows for a scalable system by splitting the different stages. The single modules can also be used for other projects.

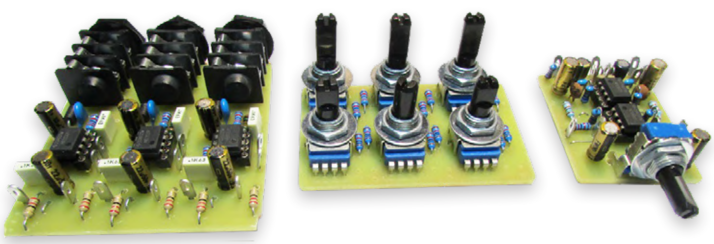


Figure 1: This latest version of the Audio Mixer is made of three boards: Input (left), Level & Pan (middle) and Mix & Master (right).

The first version [1] of this project was made with Alps RK09 series potentiometers, and consisted of only two PCBs. The first board housed the level and pan pot controls. The second handled the mix and master volume. This former version of the project, with one mic input and two line inputs, was used to monitor my field mixer for a Smartphone DI Box project [2].

The second version was implemented using Bourns PTV111 series potentiometers and adapted following the feedback from a Labs reader concerning the negative influence of the pan pot on the level control. By inserting a buffered op amp between these two potentiometers, the problem was solved. So, instead of using a single NE5534 opamp, I added an input preamp stage, switching to the dual, low-noise operational amplifier NE5532 by Texas Instruments.

This new version of the Small Audio Mixer consists of three boards, as shown in **Figure 1**:

- Input
- Level & Pan
- Mix & Master

Component List

Resistors
(1/4 W, 1 %)
RSUM = 22 kΩ (L&R)
RIN = 22 kΩ
RCR = 22 kΩ
RFT = 10 kΩ
R1, R2 = 10 kΩ
R3, R4 = 22 kΩ
R5, R6 = 33 kΩ
R7, R8 = 15 kΩ
R9, R10 = 47 kΩ
R11, R12 = 22 kΩ
R13, R14 = 600 Ω
R = 22 Ω (*)

Potentiometers
(Bourns, PTV Series)
P1 = 10 kΩ, Log (PTV111-3415A-A103)
P2 = 10 kΩ, Lin (PTV111-3220A-B103)
P3 = 10 kΩ, Log (PTV112-1417A-A103)

Capacitors
CHF = 100 pF, X7R
CIN = 22 µF, 25 V, Radial
COUT = 22 µF, 25 V, Radial
C1...C4 = 22 µF, 25 V, Radial
C5, C6 = 22 pF, ceramic
C = 100 µF, 25 V, Radial (*)
C = 100 nF, 50 V, Polyester (*)

Inductors (Axial)
FB = 10 µH or VK200

Semiconductors
OP1...OP3 = NE5532P, Texas Instruments

Connectors
Input stage: 6.35 mm Jack, F, Neutrik NRJ6HF
Aux Input: 3.5 mm Stereo Jack, F, Cliff FC681374V
L&R Output: 3.5 mm Stereo Jack, F, Cliff FC681374V

(*) On power rails, see schematic diagram.

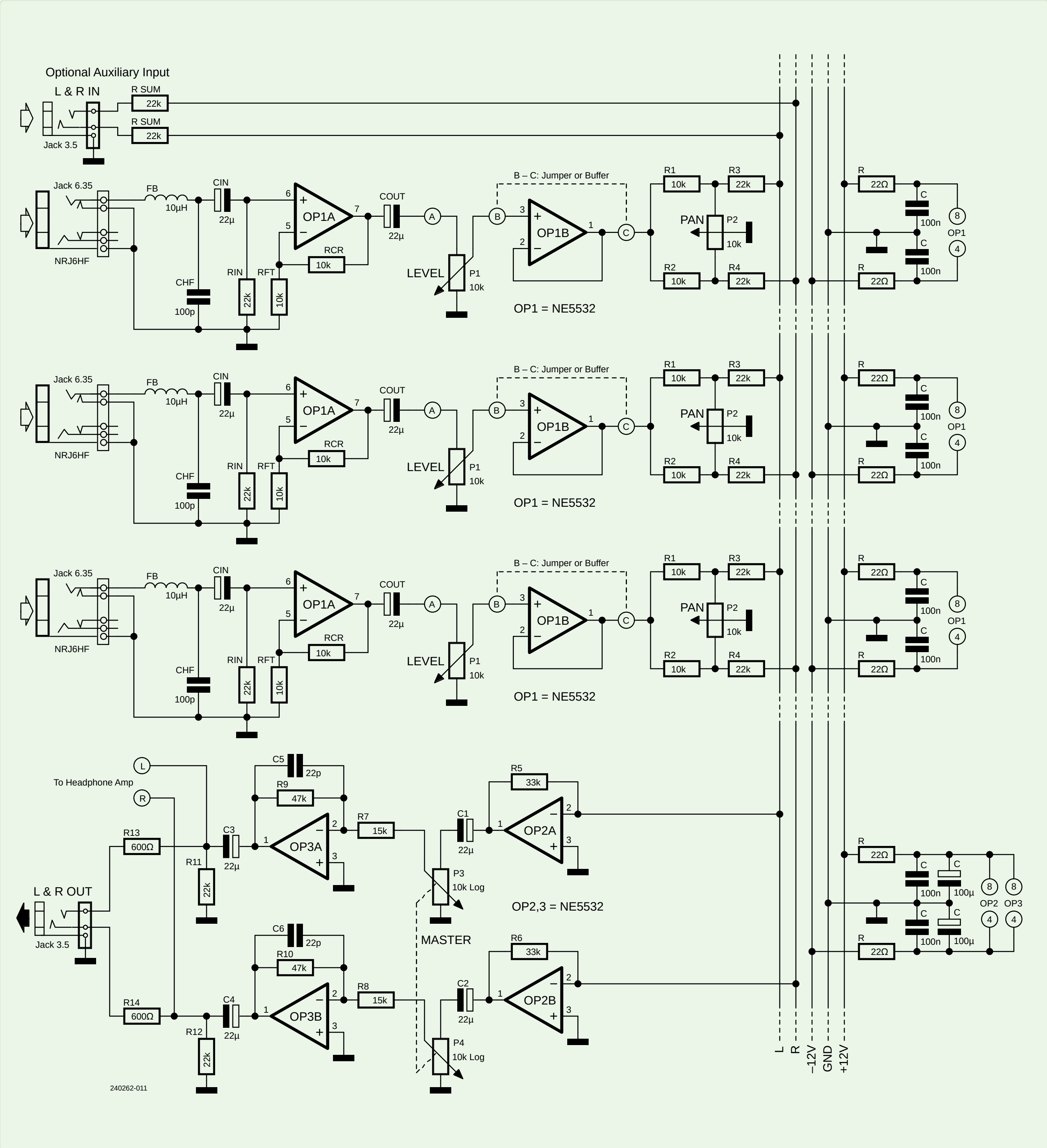


Figure 2: Circuit diagram of the Audio Mixer.

Circuit Diagram

The circuit diagram is depicted in **Figure 2**. Although I started the preamp with an NJ5FD-V jack, I quickly switched to the NRJ6HF model due to size, availability, and price. I used the TN pin so that in the absence of a signal, the input is grounded. I placed a small LC input filter consisting of a ferrite bead (or VK200) and a 100 pF X7R capacitor. C_{IN} (22 µF) cuts off any DC component. The silkscreen of the Input/Preamp PCB is shown in **Figure 3**.

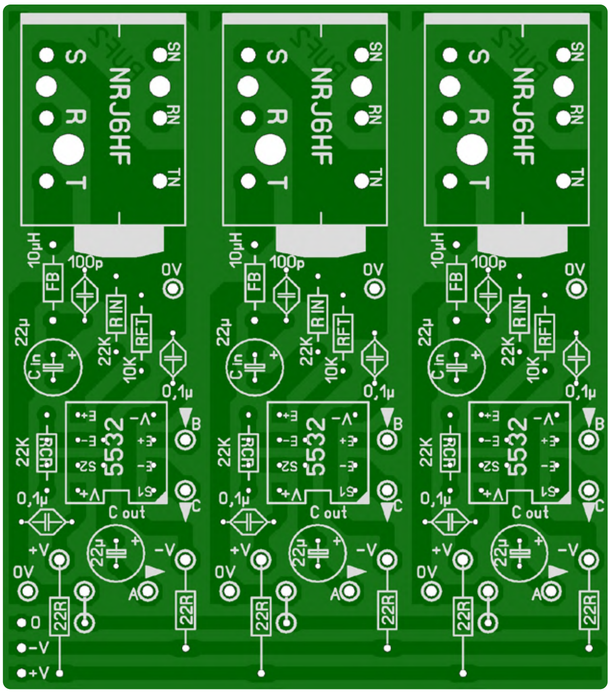


Figure 3: Silkscreen of the Input PCB. The unused inputs are grounded through the TN pins of NRJ6HF jack connectors.

R_{IN} (22 kΩ) sets the input impedance. You can adapt its value, according to your sources. The gain of OP1A is calculated as 20 log (1 + R_{CR}/R_{FT}) or 10 dB. For example, you can have a gain of 30 dB with R_{FT} = 1 kΩ and R_{CR} = 47 kΩ to build an instrument preamp. In this case, R_{IN} can go up to 100 kΩ or more (see instrument characteristics) and C_{IN} can go down to 1 µF.

The output of the preamplifier OP1A goes to the *LEVEL* potentiometer P1 (A) located on the Level & Pan board, shown in **Figure 4**. From the cursor of P1 (B), the signal is fed to OP1B, that works as a unity-gain buffer, to avoid any influence on the level resulting from the adjustment of *PAN* pot. But this buffer remains an option, which can be replaced by a jumper (B-C) under the potentiometer PCB.

R1 to R4 and P2 constitute the pan potentiometer's circuitry. When the cursor is on the right, R1 is grounded, thus putting the left channel at 0 V potential via R3. When the cursor is on the left, R2 is grounded, thus putting the right channel at 0 V potential via R4. In the mid-position, there is a balance of the left and right signals. In this central position of the *PAN* pot, the signal is attenuated by 10 dB since R1 (10 kΩ) and half of the potentiometer (5 kΩ) constitutes an attenuator whose formula is: 20 log (10 k + 5 k) / 5 k = 20 log 3. The same applies to the calculation with R3. No problem because the preamp provides a gain of 10 dB and the pan pot a loss of 10 dB: we thus connect to the L&R bus at 0 dB.

Each input module is decoupled by 100 nF capacitors. The 22 Ω R resistors can be used as a shunt to measure the current in each branch of the operational amplifier and cut off in the event of a short circuit.

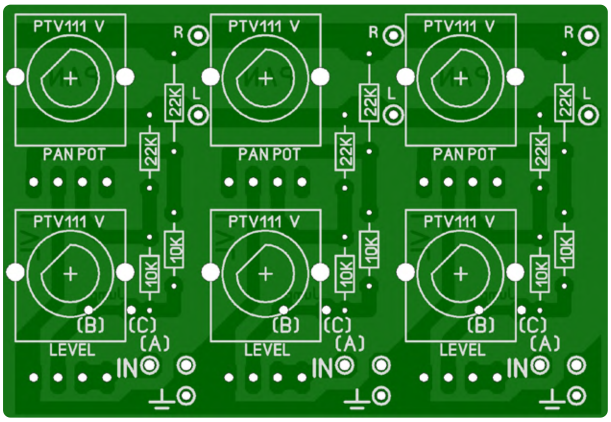


Figure 4: The Level & Pan board of the Audio Mixer, designed to host Bourns PTV111 series potentiometers.

The input module can be reproduced N-times the number of inputs you want. In my case, the spaces are 20.32 mm between channels but also between *LEVEL* and *PAN* potentiometers. Strange? No, we're in inch steps (8×2.54 mm). I always make my drilling plan with my PCB software to be perfectly correlated.

Final Stage

Going forward: OP2A and OP2B are mounted in an inverting-summing configuration, whose gain is calculated: 20 log R5/R3, as well as 20 log R6/R4. The signal is then handled by the Mix & Master board (**Figure 5**) with the stereo *MASTER* potentiometer followed by the final

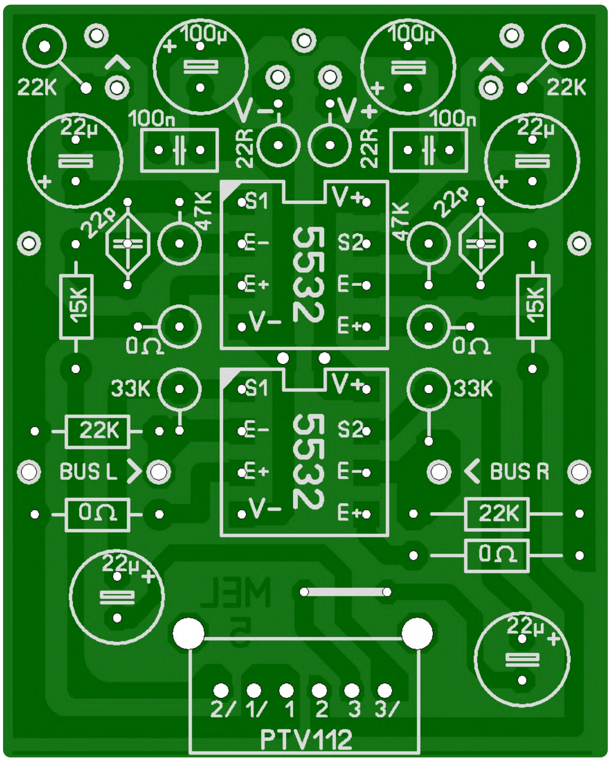


Figure 5: The Mix & Master final stage of this design.

stage, which provides a gain of 10 dB (20 log R9/R7 and 20 log R10/R8). This allows to have an output level of 0 dB with the Master control at 70% of its travel and to push the output level, if necessary, in case of attenuated input source signals.

At the output, a headphone amp or other modules can be connected. I also provided a small 3.5-mm jack (L & R OUT) which can be routed to the new generation of a Porta-Recorder. You will notice R13 and R14, two 600 Ω resistors in series with this connector. Their purpose is to avoid overloading the NE5532 in the event of an erroneous connection of a headphone (often 32 Ω or lower, nowadays). These resistors will have little or no effect on the signal of a 10 to 22 kΩ input of a downstream device. RCA cinch connectors can also be considered for this output. R11 and R12 are pull-down output resistors, whose purpose is to avoid any possible voltage build-up at the DC-separating capacitors C3 and C4.

