£9.95 | €10.95 | $12.95

elektor MAG

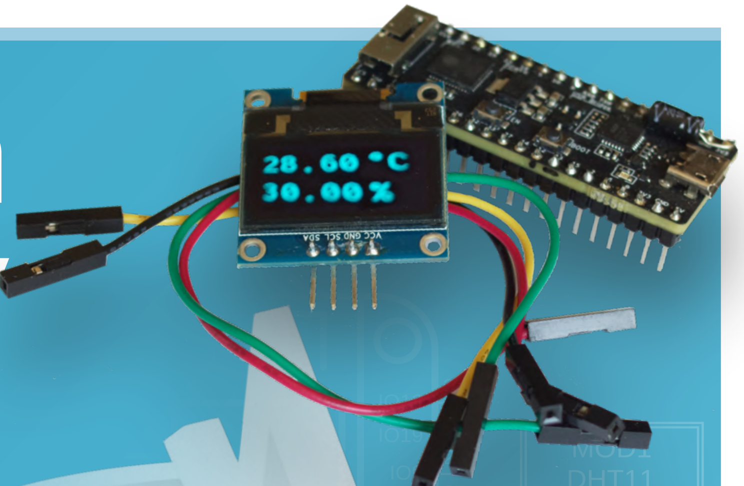
design > share > earn

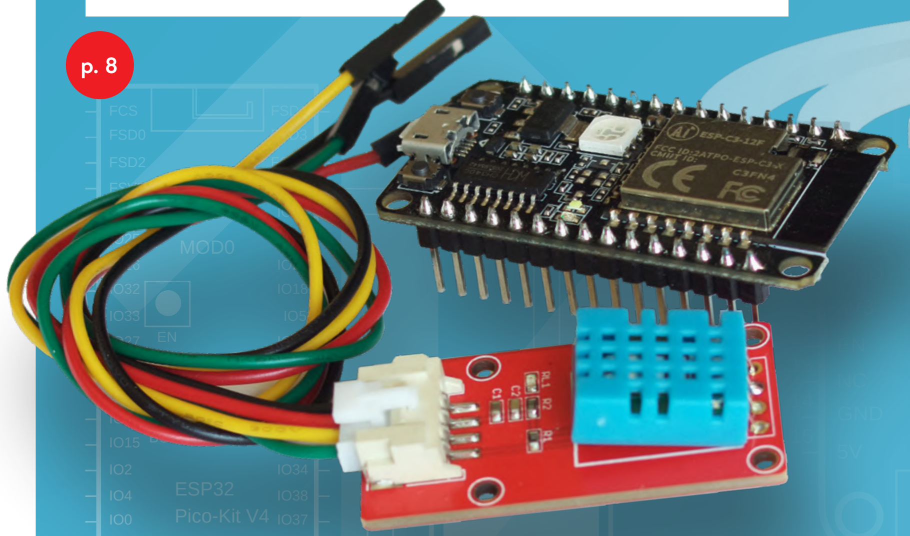
516 * SINCE 1961 *
SEPTEMBER & OCTOBER 2022
ELEKTORMAGAZINE.COM

Bluetooth Low Energy

with ESP32-C3 and ESP32

p. 8





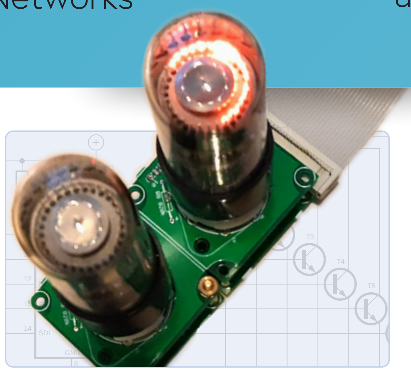
p. 12 **Bluetooth Low Energy Sniffer**
Hacking a makerdiary nRF52840 MDK USB Dongle

FOCUS ON
Wireless Applications

p. 58 **5G Just for Me**
Complete Control of 5G Deployments with Private Cellular Networks

p. 77 **Sending Data to Telegram**
Get It Done with an ESP32 and a Few Parts

Bicycle Electrification Hands-On with an E-Bike Retrofit Kit p. 26



Audio Spectrum Analyzer with Dekatrons A New Way to Use Vintage Tubes p. 70

Digital FM Receiver with Arduino and TEA5767 Stay Tuned with an Arduino Nano p. 100

COLOPHON

Elektor Magazine,
English edition

Edition 7/2022
Volume 48, No. 516
September & October 2022

ISSN 1757-0875 (UK / US / ROW distribution)

www.elektor.com
www.elektormagazine.com

Elektor Magazine, English edition
is published 8 times a year by
Elektor International Media

Head Office:
Elektor International Media b.v.
PO Box 11
6114 JG Susteren
The Netherlands
Phone: (+31) 46 4389444

Memberships:
E-mail: service@elektor.com
www.elektor.com/memberships

Advertising & Sponsoring:
Raoul Morreau
Phone: +31 (0)6 4403 9907
E-mail: raoul.morreau@elektor.com

www.elektor.com/advertising
Advertising rates and terms available on request.

EDITORIAL

Jens Nickel

International Editor-in-Chief, Elektor Magazine

Wireless in Practice

The future is wireless, and a nice personal coincidence is that I've just upgraded my hobby to "wireless" as well. When my friends and I record DJ videos outside, we transmit the music from the DJ booth to the video mixer over the airwaves; and if there's live streaming, it goes to the net with 4G. For audio transmission, I opted for a system that is also used for "silent disco" parties. One transmitter, many receivers, latencies below the perception threshold — what more could you want? So that an audience can also enjoy the DJ set, I can take loudspeakers into the green, each equipped with its own receiver and battery (for the smaller ones, USB powerbanks are enough). The whole thing is really only limited by my wallet and the Cologne regulatory office. No more plugging and unplugging, no more Y-adapters, and no more stumbling over cables. I find "wireless" really very practical!
In this issue, we're looking at wireless data transmission — via Bluetooth LE, among other things. In the cover story, my colleague Mathias Claußen shows how easy it is to transmit sensor values from an ESP32-C3 to an ESP32 without using Wi-Fi. With the right libraries, programming a small BLE application is not witchcraft. In this issue, you will also find a BLE sniffer and a small project that transmits measurement data to the Telegram messaging service. And in our Industry section, Stuart Cording looks at privately operated 5G networks.
For more on wireless, check out our webinar on October 13 and, of course, watch our regular Elektor Lab Talk (see page 24 for details)!