The three terminals A, B, C of the preamp module will be wired with their corresponding terminals on the Level & Pan board. Three boards that can be judiciously placed in a metal housing, preferably. In my case, I've used the Hammond 1590DFL (188×120×56 mm).

Splitting the Preamp, Level & Pan, Mix & Master functions allows me to evolve in my own creations. Increasing the number of inputs is therefore not a problem. I also offer an optional stereo auxiliary input on a 3.5-mm jack. Two summing resistors (22 kΩ) connect the L&R bus. It's easy to wire directly to this connector. Very practical for adding sources such as a portable radio, PC, tablet, etc. The volume can be adjusted by these devices themselves.

Other Considerations

The phase of the output signal is identical to that of the inputs because in the preamplifier stage we have non-inverting opamps and in the summing, we have two inverting circuits that follow each other. The NE5532 is a *must* in audio, but there are a host of compatible opamps that could be used for this project. I often test my circuits with TL072s and, if it's OK, I switch to the NE5532. Using these schematics, you can implement your favorite potentiometers. You can learn more in books like *National Semiconductors Audio Handbook* or *Small Signal Audio Design* by Douglas Self, shown in the **Related Product** frame in this article.

Let your imagination run wild for your listening pleasure!

240262-01

Questions or Comments?

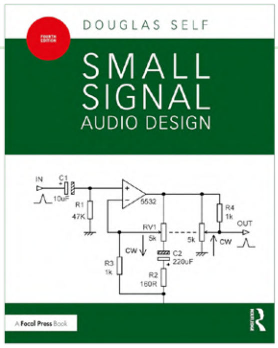
Do you have technical questions or comments about this article? Please contact the editorial team of Elektor at editor@elektor.com.

About the Author

As an electronics technician, Thierry Clinquart discovered the famous µA741 operational amplifier in 1980 during his studies at the Don Bosco Institute in Tournai (Belgium). At the time, it was so much easier to create an audio setup than with transistor technology. He has followed the evolution of this ancestor through the TL071, NE5534 up to the present day with "audio grade" products from Texas Instruments, Analog Devices, JRC, THAT Corp, etc. All the circuits Thierry presents on Elektor Labs are linked together to create customized modules. To reduce wiring, he fits Neutrik A-series connectors directly on PCBs, using Sprint Layout software to optimize their design and to ensure a consistent packaging.

Related Product

- **Douglas Self, *Small Signal Audio Design* (4th Edition), Focal Press**
  www.elektor.com/18046

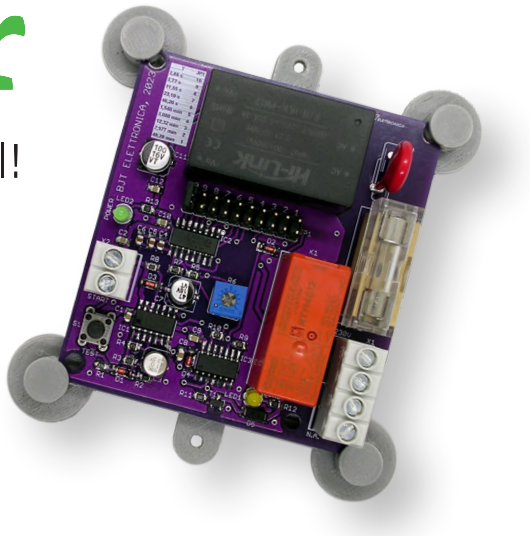


WEB LINKS

[1] Elektor Labs page for both versions of the mixer: https://tinyurl.com/ukt96ev8
[2] Elektor Labs page for the smartphone DI Box project: https://tinyurl.com/n7y4uynn

27 Smart Staircase Light Timer

Save More Money on the Energy Bill!



By Giuseppe La Rosa (Italy)

Need a light timer for staircases? This design, made simply with logic gate ICs, has unique features. In addition to the standard timer function, it allows the activation cycle to be extended and also terminated instantly. The timer is programmable in a wide range of timings that span from 0.2 s to about 50 minutes.

This timer, which uses only logic gates, can be programmed for times ranging from a few seconds to tenth of minutes by simply moving a jumper on the board. A trimmer is used to precisely adjust the base time unit (T_{base}) we want to get. As anticipated, the advantage of this timer, compared to others, consists of two extra features.

As with all normal timers, as soon as the button is pressed, the relay is energized and remains on for the set time. If, after at least 5 s, the button is pressed again, the timer sequence is restarted, thus extending the initial On period by the set T_{base}. Conversely, if after at least 5 s the button is pressed twice, the relay de-energizes instantly.

Circuit Diagram

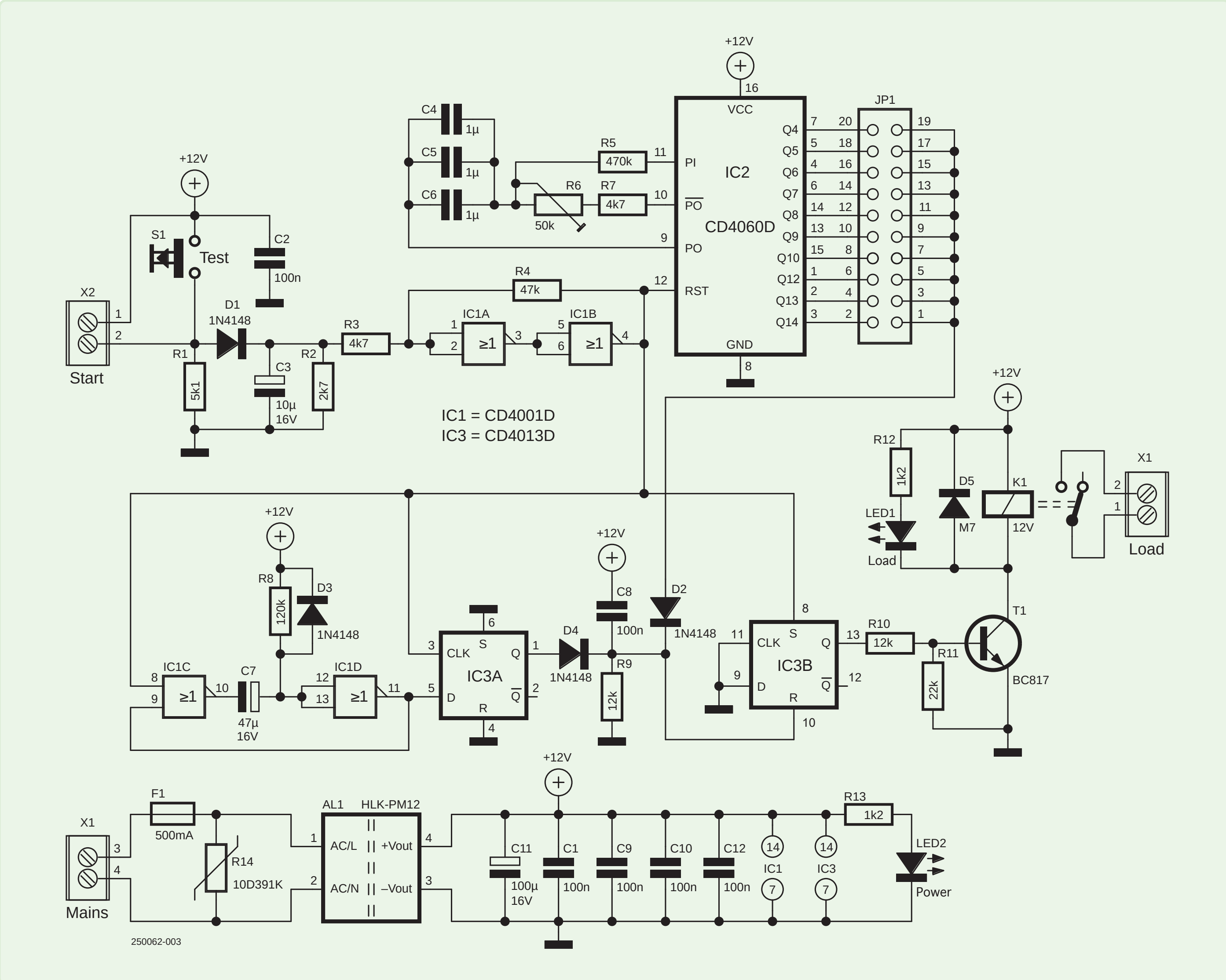
The circuit diagram is shown in **Figure 1**. Each time the *Test* push-button S1 (or one of the buttons connected in parallel to the screw terminal X2) is pressed, the capacitor C3 is charged instantly through diode D1. This voltage reaches pins 1 and 2 of IC1A through R3, bringing them to a logic level 1. IC1A and IC1B — part of a CD4001D quad NOR by Texas Instruments — are wired as a Schmitt trigger, whose output pin 4 (IC1B) goes to logic level 1, consequently. Resistor R4 gives a bit of hysteresis to the response of this two-gates network. When button S1 is released, C3 discharges through R2, allowing IC1A and IC1B network to return to a logic level 0 on the same output pin 4.

The main purpose of the electrolytic capacitor C3, downstream of diode D1, is to remove any spurious pulses potentially generated by the contacts of the S1 push-button. The logic level 1 present on output pin 4 of IC1B, triggers the reset pin 12 of IC2 (a CD4060 CMOS, a 14-Stage Ripple-Carry Binary Counter/Divider and Oscillator by Philips), bringing all outputs Q4...Q14 (corresponding to pins 7, 5, 4, 6, 14, 13, 15, 1, 2, 3, respectively) to logic level 0.

The same logic signal heading to pin 12 of IC2 also reaches pin 8 (set) of the Set-Reset Flip/Flop IC3B; as a result, its output pin 13 toggles to logic level 1, reaching (through R10) the base of transistor T1 and energizing K1 relay through its collector.

The diagram shows that this logic level 1 also reaches pin 8 of the NOR gate IC1C and pin 3 (CLK) of the D-type Flip/Flop IC3A. When pin 8 of the delay network — made with the two NOR gates IC1C and IC1D — switches to logic level 1, the same happens on output pin 11 in consequence; after 5 s (time determined by C7), it reverts to logic level 0.

In this circuit, the IC3A and IC3B Flip-Flops are used differently. When a leading edge (from 0 to 1) reaches pin 3 (CLK) of the D-type Flip-Flop IC3A, it transfers the logic level on input pin 5 (D) to pin 1. If a trailing edge (from 1 to 0) reaches pin 3 (CLK), the logic level at the output of IC3A does not change.



▲

Figure 1: Circuit diagram of the Smart Staircase Light Timer.

So, as soon as we press S1 and a rising edge pulse reaches pin 3 (CLK), since the input pin 5 (D) is at logic level 0, the Flip-Flop transfers this level 0 to output pin 1. When we release S1, since a trailing edge pulse arrives at pin 3 (CLK), its output pin 1 stays at logic level 0.

If, after 5 seconds, we press button S1 a second time, the monostable multivibrator IC1C and IC1D will already have set pin 5 (D) to logic level 0, so pin 1 will stay at logic level 0 and all IC2's output pins will be reset — i.e. kept at logic level 0. This will extend the relay's excitation time.

If, after 5 seconds, we press S1 twice, the monostable network IC1C and IC1D will set pin 5 (D) of IC3A to logic level 1, and the second press of S1 with a rising edge at CLK will set the output Q (pin 1) to level 1. So through D4, pin 10 (R) of the Flip-Flop IC3B will, by resetting, de-energize the relay.

Setting the Time

Having said that, let's see how the timer trigger times should be set. As can be seen from the schematic in Figure 1, a trimmer (R6) and some capacitors (C4... C6) are connected to pins 9 and 10 of the oscillator/divider stage IC2. Knowing the value of R6 + R7 and C = (C4 + C5 + C6), we can calculate the base time T_{base} (in s) using the following formula:

$$T_{base}\,[s] = (R6 + R7) \times C \times 0.0022$$

Here the values of the resistors must be expressed in kΩ and that of the overall capacitance C in µF.

From this formula, we will obtain the base time T_{base} for a single cycle generated by IC2. With simple calculation, it turns out that T_{base} will span from 0.031 s to 0.361 s, depending on the adjustment of R6.

Table 1: Jumper Presets and Delay Ranges.

IC2 Pin N°	PCB JP1 Silkscreen Pin N°	T_{base} Multiplication Factor	T_{min} (s) with R6 = 0 Ω	T_{max} (s) with R6 = 50 kΩ
7 (Q4)	10	× 8	0.248	2.888
5 (Q5)	9	× 16	0.496	5.776
4 (Q6)	8	× 32	0.992	11.552
6 (Q7)	7	× 64	1.985	23.105
14 (Q8)	6	× 128	3.970	46.210
13 (Q9)	5	× 256	7.941	92.421
15 (Q10)	4	× 512	15.882	184.842
1 (Q12)	3	× 2048	63.528	739.368
2 (Q13)	2	× 4096	127.057	1,487.737
3 (Q14)	1	× 8192	254.115	2,957.475

The diagram of the internal oscillator of IC2 can be found at the bottom of page 4 of the relevant datasheet available at [1]. This circuit generates a clock based on the time constant of the RC network — consisting of C and R1 — between pins 9 and 10, and is also controlled by the NAND gate via pin 11 and pin 12 (RST), which inhibits its operation when a logic level 1 is applied to it.

Keep in mind that the clock generated by the oscillator stage will raise the pins of IC2 (connected to JP1) to logic level 1 after a number of cycles depending on the IC dividing ratio, which varies from 8 to 8,192. The bridged pins at JP1 determine which IC dividing ratio will be used for the delay to switch off the load. **Table 1** illustrates all possible delays with their respective T_{base} multiplication factors.