The Team

International Editor-in-Chief:	Jens Nickel
Content Director:	C. J. Abate
International Editorial Staff:	Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf, Alina Neacsu, Dr Thomas Scherer
Laboratory Staff:	Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens
Graphic Design & Prepress:	Giel Dols, Harmen Heida
Publisher:	Erik Jansen

FIPP Connecting Global Media
Elektor is a member of FIPP, an organization that has "grown over almost 100 years to include media owners and content creators from across the world."

DEUTSCHE FACHPRESSE
Elektor is a member of VDZ (Association of German Magazine Publishers), which "represents the common interests of 500 German Consumer and B2B publishers."

THIS EDITION

Volume 48, No. 516
September & October 2022

Regulars

Features

Industry

Audio Spectrum Analyzer with Dekatrons

70

CO_2 Meter Teardown

Is It Hackable for Your Projects?

85

Projects

Next Edition

Elektor Magazine Edition 11-12/2022
(November & December 2022)
As usual, we'll have an exciting mix of projects, circuits, fundamentals and tips and tricks for electronics engineers and makers. We will focus on Production and Components.

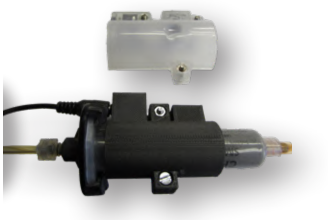
From the contents:

- High-End Amplifier Fortissimo-100
- DAB+ Radio
- Door Control with Bluetooth
- FreeRTOS driving Christmas Garlands
- PCB Design: Tips and Tricks
- Air Quality Monitor with RP2040
- Chip Whisperer
- Design Tools for Analog Filters

And much more!
Elektor Magazine edition 11-12/2022 (November & December 2022) will be published around November 10, 2022. The arrival of printed copies for Elektor Gold Members is subject to transport. Contents and article titles are subject to change.

Auto On/Off
for Solder Paste Compressor

21



E-FFWD

electronica fast forward 2022 Start- & Scale-Up Awards

November 15 - 18, 2022,
electronica, Munich

Preparations Speeding Up!

By Udo Bormann and Erik Jansen (Elektor)

After a period of internal deliberation in June, we started contacting the shortlist of candidate start-ups, scale-ups, and other innovators for the 2022 edition of the electronica fast forward Awards. And now, a jury of experts has selected nine of the most interesting innovations of the year — from solar energy via robotics to electric light vehicles.

As we announced earlier [1], the 2022 edition of the electronica fast forward Start- & Scale-Up Awards promises to be an extra-special show. With 273 m² in one of the best spots of the world's biggest event [2] for the electronics industry in Munich, it can hardly be any other way.

Selecting Start-Ups and Scale-Ups

Based on submissions and the opinions of industry connoisseurs, our initial long list has been evaluated by our jury. The jury is consisting of one electronics engineer, one editor, one marketeer, and one business expert. They have ranked each start-up and scale-up following our well-proven ranking system, specifically developed for this purpose. The first companies/institutions have already been contacted and confirmed.

In the following sections, we present to you the first overview of this year's selected companies. Each one of them is either a company or institution that is at the beginning or is already on the road to success. Innovative entrepreneurial "forces of nature" are hard at work, changing their segment forever or breaking open a whole new segment themselves. This is what Elektor and Messe München have been striving for since the first edition of the fast forward Awards in 2016 [3]: to provide a stage for disruptive young entrepreneurs and to introduce them to the key players in our industry. The electronica trade fair in Munich is home to almost every major player in the electronics industry, so you can't wish for a better launch for yourself as an ambitious new player.

Of course, beyond the stage and network of electronica 2022, there is still much for participants to gain. This year Elektor is, once again, making a marketing budget of no less than €150,000 available to the winners.

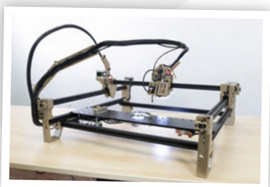
"It's always a great and exciting moment to see and hear the first reactions from the selected companies. Some are immediately thrilled, others can't quite believe it yet, but all are obviously proud to be one of the first selections made by the jury team. This year e-ffwd is looking a little wider than just the early-stage start-up and is also inviting scale-ups that will certainly improve the appeal of the Fast Forwards booth."

Udo Bormann,
Event manager e-ffwd, Elektor

The selected companies

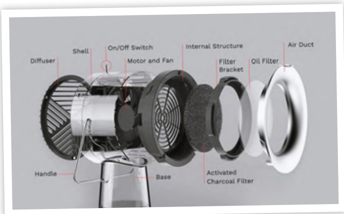
Wheel.me is the world's first autonomous wheel that comprises a robotic component, indoor navigation technology, and data analytics, enabling you to make everything and anything move effortlessly.
https://wheel.me

The mission of **Energy Robotics** is to relieve humans from dangerous, repetitive, and undesirable tasks through autonomous robotics. They apply the newest and most efficient intelligent automated software to robots in order to meet that goal. They are engineers at heart and always striving to improve their robotics software.
www.energy-robotics.com



The goal of the LumenPnP, and future **Opulo** machines, is to help folks bridge the gap between prototyping and mass production.
https://opulo.io

Treesense enables people to better understand the world of trees. They do their utmost to ensure that people can use the ecosystem's finite resources sustainably. These include, above all, water, clean air, soil fertility, climate, biotope and species protection, and especially the recreational services of the forest. They research and develop technologies based on the latest scientific findings to consider trees a central part of our ecosystem.
https://treesense.net



Airhood is the world's first portable range hood to remove grease & smoke, cooking odors, and oil film formation.
https://the-airhood.myshopify.com

V-Juice will be the smartest and overall best purchase charging device you will have. It is the thinnest and most discreet charging station that can be installed and reinstalled anywhere there is a surface.
www.v-juice.xyz

Include is a Croatian hardware company that became one of the best producers of smart street furniture in Europe. They are already the largest company in the production of solar-powered street benches in the world. Their smart bench is the most advanced street bench on the market. It provides cable and wireless charging for all smart devices, a free Wi-Fi hotspot, a streetlight in the evening, 15 sensors for tracking bench usage and environmental data, and a dashboard – an advanced web platform for real-time bench data tracking.
https://include.eu

The **Solar Team Eindhoven** strives for a sustainable future, where life is powered by the sun. As a student team, they have the power and freedom to be innovative. By creating innovative, energy-efficient solar vehicles, they inspire the current market and society to accelerate the transition to a more sustainable future - in both mobility and energy.
https://solarteameindhoven.nl

The goal of **ElectricBrands** is clearly defined - the XBUS will be the best and most innovative electric light vehicle in the world.
https://electricbrands.de

220386-01

It is never too late!