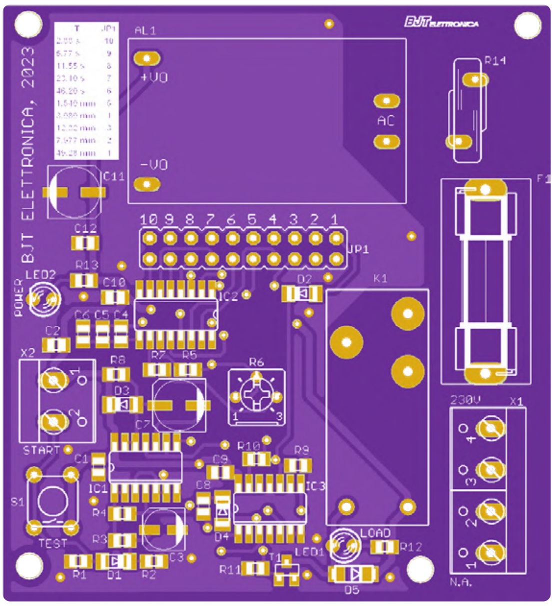


Figure 2: The silkscreen of the PCB, indicating the correct placement for all SMD components onboard.

Component List

Resistors

(All 0805, unless diff. noted)
R1 = 5.1 kΩ
R2 = 2.7 kΩ
R3, R7 = 4.7 kΩ
R4 = 47 kΩ
R5 = 470 kΩ
R6 = 50 kΩ Trimmer
R8 = 120 kΩ
R9, R10 = 12 kΩ
R11 = 22 kΩ
R12, R13 = 1.2 kΩ
R14 = 10D391K Varistor

Capacitors

C1, C2, C8...C10, C12 = 100 nF (0805)
C3 = 10 µF, 16 V, Electrolytic (4X5)
C4...C6 = 1 µF (0805)
C7 = 47 µF, 16 V, Electrolytic (5X5)
C11 = 100 µF, 16 V, Electrolytic (6X5)

Semiconductors

D1...D4 = 1N4148 Diode (MINIMELF)
D5 = M7 Diode
IC1 = CD4001D
IC2 = CD4060D
IC3 = CD4013D
T1 = BC817 Transistor
LED1 = Yellow LED, 3 mm
LED2 = Green LED, 3 mm

Miscellaneous

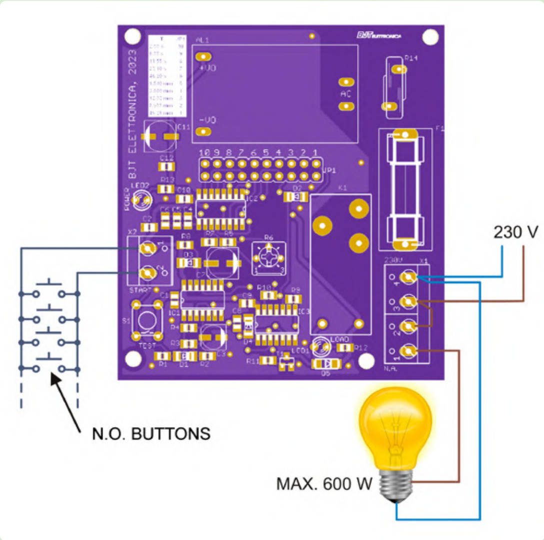
K1 = 12 V Relay 12 A RT114012
AL1 = 12 V DC Power supply, PCB Type
X1 = 2 Pole Screw Terminal Block
X2 = 4 Pole Screw Terminal Block
F1 = 500 mA T 5x20 Fuse
Fuse Holder 5x20, PCB mount
S1 = PCB push-button, N.O.
JP1 = 20 pin, dual line header, 0.1" pitch

Note: The CD4060's (IC2) output Q4...Q14 is missing the Q11 bit (2^{10} = 1024); this explains why the multiplying factors in Table 1 jump from 512 to 2048.

Practical Realization

The board construction is fairly simple. The PCB is double-sided with metallized holes and vias. As you can see from the prototype image at the beginning of this article, the circuit has been made almost entirely with SMD components and a few other traditionally TH-mount parts.

The PCB silkscreen, visible in **Figure 2**, shows the outlines, symbols, and solder points of all the parts listed in the component list. Once assembly is complete, we can move on to testing. **Figure 3** shows the correct wiring for this timer.



▲ *Figure 3: Schematic wiring for a typical application of the programmable timer.*

Checking the Timer

Set R6 trimmer to its maximum value (50 kΩ) and insert the jumper in position 10 of JP1. When the power is applied to the board, the green LED2 should light up. Pressing button S1 (TEST) will energize relay K1, which will de-energize after approximately 2.88 s. You can then check that the timer is working correctly according to the conditions described in detail above.

If necessary, the board can be installed in a plastic box. For this purpose, 3D printable supports have been provided; the respective STL files can be downloaded, aside with all the other material for this project, from the Elektor Labs website at [2]. A YouTube video, showing the prototype at work, is also available [3].

250062-01

A FINAL REMINDER ABOUT SAFETY

Some parts of this design are directly connected to the 230-V AC mains grid, and accidental contact can be dangerous! If you do not have the needed technical background and safety knowledge, do not start this project, or get help from an engineer with the necessary skills.

Questions or Comments?

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

About the Author

Passionate about electricity from an early age, Giuseppe La Rosa graduated with a degree in Electronics and Telecommunications in 2002 at I.T.I.S. “G. Ferraris” of Acireale, Sicily. Later, he began studying microcontroller systems, particularly PIC microcontrollers and the Arduino UNO open-source platform. Over the years, he has created various prototypes, many of which have been published in electronics magazines. Currently, he deals with security systems (video surveillance and burglar alarms) and software for the management of points of sale.

WEB LINKS

[1] CD4060 Datasheet, National Semiconductor: https://tinyurl.com/bdzdypk7
[2] Downloads at Elektor Labs: https://tinyurl.com/yw6xksmn
[3] YouTube Video for this project: https://youtu.be/XqvIjlrJwZQ

28 Smarten Up Your Shutters

Controlling Velux Hardware With an ESP32 and MQTT

By Sébastien Guerreiro de Brito (France)

Ever wished your Velux roller shutters could talk to Home Assistant? This is the hardware way: Using a hacked remote control, an ESP32 and MQTT to bring smart automation to your shutters, while bypassing the proprietary IO-Homecontrol protocol.

Figure 1: The Velux remote ready to be hacked.

Some time ago, I equipped my house with solar-powered roller shutters from the well-known Danish brand Velux. These are controlled by a radio remote using a proprietary protocol called IO-Homecontrol. Since discovering Home Assistant [1] and the MQTT [2] protocol, I wondered: could it be possible to control the shutters via Home Assistant? I did some research but couldn't find any simple way to interface Home Assistant with IO-Homecontrol. However, it seemed possible to hack an official Velux remote! I used one (part number: KLI 313 WW), shown in **Figure 1**, which allows you to control one or more shutters simultaneously. The price varies depending on the seller, I paid about €80 for mine.

The idea was to buy this remote, pair it with my shutters, and electronically simulate button presses by integrating the remote into an electronic circuit. For this, I used an ESP32 module as an MQTT client that receives open or close commands from the MQTT broker and controls the remote. The finished circuit in its enclosure can be seen in **Figure 2**, and **Figure 3** shows the device with the case opened. Let's take a closer look.

Figure 2: The finished IO-Homecontrol to MQTT adapter.

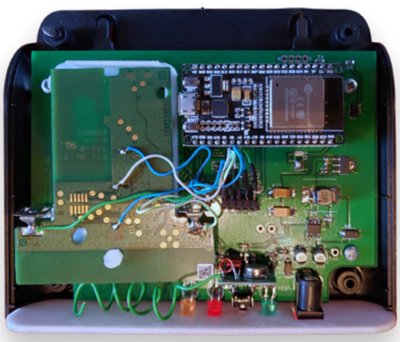


Figure 3: A look under the hood.

Hardware

The first step is to analyze the original remote control, which is straightforward. Once disassembled, you'll see that many test points were conveniently included by the manufacturer! Among them, three are connected to the tactile switches for the *Up*, *Down*, and *Stop* commands. I found that the *Up* and *Down* buttons were by far the most useful and ultimately only these two are used. The corresponding test points have been marked: U for *Up*, D for *Down*, and S for *Stop* (**Figure 4**).

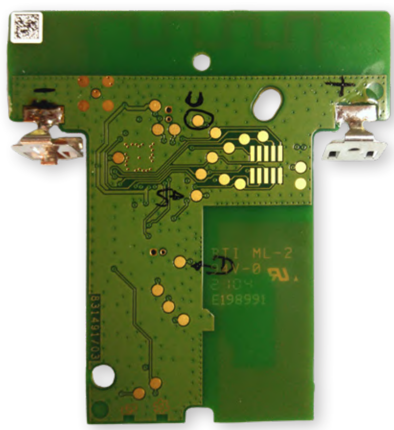


Figure 4: Marking the relevant test points on the PCB, as well as the input voltage polarity.

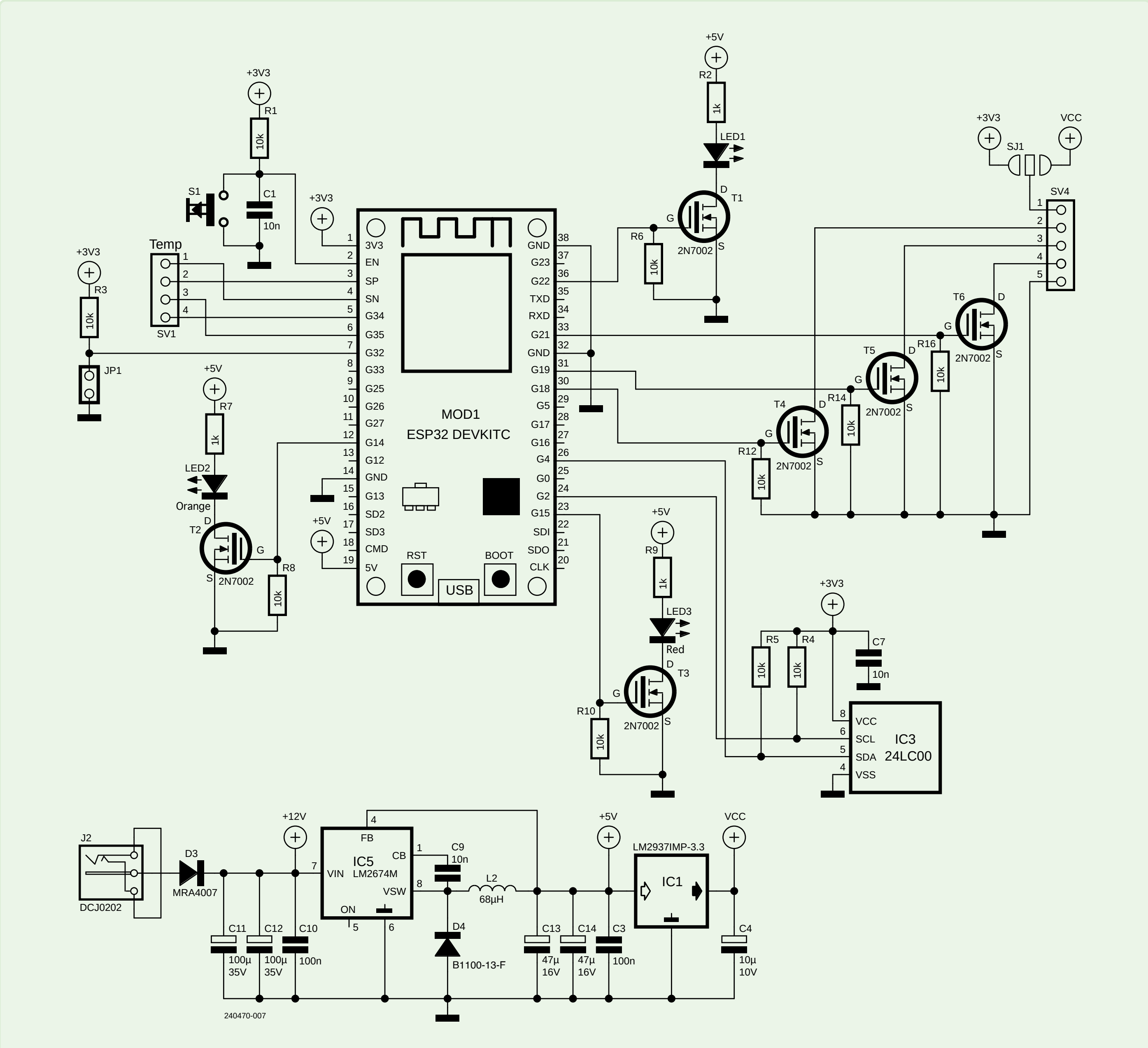


Figure 5: The schematic consists mostly of a few transistors and voltage regulators.

The schematic is shown in **Figure 5**. The device is simple: the core is a standard ESP32 module. To simulate button presses, three output pins on the ESP32 module (G18, G19, G21) are used to control the gate of three N-channel MOSFETs (T4, T5 and T6) in open-drain configuration to briefly connect the test points U, S and D to ground. These test points are connected through the connector SV4. This respectively sends the *Up*, *Stop* or *Down* commands to the shutters through the original Velux remote.

The system is powered by a standard 12-V power adapter. A DC/DC converter steps this down to 5V, primarily to power the ESP32. For this, I chose an LM2674 from Texas Instruments, which belongs to the *Simple Switcher* family and is easy to use. The Velux remote originally runs on two alkaline batteries in series, which is around 2.2 V to 3 V depending on their state of discharge. This is close enough to 3.3 V, so a 3.3 V Low Drop-Out (LDO) regulator is used for this purpose. An LM2937-3.3, again from TI, fits perfectly for this task.

The solder jumper SJ1 allows making or breaking the connection between the 3.3-V output of this regulator (labeled VCC) and the first pin of SV4, which powers the Velux remote. This is useful for testing, but in any case: Be careful to bridge only one half of this solder jumper. The ESP32 module has to be powered by either 5 V or 3.3 V, but not both at the same time.

When designing the board, the jumper JP1 and its pull-up resistor R3 were added in case it were necessary to enable or disable an option, but I ultimately didn't implement this in the code. You might have noticed a small RF transmitter in Figure 3; it ended up not being used, so it won't be described in this article either.

Three LEDs provide user feedback: a green LED flashes to indicate normal operation, a red LED signals a Wi-Fi connection issue, and an optional orange LED is available but unused in this version. Two other optional components were planned but not implemented: a footprint for adding an I2C EEPROM (IC3), since the ESP32 lacks built-in EEPROM, and an NTC temperature sensor for monitoring room temperature (connected via SV1). These could be added later if needed.

Software

The software side is straightforward, and many tutorials (such as [3]) are available for using MQTT with the ESP32. The program first connects to the home Wi-Fi. If this fails, the module reboots. If successful, it then connects to the MQTT broker, with another reboot in case this fails again. The module periodically pings an internet address to check connectivity. In this setup, I assigned a fixed IP address to the ESP32 and used the *AsyncElegantOTA* library, which makes it easy to update the firmware through a web browser at the address *http://192.168.xxx.xxx/update*, without taking the enclosure apart (**Figure 6**).

The software uses a few libraries, such as *ESPAsyncWebServer*, *AsyncElegantOTA* and a few others. For clarity, the source code was divided into several *.cpp* and *.h* files. It's too long to be printed here in full, but feel free to have a look at these files, that you can download on Elektor Labs [4]. The main source file is *WifiSwitch.cpp*. As an example, the `setup()` function in this file is shown in **Listing 1**. A few interesting functions are called there, you can explore the other source files for more details.

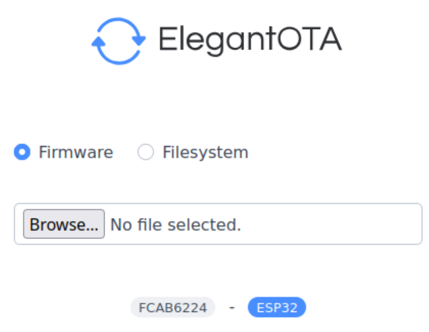


Figure 6: Firmware update is very easy thanks to ElegantOTA.

Listing 1: The setup() function in the WifiSwitch.cpp source file.

```
void setup()
{
    //Serial comport init
    Serial.begin(115200);
    splash();
    //Serial.println("I2C Setup...");
    //Wire.begin(SDA_PIN, SCL_PIN);
    Serial.println("IOs Setup...");
    pinMode (OPEN_BTN,OUTPUT);
    pinMode (CLOSE_BTN,OUTPUT);
    pinMode (STOP_BTN,OUTPUT);

    Globaltimer.attach(TIMER_PERIOD, global_mngt_timer);
    wifi_timeout_timer=WIFI_TIMEOUT;
    WIFI_start_wifi();
    delay(2000);
    //Start MQTT client
    client.setServer(broker, 1883);
    client.setCallback(incomingMQTT);
    delay(500);
    //Start OTA
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
      request->send(200, "text/plain", "Hi! This is WifiSwitch.");
      });
    AsyncElegantOTA.begin(&server);    // Start AsyncElegantOTA
    server.begin();
    Serial.println("OTA server started !");
    //Start main application
    Serial.println("------------------------------");
    Serial.println("Starting application ...");
    led_mngt_init();
    ping_request_timer=PING_PERIOD;
    ping_error_cnt=0;
}
```

For this kind of project, I don't usually use the Arduino IDE. Instead, I rely on a Makefile using the *makeEspArduino* [5] script. However, the Arduino IDE could be used too. Integration with Home Assistant is done like any other MQTT interface. I invite you to consult the official documentation [6] for details. As an example, a snippet of my *mqtt.yaml* file is provided in **Listing 2**.

Mechanical Assembly

As always, I believe a circuit is only complete when it's housed in a proper enclosure! I selected one with a distinctive look and curved back from Bahar, part number BDC 30007-A2, which is available online [7]. I then designed a custom PCB; the corresponding Gerber files are also available on Elektor Labs. This board holds the ESP32 and the internal PCB from the Velux remote. A 3D-printed plastic bracket secures the remote's PCB in place (**Figure 7**), using a nylon screw to avoid interfering with

Listing 2: Snippet of the mqtt.yaml file for Home Assistant integration.

```
 - cover:
     unique_id: "maison_logia_volets"
     name: "Volets de la maison"
     command_topic: "domobus/Maison/Logia/volets/set"
     payload_open: "OPEN"
     payload_close: "CLOSE"
     availability:
       - topic: "domobus/Maison/Logia/available"
     payload_available: "online"
     payload_not_available: "offline"
     optimistic: false
     qos: 0
     retain: false
```

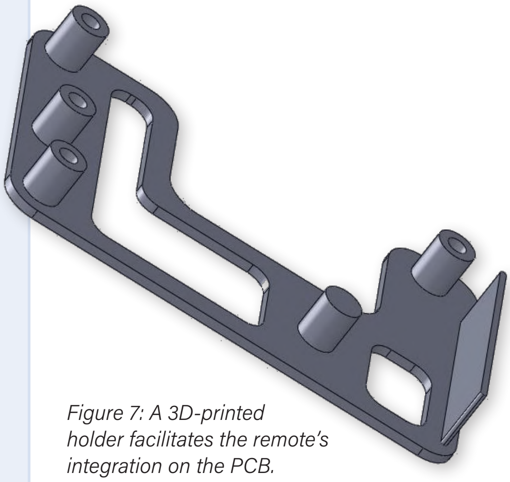


Figure 7: A 3D-printed holder facilitates the remote's integration on the PCB.

the antenna. Since a 3D printer was available, I also printed the enclosure's front panel.

A Good Opportunity to Learn

I can now control my shutters via smartphone, tablet, PC, or even a personal voice assistant I have built. This little project turned out to be quite useful, as it got me started with using the MQTT protocol on the ESP32. Since then, I've built several other circuits based on the same principle, expanding my home automation setup. Learn more about MQTT, Home Assistant and have fun!

240470-01

About the Author

Sébastien Guerreiro de Brito first discovered electronics during his teenage years when he bought a July/August double issue of *Elektor*. After earning an engineering degree in electronics and industrial computing from Polytech Nantes, he developed various boards for personal use. To share his passion, he frequently publishes some of these boards in *Elektor*.

Related Products

- **ESP32-DevKitC-32E**
  www.elektor.com/20518
- **Edi's Techlab, *Home Automation and Electronics for Starters* (Elektor 2024, E-book)**
  www.elektor.com/20831

Questions or Comments?

Do you have questions or comments about this article? Feel free to contact Elektor at editor@elektor.com.

WEB LINKS

[1] Home Assistant: https://www.home-assistant.io/
[2] MQTT on Wikipedia: https://en.wikipedia.org/wiki/MQTT
[3] MQTT tutorial with ESP32, ESP32 I/O: https://esp32io.com/tutorials/esp32-mqtt
[4] Software on Elektor Labs: https://www.elektormagazine.com/labs/elektor-articles-software-downloads
[5] MakeEspArduino on GitHub: https://github.com/plerup/makeEspArduino
[6] Home Assistant MQTT Documentation: https://www.home-assistant.io/integrations/mqtt
[7] Bahar enclosure, AliExpress: https://www.aliexpress.com/item/1005005050458272.html

29 Solid-State Foot Warmer

Energy-Efficient Comfort

By Roel Arits (The Netherlands)

Need a compact, energy-efficient foot warmer? This design features TIP41C power transistors for heating. By applying 30 to 60 W of power, an aluminum plate quickly reaches a comfortable temperature of around 40°C to 45°C. An analog proportional temperature control circuit, using a transistor as a temperature sensor, maintains stable heating without on-off cycling. The design emphasizes safety, thermal efficiency, and cost-effectiveness.

With the extreme high-energy prices, and specially gas prices, at the time of writing (November 2022), I was thinking about an energy-efficient way to keep the feet or hands warm, to minimize the use of the gas powered central heating.

Most electrical heaters that use convection or radiation to transfer heat need quite a bit of power (1000 to over 2000 W) to heat the air or objects from a distance in a relative short time. When you want to have a more direct heat transfer that requires less power, heat conduction with direct contact to your body is the way to go. On the other hand, the best way to resist the cold is to keep your feet warm because that gives an overall comfortable feeling.

The Plate

For heat transfer by conduction, we require a metal plate that has a good thermal conductivity so the heat that is produced spreads easily over the entire area of the plate. Copper, silver or gold would be ideal in that regard, but it would defeat the purpose of saving costs. Aluminum is a good and cheap alternative, though.

But how thick should the plate be? If we'd use a plate of, let's exaggerate, 1 cm thick, it would take quite some time to warm up the whole plate. But once warmed up, it would be easier to keep the plate up to temperature due to the high heat capacitance of a thick plate. But we would rather not wait for an hour before our plate is heated up. An aluminum plate with a thickness of 2 mm will be fine, so heating the volume of material up doesn't take too long.

To warm up our feet while wearing socks, we will need the plate to warm up to about 40° to 45°. We need to overcome the thermal resistance of the socks. Why not bare feet? In fact, to increase safety, as we will see later.

Can we calculate or predict how much time it takes to heat an aluminum plate when using a certain electrical power? We can use the following formula to get a rough idea:

Time needed to heat a volume of aluminum [s] = Heat capacity of aluminum [J/g K] × volume [cm^3] × density [g/cm^3] × delta temperature [K] / input power [J/s]

- Heat capacity of aluminum = 0.902 J/(g K)
- Volume of the aluminum plate = 35 cm × 25 cm × 0.2 cm = 175 cm^3
- Density of aluminum = 2.7 g/cm^3
- Temperature change that we need = 45°C (target temperature) - 17°C (ambient temperature) = 28 K
- Input power that we want to apply = 60 W = J/s

When we fill in the values, we get:

Time = (0.902 × (35 × 25 × 0.2) x 2.7 × 28) / 60 = 198 s = 3.3 minutes

So, by inputting 60 W of power, it lasts 3.3 minutes to increase the temperature of the aluminum plate by 28 K (= 28°C difference). This means it would take 6.6 minutes when using 30 W or 13.2 minutes with 15 W to increase the temperature by 28 K. Of course, this is theory. There will be some additional losses due to inefficient power transfer, thermal resistance between the heating device and the aluminum

Figure 1: Nice and warm, just big enough to put both feet on.

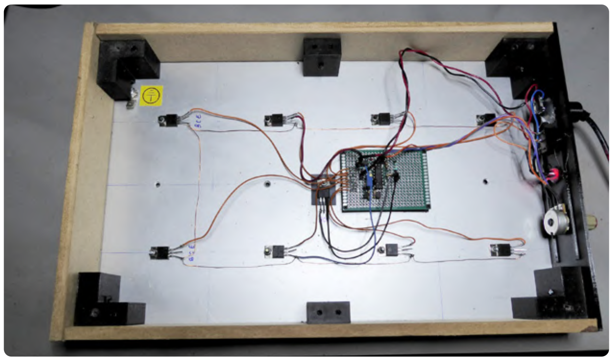


Figure 2: View on the ugly wiring, the electronics and 3D printed spacer blocks. The eight power transistors are evenly spread over the plate area. The power transistor in the middle is the temperature sensor.

plate, density difference in the aluminum. Plus the fact that we don't know exactly what type of aluminum alloy the plate is made of, etc.

When we can adjust the input power between ca. 20 W and ca. 70 W, we will have a comfortable temperature to warm the feet in low ambient temperatures without waiting very long for the plate to be heated up. For example, I used the feet warmer at 14°C ambient temperature with an input power of about 30 W to get my feet covered in thick socks warmed up to ca. 35° (**Figure 1**).

How the Plate Heats Up

Why did I call it a solid state foot warmer? For the heat production, we use eight bipolar power transistors — more precisely, TIP41C NPN power transistors. They can handle a maximum total power dissipation of 65 W, when their package doesn't exceed 25°C, and with a collector current of max. 6 A. I've chosen this type because I had enough of them lying around. Any other NPN power transistor with a similar power rating will do fine, as long as it is not a Darlington type.

I also wanted to use a device with a TO-220 casing because that is easy to screw down on an aluminum plate and is big enough to transfer enough heat into the plate. A device in a TO-247 case can also be used, but that is even more overkill since these are used for devices with close to 100 W of power. Since we are using eight power transistors, each transistor only needs to handle 10 W at the max.

Schematic

By distributing the eight power transistors equally over the area of the plate, you certainly don't feel hot spots on the plate at the locations where the transistors are mounted to the plate (**Figure 2**). In the schematic, Q1, Q2, Q5, Q6, Q8, Q9, Q11, Q12 are the heat producing power transistors (**Figure 3**). The aluminum plate I used is 2-mm thick and has a size of 35 cm × 25 cm. Be aware that when you take an aluminum plate that is twice as thick, you will need to wait twice as long before the plate is heated up. Or you have to increase the input power with a factor 2 to accelerate the process.

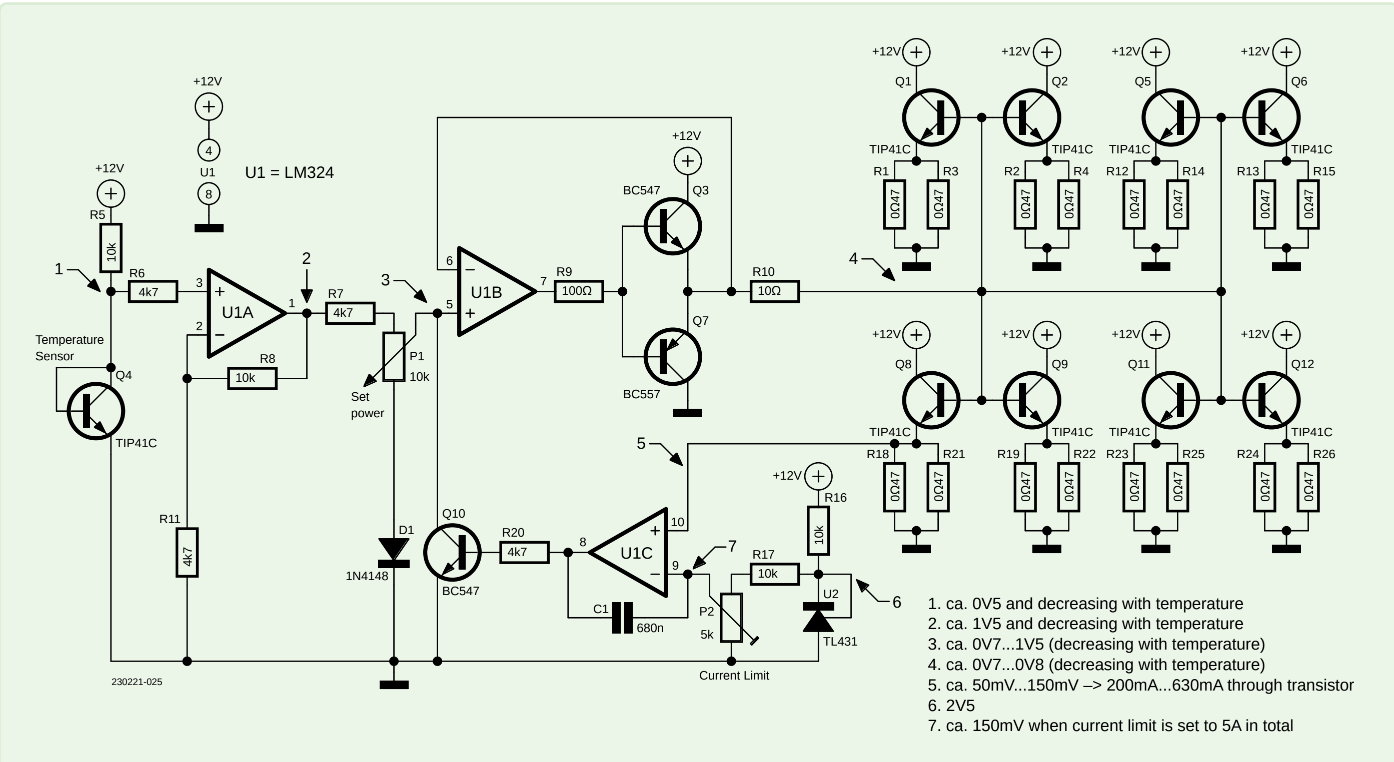


Figure 3: Circuit diagram.