Are you a start-up, scale-up, or institution that would like to be considered for our upcoming events? Send an e-mail to **marketing@elektor.com** and let us know about your product/service/innovation! Each application will be individually weighed by our team!

WEB LINKS

[1] Erik Jansen, "E-FFWD: Looking Ahead Again!", ElektorMag 7-8/2022: https://www.elektormagazine.com/magazine/elektor-260/60568
[2] electronica Website: https://electronica.de/en/
[3] "And the Winner is...", Elektormagazine.com: https://www.elektormagazine.com/news/and-the-winner-is

Bluetooth Low Energy with ESP32-C3 and ESP32

You Don't Always Need to Choose Wi-Fi!

By Mathias Claußen (Elektor)

In contrast to the ESP8266, the ESP32-C3 is equipped with a Bluetooth Low Energy RF communication link. If you only need to send small amounts of data over short distances, this standard is an energy-saving alternative to Wi-Fi. We demonstrate this here with a small project: A temperature/humidity sensor with an ESP32-C3 transmits its data to an ESP32 and a small OLED display.

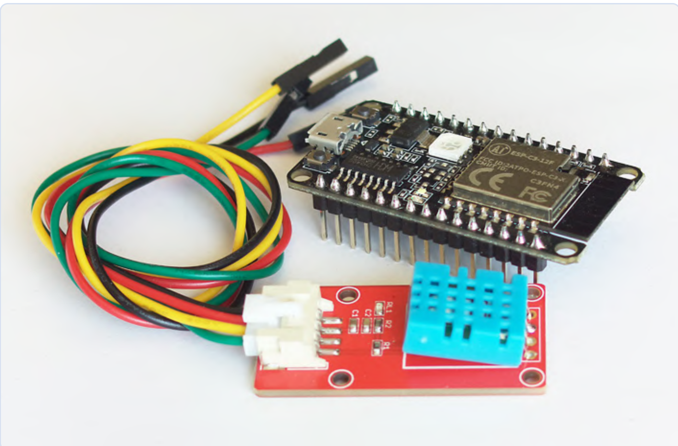


Figure 1: All you need to build the sensor node.

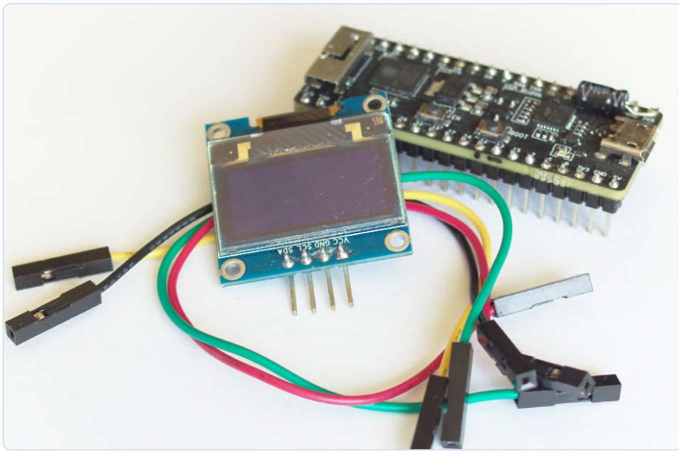


Figure 2: And for the display.

The ESP32-C3 [1] with its RISC-V core and its particularly good bang-for-buck ratio can be considered a successor to Espressif's ESP8266 MCU. One of the advantages that this new chip brings into the mix is its integrated Bluetooth Low Energy (BLE) communication capability. BLE allows data to be exchanged between end devices over a short-range and is very energy efficient. This communication standard is ideal for a wide range of applications, where small amounts of data need to be sent over short-range. Headphones, microphones, headsets or even watches use BLE to connect to various end devices (mostly smartphones).

But why BLE and not just Wi-Fi? When it comes to the periodic transmission of data over short distances, Wi-Fi is quite power-hungry. On top of this, Wi-Fi is intended for operation with an access point within an Ethernet network. By comparison, a small battery-powered ESP32-C3-based device communicating over BLE will achieve a much longer battery life.

This project will lead you through the first steps of using BLE communication. The setup uses a humidity and temperature sensor connected to an ESP32-C3 which sends the measured values to an ESP32 module where they are displayed on a small OLED display.

The Components

For this project, we have used standard components which can be ordered from the Elektor Store. They are nothing special, you may even

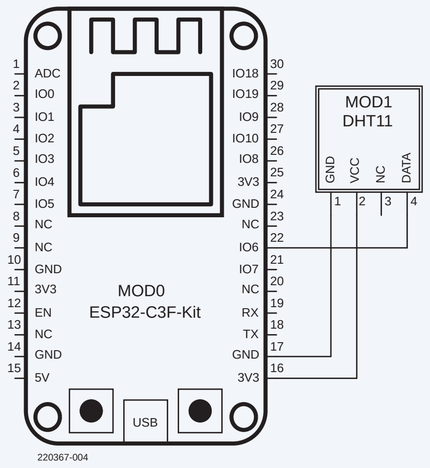


Figure 3: Sensor node circuit diagram.

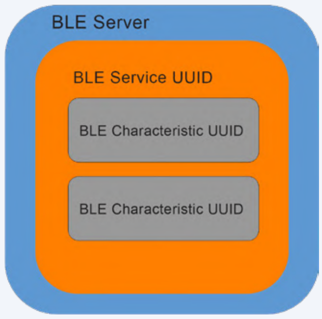


Figure 4: Structure of the BLE Servers and the UUIDs.

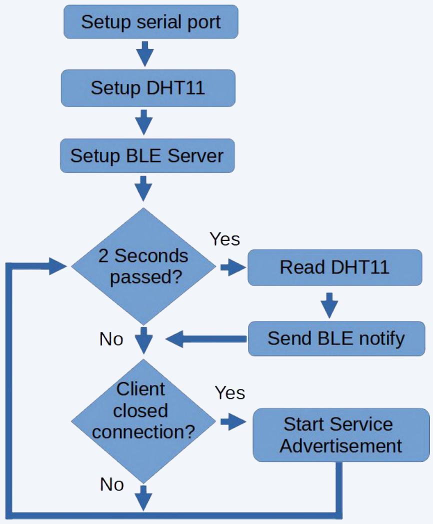


Figure 5: Sensor node flow chart.

have them already in your spare box. The sensor node only contains a DHT11 sensor which supplies both the temperature and humidity readings. It is contained, along with many other useful peripherals, in the Raspberry Pi Pico Experimenting Bundle. The ESP32-C3 controller comes in the form of the ESP32-C3-12F kit development board (both of which are available from the Elektor Store — see the textbox). All the components used are shown in **Figure 1**.

For the display, we use an ESP32-PICO-Kit V4 controller and a small 0.96" OLED display (see products textbox) together with just four jumper leads for wiring hookup. A WeMos LoLin ESP32 with an integrated OLED display could also be used, but the display pin assignments will need to be changed. The parts for the display unit can be seen in **Figure 2**.

BLE Data Packets

The communication protocol used to send data via Bluetooth Low Energy is not compatible with the earlier Bluetooth Classic protocol. With BLE there are basically servers and clients, both of which are able to exchange packets of data using Attribute Protocols (ATT) and Generic Attribution Profiles (GATT).

GATT provides a list of services and characteristics that contain procedures and attributes. For example, an attribute can represent a sensor value. Each of the attributes is addressed by a UUID, which can be assigned by the developer. The attributes are in turn packaged into services, one or more per server, which in turn also have a UUID. An example of service would be the provision of a data set containing sensor values (temperature, humidity, etc.).

With GATT, access authorization is per connection, i.e. no distinction is made as to which end device sets up the connection, as long as the parameters and keys allow the connection to be established.

This greatly simplified representation of GATT should be sufficient for this project. This is, after all, only intended as an introduction to working with BLE, so no security mechanisms are implemented here. More information about BLE can be found on the Bluetooth SIG page [2] or by watching the Elektor Webinar about BLE Android Apps [3]. Now we can get down to setting up our BLE server and BLE client.

BLE Server

Before we start with the software, let's take a closer look at the hardware. The connection of the DHT11 to the ESP32-C3 is shown in **Figure 3**. VCC connects to 3.3 V, GND to ground, and the sensor's DATA signal to IO06 of the ESP32-C3.

The structure of the BLE server can be seen in **Figure 4**. It is structured in layers, with the Server itself forming the outer layer. Following on from this comes the Services, which in this case is just one, with the UUID *91bad492-b950-4226-aa2b-4ede9fa42f59*. Contained in the Service are the characteristics that will be provided. One of these has the UUID *cba1d466-344c-4be3-ab3f-189f80dd7518* for the temperature in degrees Celsius (°C) and the other UUID *ca73b3ba-39f6-4ab3-91ae-186dc9577d99* is for the humidity values. Each of these Characteristics has a value and a description. This description again has a UUID and is set here to *0x2902* — where this value indicates that it is a characteristic description. If you want to learn more about this, refer to the "Bluetooth low energy characteristics, a beginner's tutorial" by Nordic Semiconductor.[4]

In our software, all parts of the BLE server are configured first. A new measured value is then read in by the DHT11 every two seconds and sent to connected devices as a BLE notification (*Notify*). This is a push message; the client does not need to confirm that the message has arrived. The software sequence can be seen in **Figure 5**.

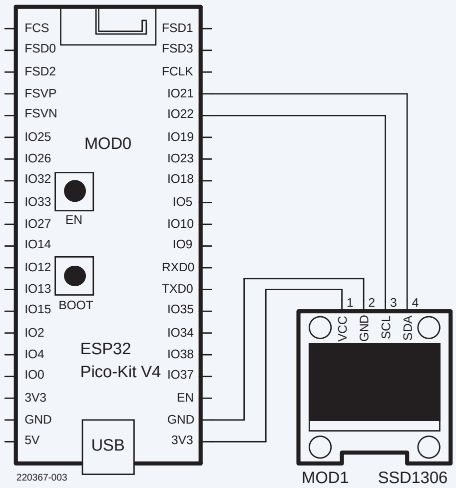


Figure 6: The display circuit diagram.

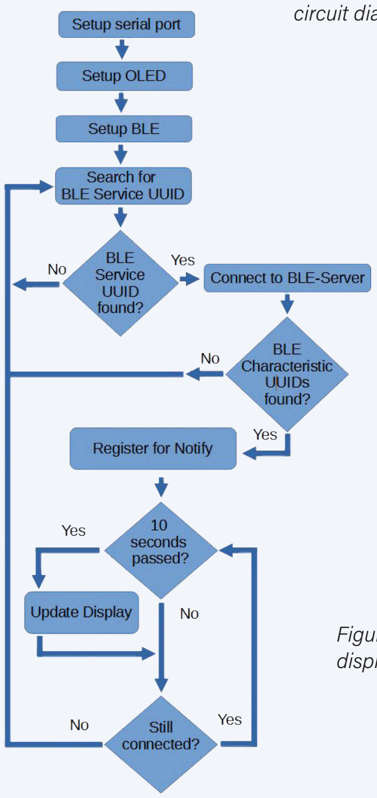


Figure 7: The display flow chart.