Figure 4: Side view with the adapter plug and knob to set the power. Temperatures are just an indication, since the end temperature is relative to the ambient temperature.

Component List

Resistors
R1...R4, R12...R15, R18, R19, R21...R26 = 0.47 Ω/0.25 W
R5, R8, R16, R17 = 10 k / 0.25 W
R6, R7, R11, R20 = 4.7 k / 0.25 W
R9 = 100 Ω
R10 = 10 Ω
P1 = 10 k linear potmeter
P2 = 5 k linear trimpotmeter

Capacitors
C1 = 680 nF /25 V

Semiconductors
Q1, Q2, Q4...Q6, Q8, Q9, Q11, Q12 = TIP41C
Q3, Q10 = BC547 (or generic NPN)
Q7 = BC557 (or generic PNP)
D1 = 1N4148
U1 = LM324
U2 = TL431

Now we come to the question of how to implement the temperature control and how to measure the temperature. I would rather not use an on-off temperature control, as is commonly implemented using a thermostat. This kind of rough temperature control makes the temperature oscillate around a fixed temperature. A PID temperature controller also seemed a bit overkill to me because we're not so much interested in a very precise temperature control. So I decided to go for an analog proportional temperature control because that gives a nice steady temperature without any on-off behavior.

To measure the temperature, I used the same type of power transistor as is used for the heating. I had enough of them lying around, they are easy to mount and give a good thermal contact with the aluminum plate. When using an NTC thermistor, you still need to figure out how to make a good thermal contact with the plate. So a power transistor is ideal in that regard.

The temperature sensor is transistor Q4 in the schematic. The negative temperature coefficient of the base-emitter junction of Q4 is used to control the temperature of the plate. The base-emitter voltage of Q4 will drop in a rather linear way with rising temperature, in the temperature range that we are aiming for.

The temperature dependent base emitter voltage of Q4 is amplified ca. 3× by the opamp U1A (LM324) which is configured as a non-inverting amplifier, resulting in a voltage of 150 mV that decreases when temperature rises. The output voltage of this opamp is used to control the voltage over potentiometer P1 that is used to set the power that goes into the heater transistors (**Figure 4**). The voltage at the potentiometer is buffered by opamp U1B, which is configured as a voltage follower with a current booster, formed by Q3 and Q7. The current booster provides the current for the bases of the eight power transistors.

When potmeter P1 is set to a certain value, the power transistors will start heating the aluminum plate with a power close to the maximum power. While the plate is heating up, the voltage over the base-emitter junction of Q4 will start dropping along with the rising temperature. When this voltage drops, this means that the voltage at the junction between R7 and P1 decreases. The minimum voltage that can be set with P1 is 0.7 V due to D1. So with the minimum setting of P1, the heater transistors are already at the edge of conducting. Without D1, you would need to turn P1 more than halfway to actually start heating.

This means that opamp U1B will provide less and less current to the bases of the eight power transistors while the plate is heating up. So, the temperature rise will slow down to the point where a balance is reached between the temperature generated by the power transistors and the temperature sensed by the temperature sensor. At what temperature, this balance will be reached is determined by the position of potmeter P1 and the ambient temperature that the plate has to conquer.

The maximum current is limited by a current limiter formed with the circuit around opamp U1C. The voltage across the emitter resistors of Q8 (2x 0.47 Ω parallel) is used to measure the current that is flowing through the transistor. We can assume that all eight transistors draw about the same current, so it is enough to measure the current through one of them to know the total current drawn by all eight power transistors together.

The current limit needs to be adjusted with trim-potmeter P2 so the total current drawn by all eight power transistors together is maximum 5 A. So the current through one of the power transistors should be maximum ca. 5 A / 8 = 0.63 A. A current of 0.63 A through the 2x 0.47 Ω emitter resistors in parallel (= 0.235 Ω) results in a voltage of 0.63 A × 0.235 Ω = ca. 150 mV (test point 5 in the circuit diagram).

Opamp U1C compares this voltage to the voltage set by P2. The voltage set by P2 is derived from a stable voltage reference U2 (TL431). U2 is configured to output a 2.5 V reference voltage. This reference voltage is independent of temperature or input voltage, so our current limit setting is not affected by input voltage or temperature fluctuations. Because the PCB that house-vests all the electronics is mounted close to the heater plate (**Figure 5**), we needed to make sure that the temperature of this plate does not influence the current limiter setting.

When the voltage over the emitter resistors of Q8 becomes higher than the voltage set by trim-potmeter P2, the output of opamp U1C will swing to the positive power supply voltage **(Figure 6)**. This will cause Q10 to conduct and will pinch down the voltage set by P1 to nearly 0 V. This means that the power going to the power transistors gets cut off when the total current drawn by the eight power transistors would exceed 5 A.

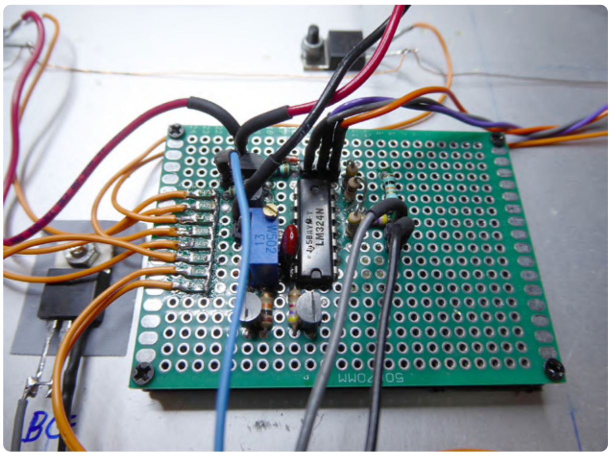


Figure 5: Temperature controller and current limiter with LM324 opamp.

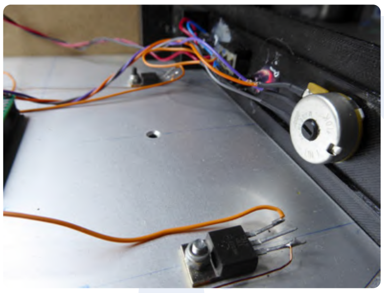


Figure 6: Potmeter, LED and power switch.

This way, the current limiter protects the eight power transistors and makes sure that the maximum power can never be exceeded. C1 is there to reduce the gain at high frequencies to keep the current limiter stable.

The 12 V DC input voltage is supplied by a switching power adapter that can deliver a maximum of 72 W of power at 12 V, which means it can deliver a maximum current of 6 A (**Figure 7**).

When the plate is cold, the temperature sensor voltage will be maximum (low temperature), so the power transistors get maximum base voltage/current and would draw a large current when we wouldn't use a current limiter. Once the plate is warmed up, the voltage over the temperature

Figure 7: The switching power adapter and power plug.

sensor will progressively decrease, thereby decreasing the current going through the eight power transistors.

So only at the beginning, when the plate is cold, we need to limit the current. By limiting the current, the power transistors will always operate in their safe operation area. This also prevents exceeding the maximum power of the power adapter, which is powering the feet warmer. R9 and R10 protect U1B and the current booster (Q3 and Q7) against failure by limiting short-circuit currents. It is good to have a current limit anyway to protect the electronics and the power adapter, just in case something breaks.

230221-01

SAFETY WARNING:
When the collectors of the power transistors are all electrically connected to the aluminum plate, the plate will carry the 12 V power supply voltage. I didn't use a medical rated power supply with a guaranteed high voltage isolation. So for safety, use a silicon thermal pad (Sil-pad), mica and isolation kits for each of the eight power transistors that are mounted to the plate, so their collectors are electrically isolated from the plate. By the way: I always wear socks when putting my feet on the plate.

When you decide not to isolate the heater transistors, so their collectors are electrically connected to the aluminum plate, then make sure that the transistor (Q4) that is used as the temperature sensor is isolated from the plate. Otherwise, the collector of this transistor (Q4) will be short-circuited to the positive supply rail and lead to failure.

About the Author

Roel Arits is an electronics engineer, born in 1964, with a background in writing embedded software, developing test systems and designing analog and digital hardware. Because he worked in several areas during his career, he gained broad knowledge, including optical systems, medical devices, professional audio and mechanical design. In his spare time he enjoys flying RC model airplanes, riding motorcycles, traveling, learning about all kinds of technical area's and passing basic electronics knowledge via his blog.

Questions or Comments?

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

Related Products

Is the M5Stamp Fly Quadcopter the Next Tello?

By Jim Solderitsch (USA)

With the Ryze/DJI Tello drone discontinued, finding a low-cost, programmable alternative has become a pressing challenge. Could the new M5Stamp Fly step in as a capable and affordable replacement?

I have been using drones for educational courses (online and classroom based) where drone programmability has been an important concern. The drone I have mainly been using, the Ryze/DJI Tello, has been withdrawn from commercial sale, and I am looking for inexpensive alternatives to use in my courses instead. When the M5Stamp Fly was announced, I was hoping that this platform could be the alternative I was looking for. Does it qualify in this regard?

The controller and drone come as separate products, but out of the box, there is no way to fly the M5Stamp Fly [1] without the M5Atom Joystick [2].

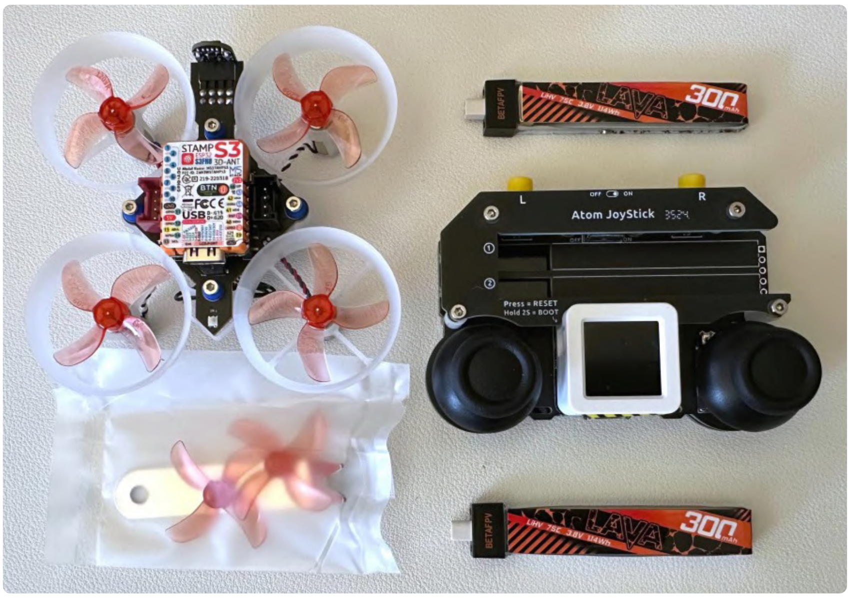


Figure 1: M5Stamp Fly and M5Atom Joystick unboxed.

Here is what you get in the two product boxes (**Figure 1**):

- M5Stamp Fly drone
- Two batteries (one in each box)
- Two spare propellers with a propeller removal tool (one for right rotation and one for left rotation)
- M5Atom JoyStick Controller

There are two charging slots in the body of the Joystick device where both batteries can be recharged using a USB-C charging cable (not supplied in either box).

M5Stamp Fly Product Summaries

The Elektor product pages can be found here: M5Stamp Fly [1] and M5Atom Joystick [2]. These pages contain M5Stack-supplied pictorial summaries of the Stamp Fly and Atom Joystick Controller as shown in **Figure 2**.

The product pages both have a reference to the same video that documents the pairing process for the Fly and Joystick controller. While there is a brief flying demo at the end of the video, there is no help within the video to orient a new user on how to use the joystick controls to safely fly the drone. Hints on this are provided below.

Notable included features of these devices are highlighted in each figure. Each device is based on an ESP32S3FN8: StampS3 on the drone and Atom CoreS3 on the Joystick. Both are connected to the main PCB through pin headers and are thus removable.

Figure 2: Side-by-side Stamp Fly and Atom Controller summaries. (Source: M5Stack)

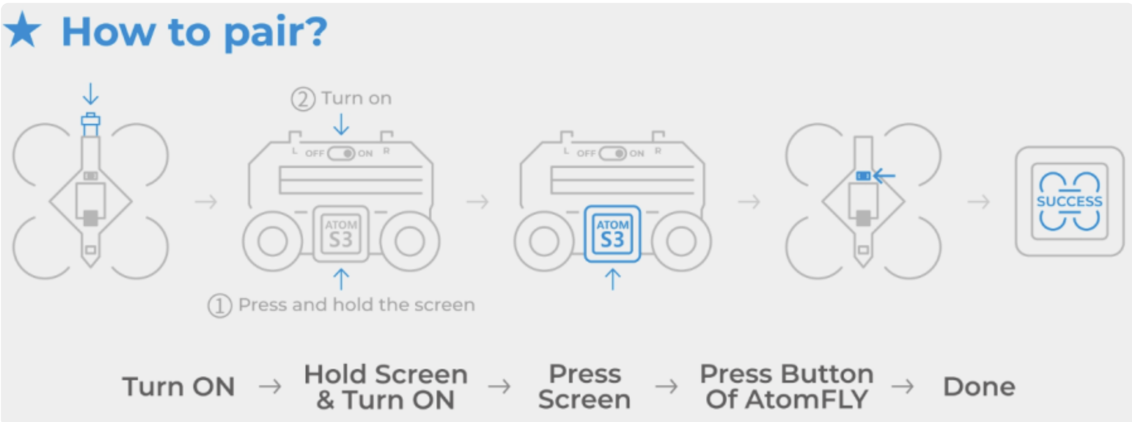


Figure 3: Stamp Fly and controller pairing process. (Source: M5Stack)

In addition, the Stamp Fly includes a BMI270 IMU sensor, a BMM150 magnetometer, a BMP280 pressure sensor, two VL53L3C Time-of-Flight (ToF) sensors, a PMW3901 optical flow sensor, a *Reset/Pair* button for pairing and calibration reset, and two WS2812C status RGB LEDs (top and bottom).