One thing that stands out in the code are the calls to `delay(5);` after the `notify();` calls. For example:

```
dht11HumidityCharacteristics.setValue(String(event.
   relative_humidity).c_str());
dht11HumidityCharacteristics.notify();
delay(5);
```

This is to prevent packet congestion from occurring within the BLE stack. Unfortunately, depending on the version of the Arduino environment for the ESP32 and ESP32-C3, this can still lead to the BLE stack unintentionally giving up the ghost.

Even after a `Disconnect`, we invoke a 500-ms delay period for the BLE stack processing. Incidentally, this (unfortunately) affects both the ESP32 and the ESP32-C3, since both use the same BLE stack by default.

An alternative to the default Bluetooth Classic and BLE stack (BlueDroid based Stack)[5] is the Apache MyNewt NimBLE [6] which can only be used for purely BLE (not Classic Bluetooth) applications in the Arduino framework [7]. We will now use this stack for the client unit, which in our case is made up of an ESP32 with an OLED display.

BLE Client

As with the server, we will first take a look at the client hardware, then the software. Here we have used an ESP32-PICO-Kit V4, on which an ESP32 module is fitted to provide the processing power. Even though their names are similar, these two modules are not closely related. The ESP32 has two Xtensa LX6 processor cores [8], whereas the ESP32-C3 uses a single RISC-V-based core (RV32IMC).

Only four wires are needed to connect the OLED display to the ESP32 PICO kit. The VCC of the display is connected to 3.3 V, and the GND of the display to the GND of the ESP32 PICO kit. Then we hook up the I²C connections, SDA and SCL. SDA connects to GPIO21 and SCL to GPIO22 of the ESP32-PICO kit. The circuit diagram can be seen in **Figure 6**.

A software flow diagram of the complete process is shown in **Figure 7**. After starting the ESP32, the serial interface and the OLED are initialized, followed by the BLE stack. Then begins the five-second search for new BLE servers.

If a server is found, the `onResult` function in the `Configured_AdvertisedDeviceCallbacks` class is called in response. Here it is determined whether the server offers a service with the UUID of *91bad492-b950-4226-aa2b-4ede9fa42f59*. If the server has a service with the appropriate UUID, the search for a new BLE server terminates and a connection to the found BLE server is established. After the connection is established, the server's service is checked for two UUIDs for the two characteristics *cba1d466-344c-4be3-ab3f-189f80dd7518* and *ca73b3ba-39f6-4ab3-91ae-186dc9577d99*. These are the two characteristics of UUIDs used by the server to identify the temperature and humidity readings. If the characteristics are both available, the connection is maintained. If no suitable BLE server or service is found, a new search will be initiated.

First, the client checks whether the two UUIDs for temperature and humidity can also send notifications (*Notifies*). The following code (the example here is just for the humidity readings) is used to achieve this:

```
pRemoteHumCharacteristic =
pRemoteService->getCharacteristic(humUUID);
…
    if (pRemoteHumCharacteristic != nullptr) {
      if(true==pRemoteHumCharacteristic->canNotify()){
        pRemoteHumCharacteristic->
        registerForNotify(NewHumNotify);
      }
…
}
```

If notifications can be sent, a callback is set up for them. Every time a new notification arrives, the `NewHumNotify` function is then called for the humidity. A notification can contain up to 20 bytes of user data, which can be freely assigned and of any chosen format. In our software, the BLE server sends the notification value as fully formatted readable strings. To show them on the display, they only have to be prepared appropriately.

As soon as there is a connection to the BLE server, the OLED display is simply updated every 10 s to show the latest values as they are received. The finished BLE server and BLE client can be seen in **Figure 8**.

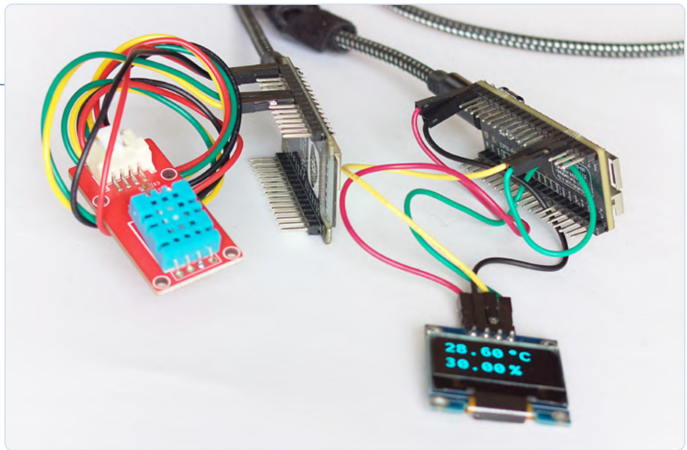


Figure 8: The BLE Server and Client working together.

BLE and the ESP32/ESP32-C3

The example BLE application described here is not the most sophisticated. You could imagine using BLE communication with an ESP32 or ESP32-C3. The intention is to demonstrate the principles involved so that you can become familiar with the basics and gain confidence to explore the topic further. BLE offers a power-saving option for short-range data transport that does not require the infrastructure of a Wi-Fi network.

220267-01

Questions or Comments?

Do you have any questions or comments relating to this article? Feel free to contact the author at mathias.claussen@elektor.com or contact the team at Elektor at editor@elektor.com.

WEB LINKS

[1] Mathias Claußen, "Getting Started with the ESP32-C3 RISC-V MCU": www.elektormagazine.com/news/getting-started-with-the-esp32-c3-riscv-mcu
[2] Bluetooth SIG: "Intro to Bluetooth GAP (GATT)": www.bluetooth.com/bluetooth-resources/intro-to-bluetooth-gap-gatt/
[3] C. Valens, "Rapid Prototyping Bluetooth Low Energy Android Apps Using MIT App Inventor," Elektor.TV, June 2021: www.youtube.com/watch?v=Jxv9h0nHlBA&t=2930s
[4] Nordic Semiconductor, "Bluetooth low energy Characteristics, a beginner's tutorial": https://bit.ly/3sCEVzR
[5] ESP32 Bluetooth Classic and BLE Stack: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/index.html
[6] ESP32 NimBLE Stack: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/nimble/index.html
[7] NimBLE-Arduino: https://github.com/h2zero/NimBLE-Arduino
[8] ESP32 LX6 Core: https://en.wikipedia.org/wiki/Tensilica

Bluetooth **Low Energy** Sniffer

Hacking a makerdiary nRF52840 MDK USB Dongle

Figure 1: The makerdiary nRF52840 MDK USB dongle.