The Atom Joystick includes an additional MCU (STM32F030), two buttons for settings changes, two Hall effect joysticks, two RGB LEDs for battery charge monitoring for each charge bay (red means charging is happening and green means charging is complete), and a USB-C charge port.

Setting Up the M5Stamp Fly

While not necessary, there are instructions here on how to update the firmware to the latest official version. I performed a firmware update because I wasn't sure whether the behavior I observed during a takeoff attempt in *Altitude Auto* mode was expected. After updating the firmware, you'll need to pair the Joystick and Fly again if they were previously paired. Once paired, you are ready to fly the drone with the controller.

Pairing Process

The pairing process is summarized in **Figure 3**.

- Press and hold the center button on the Atom to power on, then follow the screen prompts to enter pairing mode by pressing the button again.
- Press the reset button on the Stamp Fly to send the pairing signal.
- Wait for pairing to complete.

LED Status Lights of the M5Stamp Fly

Before flying, it is helpful to learn the meaning of the LED status light colors that appear on the rear of the Fly, as seen in **Figure 4**. The same LED status light effects are shown on the LED at the bottom of the Stamp Fly.

The top part of Figure 4 documents *Power On* effects, while the bottom one shows operational ones. You can cause a self-check with the Fly on the ground by pushing the reset button on the top front of the Fly just below the M5Stamp S3 board. This will rerun the self-check.

1. Power On

- White: Device self-check initiated.
- Purple: Sensor offset calibration in

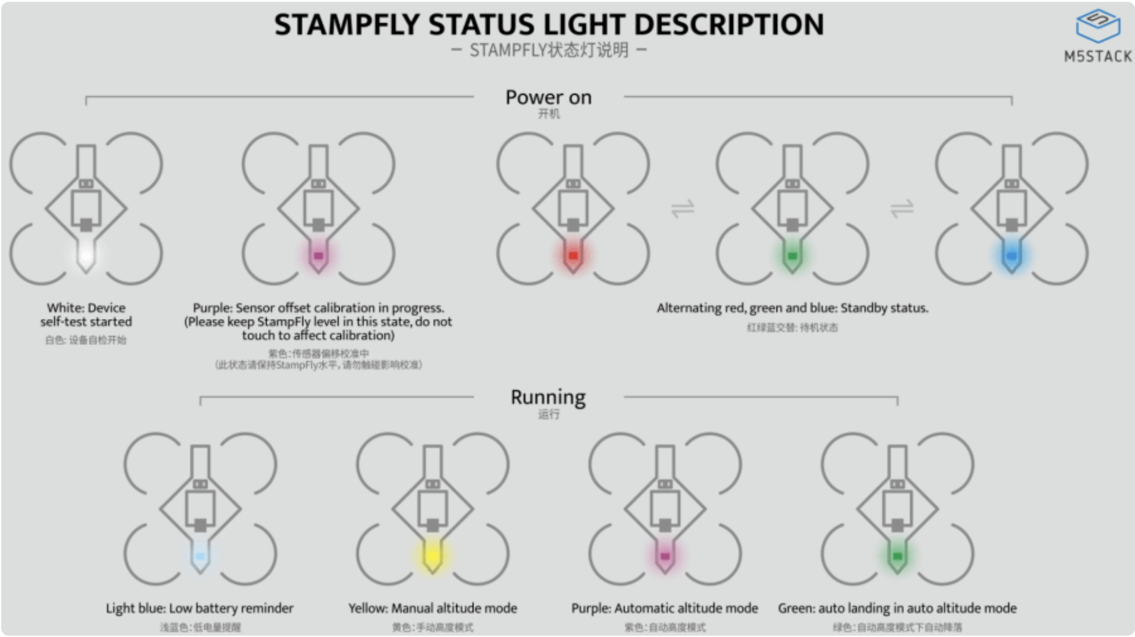


Figure 4: Stamp Fly status light meanings. (Source: M5Stack)

progress (keep the Stamp Fly level and avoid touching to prevent calibration interference).
- Red, green, and blue alternating: Standby mode (ready for takeoff).

2. Operation
- Light blue: Low battery warning.
- Yellow: Manual altitude mode.
- Purple: Auto altitude mode.
- Green: Auto descent in auto altitude mode.

First Flight

The firmware documentation [3] summarizes the two function modes, along with the "trick" flight effect of a flip.

1. Left front button on Atom Joystick (*Control Mode* switch)
- Stabilize Mode: Provides stable control, suitable for regular flight and cruising.
- Sport Mode: Offers maximum control freedom, allowing for complex maneuvers but requiring high operational skill.

2. Right front button on Atom Joystick (*Altitude Mode* switch)
- Auto Altitude: Maintains a stable altitude at the set level; pushing the left joystick up or down will change the set altitude.
- Manual Altitude: Full throttle control of altitude, requiring higher operational skill.

3. Center button on right joystick (*Aerial Flip* action)

Controller Operation

When in *Auto Altitude* mode (recommended for inexperienced fliers), the screen button on the controller is used to initiate takeoff when the *Ready-to-Fly* message appears on the screen as shown in **Figure 5**. While in flight, press the controller screen button to land.

You should make sure the controller screen indicates that the flight modes are set as shown in **Figure 6** of the CoreS3 controller screen. Flight modes are set with the yellow buttons at the top of the controller as shown in Figure 2.

M5Stamp Fly Flight Experience

After the firmware was updated, I was a little braver in my flight actions and in auto altitude. Once takeoff happens, I can assert control with the right joystick to keep the Fly from drifting off. The altitude gets a little higher when that happens. The altitude holding wasn't very good, but still did not need a lot of micromanaging using the left joystick. The drone did yaw a bit without joystick intervention, but I could use the left joystick to compensate. Usually, the altitude changed a little too, when I did this, but not drastically. Most of the time, the land button (center screen button) worked to land the Fly. You need to always be attentive with the right joystick to keep the Fly stable.

Figure 5: Flight controller artificial horizon view.

Figure 6: Flight modes set for first flight.

When flying, the LCD screen on the AtomS3 shows an artificial horizon that indicates the Fly's flight orientation in real-time.

If you leave the right joystick in the center (home) position without hands-on control, the joystick does NOT provide drift-free/hover behavior, and the Fly will drift unless controlled using the right joystick. I initially expected takeoff and hover stabilization based on my experience flying the Tello drone. I think my expectation was incorrect because the *Altitude Auto* setting (sometimes called *Altitude Hold* for other drone platforms) is not the same as *Position Hold*, which the Tello does support. In *Position Hold* mode, the drone would assume a hover position with only a minimal amount of drift without touching any joystick.

Open-Source Considerations

Both the M5Stamp Fly drone [4] and the Atom controller [5] have GitHub pages. The code is presented as running using the PlatformIO framework rather than the Arduino IDE. I looked for but did not find instructions to help with programming the M5Stack devices reviewed here using Arduino. Other M5Stack devices seem to be well supported from the Arduino perspective, and perhaps I'm just missing an illustrative example to help me learn how to program basic alternate flight control patterns, such as implementing *Position Hold*.

ESP-NOW

The main communication protocol used for controller-to-drone communication is ESP-NOW. This protocol is very well documented, and numerous examples exist, including many that work inside the Arduino IDE for ESP32 platforms in general. In fact, the picture at the top of the ESP-NOW page [6] shows a dual Joystick controller that looks very much like the controller sold as the flight controller for the Stamp Fly drone.

There is no high-level description of the basic API, at least none that I have been able to find. There is nothing like the Tello drone API information that was published by the Tello production company Ryze/DJI, which has been used in a number of independent projects, including a robust Python library called *DJITelloPy*.

Note that the Tello uses Wi-Fi communication to receive flight commands from external devices like the ESP32-based controller I have built for my courses. This communication style is well documented.

Since the Stamp Fly and Atom Joystick firmware source is available on GitHub, it should be possible to reverse engineer both a rudimentary API and the ESP-NOW communication mechanisms used between these devices.

Manual Flight

For manual flight, the Stamp Fly performs as minimally documented in the M5Stack online information pages. There are no significant printed materials included in the product boxes except the status light summary as a small paper insert and the pairing process printed on the back of each box.

With proper coaching, the drone is easy enough to control so that kids might enjoy flying the drone using *Stable Control* Mode and *Auto Altitude*. Anyone with experience flying drones using the same joystick setup — *Throttle* and *Yaw* on the left joystick, and *Pitch* and *Roll* on the right — can fly with *Sport* Mode and *Manual Altitude*. Since I was testing this product in an indoor location with limited air space, I did not try this method of flying.

Programmable Flight

This product is not yet positioned to take the place of the Tello in the educational drone space. For this to happen, the API documentation must be refined to describe the basic communication patterns employed as well as a list of commands that can be sent from the controller to the drone. For example, a programmable flight requires at least an approximation of *Position Hold* flight behavior so that a *Takeoff* and then *Hover* opening flight action without joystick intervention must be possible.

I plan to do some research of my own to see if I can learn the inner workings of the Stamp Fly and drone-to-controller communication. I would like to adapt my own ESP32-based controller that flies the Tello to also fly the Stamp Fly.

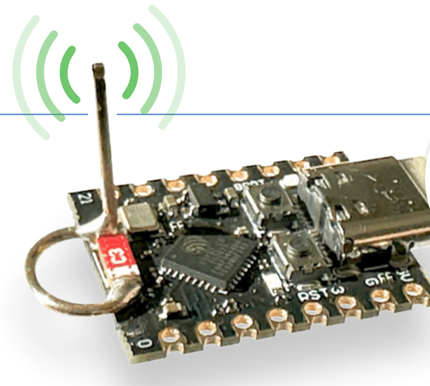
250330-01

Related Products

- **M5Stamp Fly Quadcopter (with M5StampS3)**
  www.elektor.com/21008
- **M5Atom Joystick (with M5AtomS3)**
  www.elektor.com/21007

WEB LINKS

[1] M5Stamp Fly Quadcopter: https://www.elektor.com/products/m5stamp-fly-quadcopter-with-m5stamps3
[2] M5Atom Joystick: https://www.elektor.com/products/m5atom-joystick-with-m5atoms3
[3] Firmware documentation, M5Stack: https://docs.m5stack.com/en/guide/hobby_kit/stampfly/stamply_firmware
[4] M5Stamp Fly drone on GitHub: https://github.com/m5stack/M5StampFly
[5] Atom controller on GitHub: https://github.com/m5stack/Atom-JoyStick
[6] ESP NOW, Espressif: https://www.espressif.com/en/solutions/low-power-solutions/esp-now



30 Boosting Wi-Fi Range of the ESP32-C3 SuperMini

A Simple and Effective Antenna Mod

By Peter Neufeld (Germany)

The ESP32-C3 SuperMini modules are incredibly affordable, costing around €2, and are equipped with a compact SMD antenna. However, this tiny antenna significantly limits the usable Wi-Fi range, due to its design. To address this issue with minimal effort, I implemented a simple antenna modification that drastically improved performance.

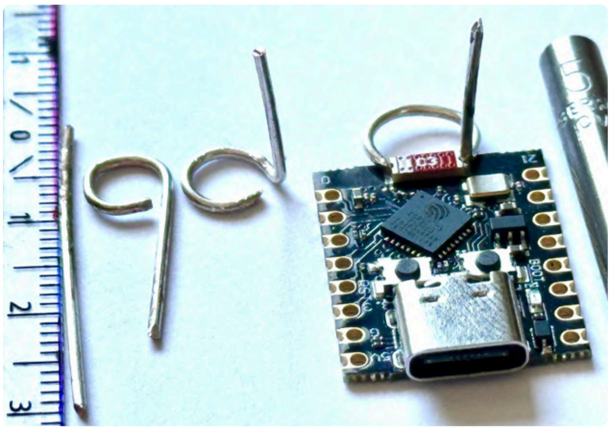


Figure 1: Part of the wire is bent into a horizontal loop, while the remaining section is angled vertically upwards.

This modification involves adding a 1.0-mm silver-plated wire with a total length of 31 mm, configured as a quarter-wavelength (λ/4) antenna. The bottom section of the wire is bent into a horizontal loop (approximately 16 mm of the wire length, forming an oval loop with a diameter of about 8 mm), while the remaining section is angled vertically upwards (**Figure 1**).

I wrapped the circular loop around the shaft of a 5-mm drill and then widened the ends of the loop so that they touch the terminals of the SMD antenna (**Figure 2**). The SMD antenna effectively completes the wire loop — mechanically at a quarter of its circumference, but electrically as a λ/4 element in parallel with a λ/8 wire.

I then soldered the new antenna directly to both ends of the original SMD antenna on the ESP32 module. Specifically, I soldered the wire to the 50-Ω antenna pin (the left end of the original antenna) and to the other "hot" end of the original antenna. This effectively bypasses the original PCB antenna. In any case, two good solder joints at both ends of the PCB antenna are crucial and one have to pay attention to the position of the wire that goes up from the loop. The old aerial has been left in place as it becomes electrically ineffective in this configuration — more on this below.