Figure 2: The USB dongle.

Figure 3: With the cover off.

By Mathias Claußen (Elektor)

It's always useful when you can repurpose a kit so that it performs a completely different function than what it was designed for originally. Here we reflash the software of an evaluation board and turn it into a useful BLE data traffic sniffer — a handy tool for anyone developing and testing Bluetooth applications. The evaluation board in question is the low-cost nRF52840 USB dongle for IoT development from makerdiary.

Anyone developing Bluetooth Low Energy (BLE) applications would find it useful to be able to view data packet exchange in real time between devices to help with debugging. As with the recording of Wi-Fi packets, suitable hardware for BLE will be required. We can use an off-the-shelf, low-cost evaluation board based on the Nordic nRF52840 Bluetooth SoC which has all the necessary communication peripherals built in. Two examples are the makerdiary nRF52840 MDK USB dongle [1] (**Figure 1**) and the Nordic nRF52840 dongle [2]. This SoC is also used as the primary processing component in a number of other boards, such as the Arduino Nano 33 BLE [3], the BBC micro:bit V2 (here the nRF52833 with less memory is used)[4], and the Adafruit CLUE [5].

In addition to the hardware module which will handle the BLE packets, we also need some software and a PC. For the computer we could use a standard PC based on an AMD64/x86-type processor or a Raspberry Pi. The software used here is Wireshark, a tool that some of you will probably already be familiar with. Altogether it builds a system to record and display the transfer of BLE packets in real time.

Step by Step

The installation and setup procedure described here applies to an AMD64/x86 PC running Windows 10. We also make use of the makerdiary nRF52840 MDK USB dongle. This small board is actually intended as a development kit for the Nordic nRF52840 and comes in neat dongle housing (**Figure 2** and **Figure 3**) which plugs straight into the PC's USB port. In addition to BLE 5.0 and Bluetooth Mesh, the chip also supports ZigBee and Thread networking protocols. The technical data can be found in **Table 1**. As well as its ability to record BLE packets, the dongle could also be configured to do a similar job for other wireless transmission standards.

From the factory, the makerdiary nRF52840 MDK USB dongle is equipped with the OpenThread Network Co-Processor (NCP) firmware. For our application we will be using BLE communications instead of Thread so we need to replace firmware and possibly the bootloader (depending on which version of the board you have). Updating the bootloader from Open-Bootloader to the UF2 bootloader later makes programming the makerdiary nRF52840 MDK USB dongle really easy because it is recognised as a USB mass storage device in the system (in the same way a Raspberry Pi Pico does). If you want to swap out the firmware again later, for example with CircuitPython [6], you can do that very easily.

Update Using the uf2 Bootloader

A few tools are required to update the bootloader. Here we use *nrfutil* [7] to update it [8]. The *nrfutil* and updated bootloader files should be copied into one folder (do not unzip the zipped files).

Now it is necessary to put the makerdiary nRF52840 MDK USB dongle into bootloader

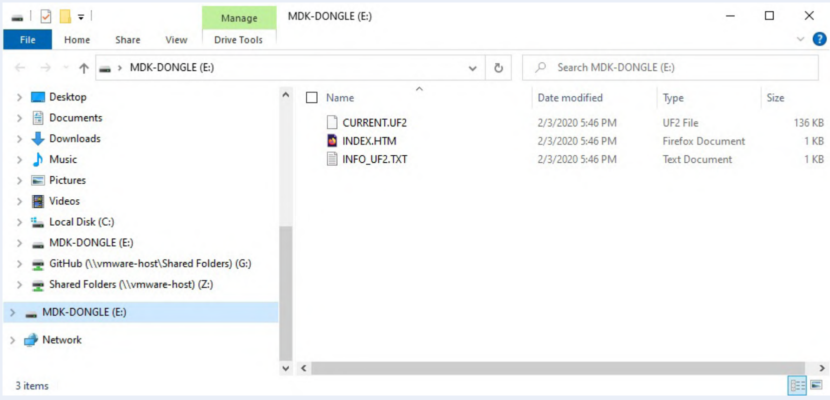


Figure 4: The nRF52840 is recognised as an external mass memory device.

Table 1: makerdiary nRF52840 MDK USB Dongle.

- Nordic nRF52840 System-on-Chip
- ARM Cortex M4F
- Optimised for Ultra-Low-Power
- Bluetooth 5, Bluetooth Mesh
- Thread, IEEE 802.15.4, ANT
- On-chip NFC-A-Tag
- On-chip USB 2.0 (Full speed) Controller
- ARM TrustZone Cryptocell 310 Security Subsystem
- 1 MB FLASH
- 256 kB RAM
- Up to 12 GPIOs
- Pushbutton and RGB LED
- On-board 2.4G-Antenna
- 3.3 V regulator 1 A max output
- VBUS & VIN Power Path Management
- Handy USB Dongle form factor
- Breadboard-friendly with dual 10-way headers

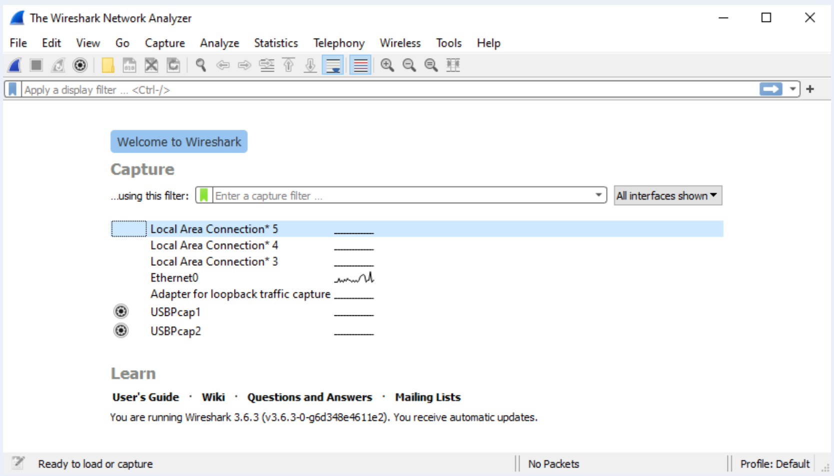


Figure 5: The Wireshark GUI.