Testing with a Wi-Fi Logger Program

To evaluate the effectiveness, an original and a modded module were mounted side by side on a portable power bank to ensure identical testing conditions while moving around various locations near an

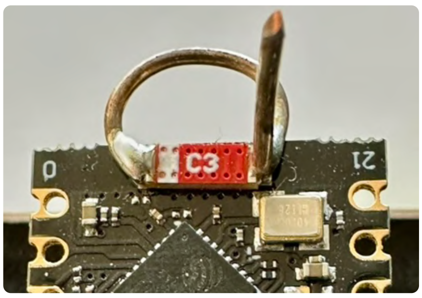
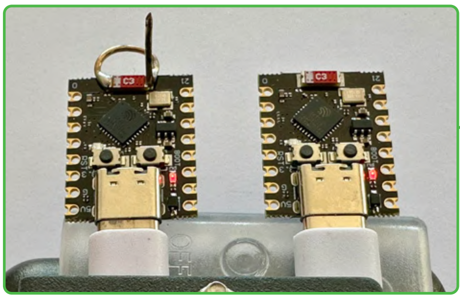


Figure 2: The SMD antenna effectively completes the wire loop.



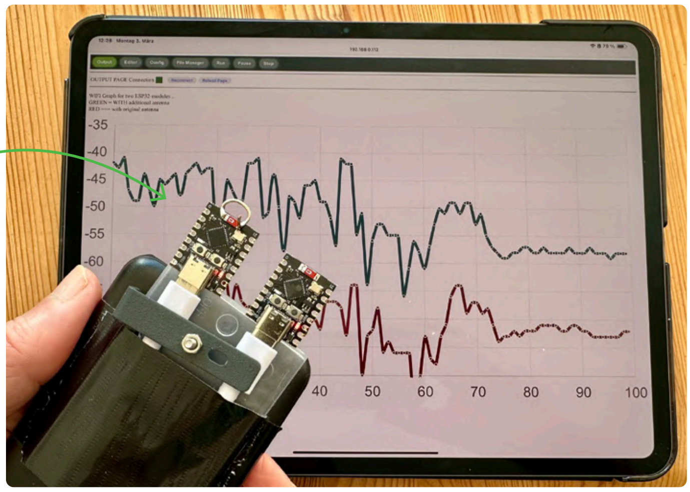


Figure 3: To evaluate the effectiveness, both modules were mounted side by side on a portable power bank.

access point (**Figure 3**). With a self-written ESP32 program, I monitored the Wi-Fi strength over time. The curves are displayed on a webpage, offered by one of the ESP32 controllers, which can be shown in a browser window, for example on a tablet.

Both ESP32 boards run the same BASIC script (see textbox **Programming with Annex32 Basic**). On one board, the script is configured to also retrieve and display the signal strength measured by the other ESP32, presenting it as a second signal curve. This setup effectively visualizes the impact of the antenna modification under otherwise identical conditions. My code is available for download from [1].

Programming with Annex32 Basic

It has turned out for me that I can program my ESP32 projects most effectively with Annex32. This is an extremely versatile BASIC interpreter and an integrated development environment that runs completely in the ESP32 after a one-time firmware installation. Fast browser-based program development and adaptations are thus extremely simplified.

Forget all that you think you know about ancient BASIC dialects… It's worth taking a look at the Annex32 Firmware for the ESP32 at [2]. The web-based Annex32 firmware installation can be found at [3].

Results

The signal strength curves recorded with the Wi-Fi Logger program consistently demonstrated a clear superiority of the modified module over the original (**Figure 4**).

On average, the improvement in signal strength was at least approximately 6 dB. In many cases, particularly at the edges of the Wi-Fi signal range or in environments with more interference, the improvement even exceeded 10 dB. This difference in signal strength had a significant impact on the stability of the connection. The modified module maintained a stable connection while the unmodified module was more prone to disconnections or performance degradation. In environments with obstacles (e.g., walls), the modified antenna showed better penetration and provided a more stable connection.

These improvements led to a significantly extended Wi-Fi range. Theoretically, every 6 dB increase in signal strength corresponds to a doubling of the range.

Observations

The original SMD antenna CrossAir CA-C03 placed at the ESP32-C3 SuperMini module — due to its small size, high dielectrical ceramic material, and the compact antenna geometry — is inherently less efficient at radiating and receiving RF energy than an in-air-standing antenna wire. Additionally, the original antenna is not positioned at the location on the PCB explicitly recommended by the manufacturer

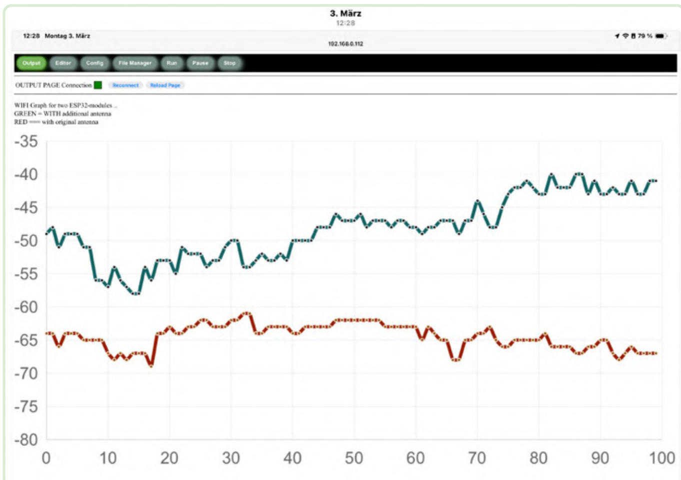


Figure 4: Signal strength curves recorded with my Wi-Fi Logger program. The modded antenna clearly wins the contest!

Caveats of a Compact Antenna Design

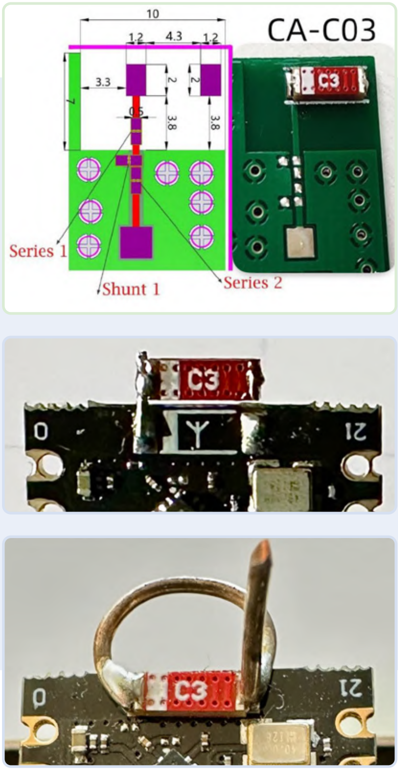
The Datasheet [4] of the SMD antenna CA-C03 by Cross Air requires:

- PCB edge positioning,
- sufficient clearance around it,
- a 3.8 mm × 0.5 mm stripline,
- a 3 nH inductance "Shunt 1"

The actual ESP32-C3 SuperMini PCB design lacks several key features, most notably adequate free space around the antenna and an orthogonally radiating stripline, which also affects the antenna's resonance frequency. These limitations result in misalignment, partially reflected RF energy and reduced RF efficiency.

An experimental, freestanding mounting of the SMD antenna, more in line with the datasheet specifications, has already demonstrated an approximate 4 dB improvement in gain. However, this could not overcome the fundamental limitations of this very compact SMD antenna compared to the simple quarter-wave wire antenna described as a modification here.

The latter angled wire antenna benefits from its electrically non-shortened geometry and a more unobstructed and more effective omnidirectional radiation, achieving sure gains of 6 to 10 dB compared to the original SMD antenna.



Source of the PCB drawing: [4]

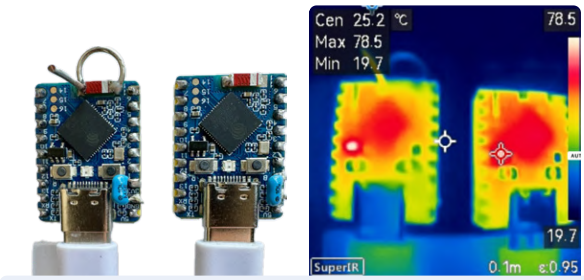
in the datasheet [4]. Instead, the SMD antenna is mounted too close to the ground plane and other components. This positioning shields the RF energy rather than allowing for effective radiation.

It was observed that strong signal reflections caused the chip's internal protection circuits to temporarily reduce performance. This explains why, paradoxically, intentionally lowering the transmit power can sometimes lead to more stable near-field communication.

In contrast, the new, non-shortened and freestanding quarter-wave antenna can produce an undampened and more consistent field. This is likely due to the combination of now unshielded horizontal and vertical elements, which helps minimize null points in the omnidirectional radiation pattern and enhances signal coverage. The loop at the bottom of the antenna creates a horizontal radiation pattern without nulls, while the orthogonally angled element optimizes the vertically polarized component of the signal. You can find even more details at [5].

However, it is important to note that the correct placement and orientation of the angled end of the wire, along with its total length and diameter, are crucial for optimal performance.

250174-01



The Dilemma of Reducing Board Size

Many tiny ESP32-C3 and -S3 boards on the market prioritize compactness over proper Wi-Fi antenna design. This may be acceptable for applications without Wi-Fi or with very short range to the access point. However, poor antenna design often reflects signal energy back into the chip, triggering internal protection circuits and reducing transmission power.

My additional tests with two ESP32-S3-Zero boards (see photo) confirmed this [6].

A thermal camera revealed that the original board heated up significantly more than my modified version when running my Wi-Fi test program. Additionally, I observed a much warmer voltage regulator IC and a 3.3-V supply voltage with occasional short voltage drops.

My antenna modification — taking into account the different antenna orientation — brought these issues back to safe levels. Modules that were initially unstable and prone to rebooting became stable and reliable after adding the wire antenna, improving Wi-Fi signal strength by over 6 dB.

Questions or Comments?

Do you have technical questions or comments about this article? Then please contact the author by email at peter.neufeld@gmx.de or the Elektor editorial team at editor@elektor.com.

About the Author

For more than 40 years, Peter Neufeld was allowed to take care of other people's data processing, electrical, and building control systems, as well as communication and media technology. Now he finds time again to devote himself to the practical application of classical electronics and modern microcontrollers in hobby projects. He has described some of these projects on Elektor Labs or on his blog.

Related Product

- **Siglent SSA3032X Plus Spectrum Analyzer (9 kHz – 3.2 GHz)**
  www.elektor.com/20635

WEB LINKS

[1] Software download at Elektor Labs: https://www.elektormagazine.com/labs/esp32-c3-supermini-antenna-mod
[2] Annex RDS Useful Links, CiccioCB.com: https://cicciocb.com/links/
[3] Annex ESP32 installer: https://flasher.cicciocb.com/dist/index.html
[4] CA-C03 Specifications, Cross Air: https://fcc.report/FCC-ID/2ASYE-T-ECHO/6227962.pdf
[5] "ESP32-C3 Supermini Antenna Modification," Website of the Author: https://peterneufeld.wordpress.com/2025/03/04/esp32-c3-supermini-antenna-modification/
[6] "ESP32-S3-Zero Antenna Mod," Elektor Labs: https://www.elektormagazine.com/labs/esp32-s3-zero-antenna-mod

ZD-8968 Hot-Air Soldering Station

A Budget-Friendly Workhorse or Just Hot Air?

By Clemens Valens (Elektor)

Looking for an affordable hot-air soldering station with helping hands? The ZD-8968 KIT combines precision temperature control with a flexible workspace — but does it deliver on usability and performance? Let's find out.

The ZD-8968 KIT [1] combines a hot-air soldering/desoldering station with a set of soldering helping hands in a single package. Inside the box, along with the helping hands, you'll find the hot-air station with the gun attached. The base station has a glossy black carbon finish, resembling the ZD-8965 desoldering station [2], which I reviewed some time ago. While lightweight, it remains stable and doesn't slide when pressing its buttons.

The hot-air gun is connected to the base station via a one-meter-long hose (**Figure 1**). It appears to be a standard hot-air gun, identical to the one used in the VTSS230 rework station [3], which I also reviewed. A standalone metal holder for the gun is included, though it feels too light for my taste — you often need both hands to remove the gun from it. The gun comes with three nozzles: 5 mm, 9 mm, and 12 mm in diameter.

Figure 1: The ZD-8968 hot-air rework station with helping hands. (The separate stand is not shown here.)

The ZD-8968 includes two power cords: one with a European right-angle plug and another with a UK-standard plug.

The printed manual (English only) consists of three pages printed in tiny characters on a single A4 sheet.

Features and Performance

The ZD-8968 hot-air station is rated at 300 W and can heat from 100°C to 500°C (212°F to 932°F). A push button allows switching between °C and °F. According to the manual, it features accurate and stable temperature control. Indeed, after turning the device on and picking up the gun, the temperature begins to rise and responds well to adjustments. Once the target temperature has been reached, it remains stable.

The display shows the actual temperature in light-blue 7-segment characters. When you press one of the *TEMP* buttons ("+" and "–"), the target temperature is displayed. However, since the device is responsive, the set temperature and actual temperature will usually be the same.

The station has one programmable temperature preset, which should be sufficient for most users. Setting the temperature is quick and easy — when holding down a *TEMP* button, the value increases in 10° increments, making adjustments in °C mode faster than in °F mode.

The fan speed is adjustable from "Lo" to "Hi" in 11 steps using two dedicated push buttons ("+" and "–").

As mentioned earlier, the gun responds to movement. When picked up, it starts heating, and the display shows "Heat On." When placed in its metal stand (pointing upward), sleep mode activates.

The display shows "Sleep." In this mode, the fan continues running until the gun cools to approx. 80°C, at which point the fan turns off, restoring quiet to your lab. A spinning icon on the display indicates fan activity.

To give an idea of the heat-up speed, I tested the system in sleep mode with the target temperature set to 500°C and the fan at its lowest speed. The display initially showed 56°C. After removing the gun from its stand, it took 106 seconds to reach 500°C. After placing it back in the stand, it took a little over four minutes to cool down to 80°C, at which point the fan shut off. The ambient temperature was 18°C.

Soldering Helping Hands (ZD-11P)

Now, let's take a look at the ZD-11P soldering helping hands, which complete the kit. The setup consists of a metal baseplate with slots on three sides, allowing up to four flexible gooseneck arms with alligator clips to be mounted, as shown in **Figure 2**. The clips are covered with heat-shrink tubing, preventing them from biting too hard into a board.

A fifth flexible arm is included to hold a flashlight (included, but the required 1.5 V AA battery is not). A metal rod with an articulated clamp is also provided to secure the hot-air gun.