mode. To do this hold down the reset/user button before the dongle is plugged into the PC's USB port. If the LED starts flashing red, the USB dongle is in bootloader mode and a new serial port should have been identified by the computer. Now you need to open a terminal window/command prompt so that you can navigate to the folder where the *nrfutil* and the new bootloader files have been saved. The following command line now needs to be entered here:

```
nrfutil dfu usb-serial -pkg uf2_bootloader-0.2.13-44-gb2b4284-nosd_signed.zip -p <serial-port>
```

`<serial-port>` is the new serial port number allocated to the makerdiary nRF52840 MDK USB dongle. Once this has been completed the USB dongle starts and a new mass storage drive is identified (**Figure 4**).

The BLE Sniffer Firmware

Installing the firmware for the BLE sniffer is quite straightforward. First, we need to download the appropriate firmware [9], which has a *.uf2* file extension from the Adafruit repository. Then we copy this to the dongle which has now been recognised as a mass storage drive. Now after reboot the makerdiary nRF52840 MDK USB dongle will fire up running the Bluetooth LE sniffer firmware and will monitor BLE data packet exchanges. Only one more software link in the chain remains to be added.

Wireshark and Python 3

In preparation, Wireshark [10] (**Figure 5**) and Python 3 [11] must first be installed on a PC. When installing Python 3, care should be taken that the environment variables are correctly registered (**Figure 6**) and that the Python launcher (**Figure 7**) is also available. Once these are installed, we need to install *pyserial* which allows Python applications to access the system's serial ports. For this we need to open a command prompt and type `pip install pyserial` (**Figure 8**).

Wireshark cannot communicate with the BLE sniffer firmware by default so it is necessary to install an extension. For this it is necessary to download the *nrf_sniffer_for_bluetooth_le_4.1.0.zip* [12] (or newer build) from Nordic Semiconductor. The *extcap* folder can be found in this zip file (**Figure 9**). In the Wireshark installation folder (under Windows this usually C:\

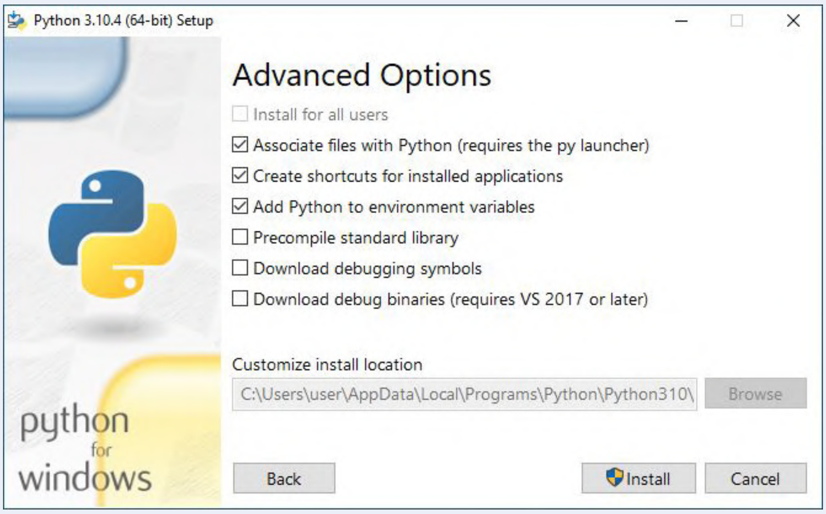


Figure 6: Select advanced Options for Python.

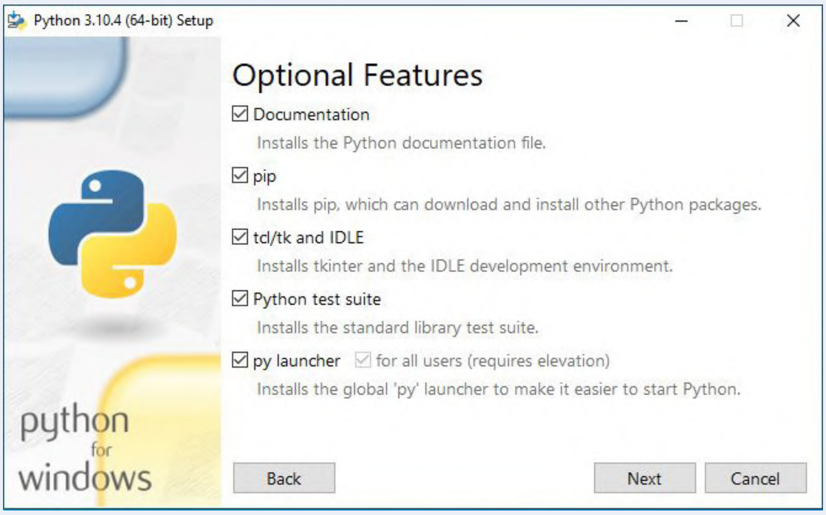


Figure 7: The Python launcher option needs to be selected.

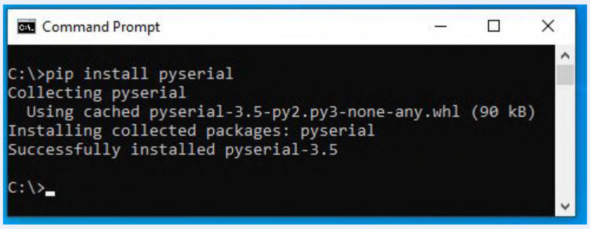


Figure 8: Installation of pyserial.