The base measures 210 × 133 mm, but the usable workspace is smaller when the gun is clamped in place. The effective working area shrinks to about 160 × 100 mm, and the vertically mounted gun doesn't fully cover the area. However, it can be tilted, although moving the workpiece may be easier.

The flexible arms are 20-cm long and can hold relatively large workpieces. That said, larger boards are best placed directly on the bench. The hot-air gun can be rotated away from the base, so boards don't necessarily need to be placed on the stand for the helping hands to be useful.

The assembly is very stable — it doesn't slide easily or tip over, as long as the gun hovers over the base and not behind it. The arms are easily repositioned, and while the included flashlight feels flimsy, it is still practical.

One issue to note: when the gun is clamped in place, it will not enter sleep mode, even if positioned upright. A spring-loaded clamp would have been more convenient than the included screw clamp, allowing quick removal of the gun so it could be placed in its stand for sleep mode activation.

Final Thoughts

Overall, the ZD-8968 KIT offers a functional hot-air station with responsive temperature control and a helping hands setup that can handle small to medium workpieces. While some minor design choices — such as the lightweight gun holder and lack of auto-sleep in clamp mode — could be improved, the kit remains a solid option for those looking for an affordable hot-air soldering solution.

Would I recommend it? For occasional use, definitely. But for heavy-duty work, a more robust hot-air station and higher-quality helping hands might be a better investment.

250383-01

Figure 2: A flashlight helps to see what you are doing.

WEB LINKS

[1] ZD-8968 KIT: https://www.elektor.com/products/zd-8968-kit-hot-air-rework-station
[2] Clemens Valens, "Zhongdi ZD-8965 Desoldering Rework Station (Review)," elektormagazine.com, July 2024: https://tinyurl.com/review-zd-8965
[3] Clemens Valens, "Hot Tool: Velleman VTSS230 2-in-1 SMD Hot Air Rework Station," elektormagazine.com, July 2021: https://tinyurl.com/velleman-VTSS230

31 Parking Sensor Tester

Finding Defects in the PDC System of a Car

By Alfred Rosenkränzer (Germany)

A defective parking sensor can be expensive if you don't notice it in time and get it repaired. Following the engineer's ironclad rule "Do it yourself," here is a small circuit that you can use to quickly and easily detect defective parking sensors.

A few weeks ago, our son came to visit. On this occasion, he mentioned that the parking assistance at the back of his car probably no longer worked. Research on the Internet revealed that the on-board computer of his car initially checks the sensors every time it is put into reverse. As soon as a sensor is defective, there is a long warning tone and the entire parking assistance is deactivated.

Bat Detector

I thought for a while about how to check the sensors. Then I remembered the prototype of a bat detector that I had built for fun some time ago. **Figure 1** shows its block diagram. The detector is logically constructed: a microphone that is also sensitive to ultrasonic signals is followed by an amplifier with an automatic level control. After the signal bandwidth has been limited by the series connection of a high- and low-pass filter, a further amplification with a second level control is carried out.

A suitable microphone of the type SPU0410HR5H is available from the usual distributors or at a reasonable price on eBay. The second block from the left in Figure 1 (preamp/AGC/limiter) is a MAX9814 on a small circuit board, which is also available ready-made for a few euros on eBay. The original microphone on this module has been removed. The high-pass and low-pass filters are built on a prototype board. The signal was routed from directly after the low-pass filter cable to the input of my oscilloscope using a coaxial cable. The output amplifier of the bat detector with an AGC and a limiter based on an SSM2167 was not used here.

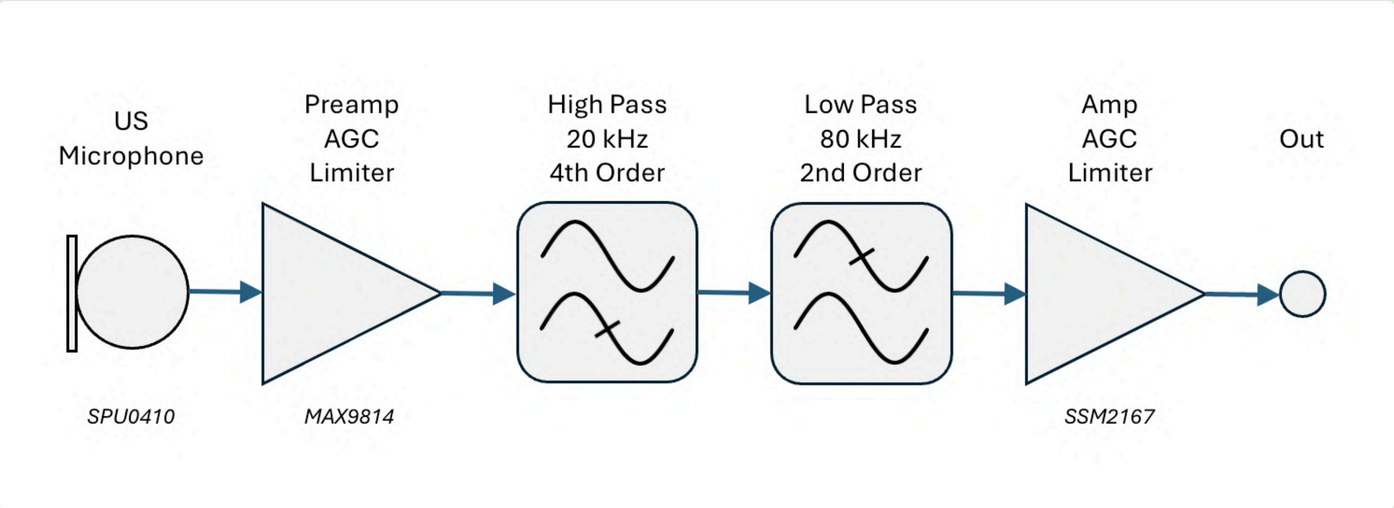


Figure 1: The block diagram of the bat detector. The last module on the far right is not needed here.

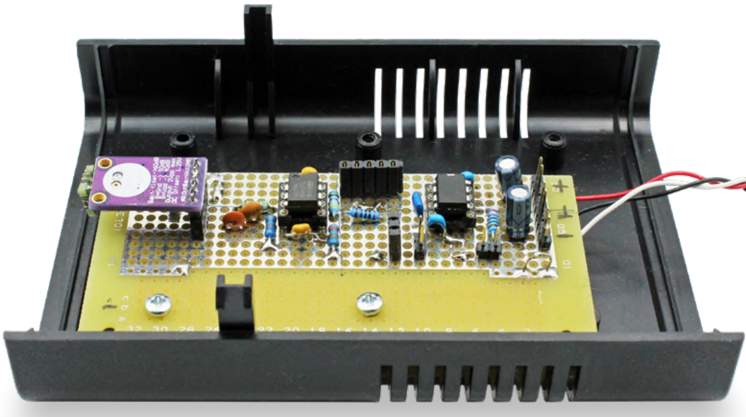


Figure 2: The prototype of my repurposed bat detector built into a plastic casing.

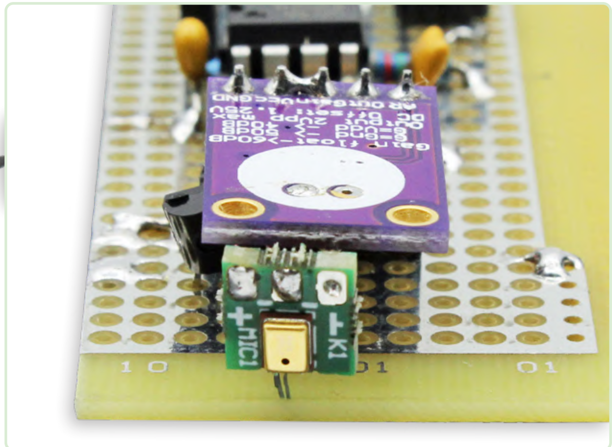


Figure 3: The SPU0410HR5H microphone connected to the MAX9814 module from Adafruit.

Prototype

Figure 2 shows my prototype in its plastic casing. **Figure 3** shows how I connected the new microphone to the module board of the MAX9814. The real measurement in the car is shown in **Figure 4**. **Figure 5** shows the oscillogram of the measured signal of a functioning sensor.

Figure 4: Testing the parking sensors on my son's car. This one worked fine, as you can see on the oscilloscope.

Component List

Resistors

R1 = 1 k
R2 = 1k8
R3, R6 = 8k2
R4 = 6k8
R5, R7 = 10 k
R9 = 220 Ω

Capacitors

C1, C8...C12 = 100 n
C2 = 1n5
C3 = 820 p
C4 = 1 n
C5 = 2n2
C6, C7 = 2n7
C13, C14 = 22 µ / 16 V, Elko

Semiconductors

IC1, IC2= TL072
IC3 = 78L05
M1 = Electret microphone amplifier MAX9814, module, Adafruit

Miscellaneous

Microphone SPU0410HR5H
Prototyping board

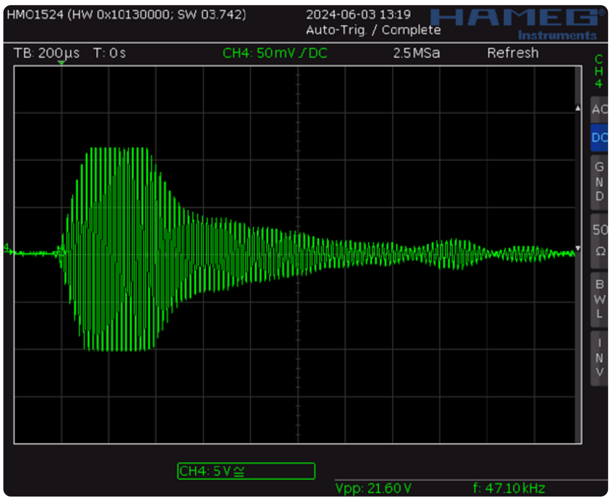


Figure 5: Screenshot of my oscilloscope. This is what the signal from a functioning parking sensor looks like.

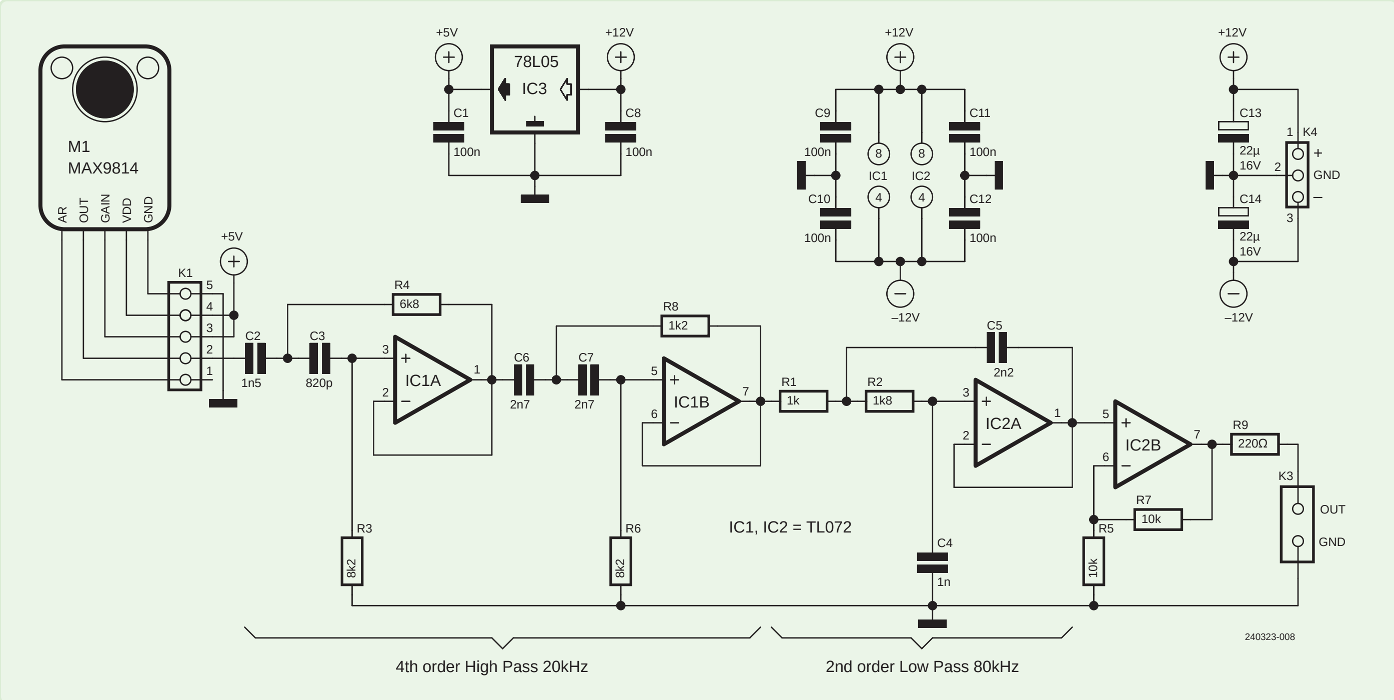


Figure 6: Schematic of the adapted version of my bat detector without the superfluous output stage of Figure 1.

I redrew the circuit diagram (**Figure 6**) without the second amplifier of Figure 1 (SSM2167). IC2B serves here as a dual-amplifying buffer for the output.

Since the circuit is simple, and I had quickly built it on a prototyping board, there was no time to design a circuit board. However, there is nothing to stop you from doing so and, if necessary, turning it into a compact device — possibly with a small oscilloscope built into the case. Ideally, the power should then be supplied via the car battery.

You can test the circuit with a function generator and a 40-kHz ultrasonic transmitter or a suitable tweeter. An alternative signal source is, for example, the inexpensive Grove ultrasonic distance sensor from Seeed Studio, which is available from Elektor. In the case of our son's car, it was very easy to find the fault using this test circuit: One sensor was no longer emitting an ultrasonic signal.

Translated by Jörg Starkmuth — 240323-01

About the Author

Alfred Rosenkränzer worked for many years as a development engineer, initially in the field of professional television technology. Since the late 1990s, he has been developing digital high-speed and analog circuits for IC testers. Audio is his private passion.

Questions or Comments?

Do you have questions or comments about this article? Email the author at alfred_rosenkraenzer@gmx.de, or contact Elektor at editor@elektor.com.

Related Products

- **DIY Mini Digital Oscilloscope Kit**
  www.elektor.com/20938
- **Seeed Studio Grove Ultrasonic Distance Sensor**
  www.elektor.com/20027