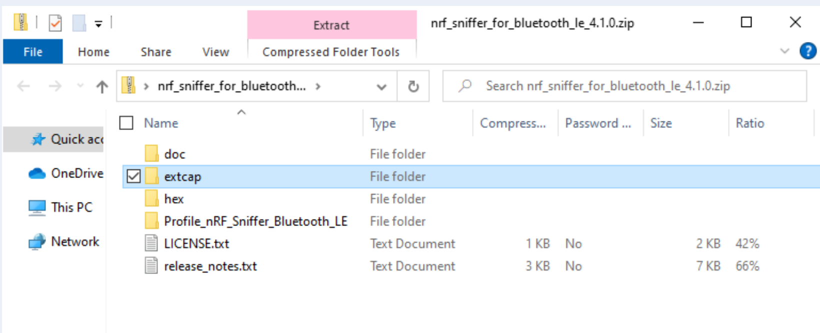


Figure 9: The extcap folder in Zip.

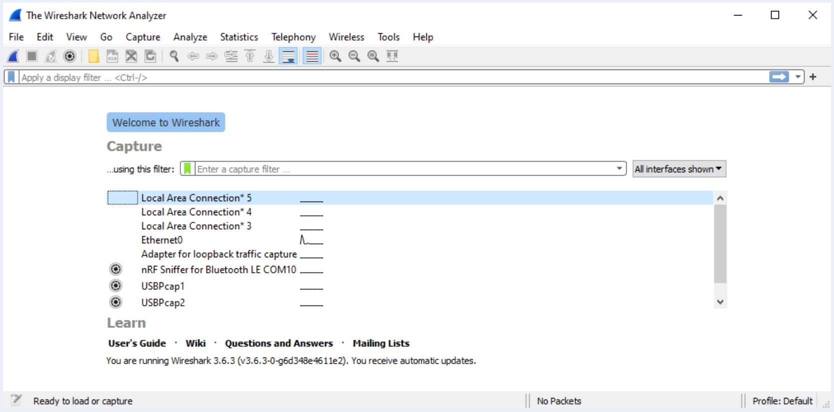


Figure 10: A new interface connection in Wireshark.

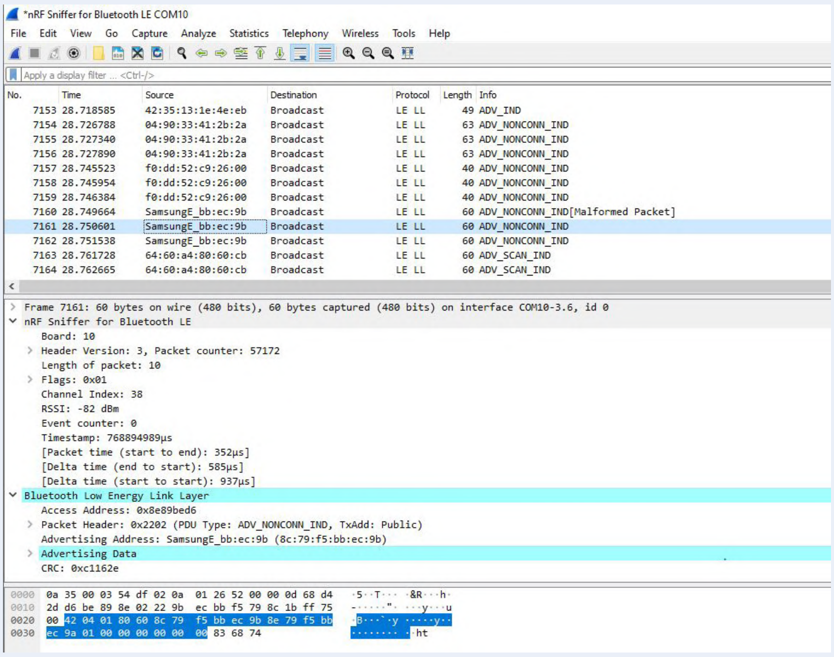


Figure 11: Display showing the BLE data packets.

WEB LINKS

[1] makerdiary nRF52840 MDK USB-Dongle: https://wiki.makerdiary.com/nrf52840-mdk-usb-dongle/

[2] nRF52840 Dongle: www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle

[3] Arduino Nano 33 BLE: https://docs.arduino.cc/hardware/nano-33-ble

[4] BBC micro:bit V2: https://microbit.org/new-microbit/

[5] T. Hanna, "CLUE from Adafruit: A Smart Solution for IoT Projects," Elektormagazine.com: www.elektormagazine.com/clue-adafruit-iot

[6] CircuitPython : https://circuitpython.org/board/makerdiary_nrf52840_mdk_usb_dongle/

[7] Nordic nrfutil: https://github.com/NordicSemiconductor/pc-nrfutil/releases

[8] UF2 Bootloader: https://bit.ly/3atr9JI

[9] BLE Sniffer Firmware: https://bit.ly/3LTMEQP

[10] Wireshark Homepage: www.wireshark.org/

[11] Python Homepage: www.python.org/

[12] Wireshark Interface in nrf_sniffer_for_bluetooth_le_4.1.0.zip: https://bit.ly/3Gq0yZQ

Program Files\Wireshark), you need to create an *extcap* folder, into which you need to copy the contents of the *extcap* folder from the zip file. Now we have everything we need to start listening to BLE packets.

When Wireshark is now started, another interface can be seen (**Figure 10**), which is called *nRF Sniffer for Bluetooth LE COMxx*, where xx indicates the number of the Com port used by the makerdiary nRF52840 MDK USB dongle. To record packets, just select that interface and start recording. If there are any BLE devices nearby, Wireshark will now start receiving data (**Figure 11**).

A Handy BLE Sniffer

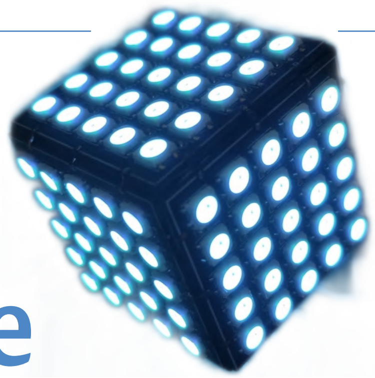
With a few simple steps, the makerdiary nRF52840 MDK USB dongle can be turned into a really useful BLE sniffer. Its now an invaluable tool for developers working on BLE applications especially when setting up ESP32 and Co. Not only can you confirm that data is being exchanged between devices, but with Wireshark, you can actually read the packet contents. In addition to BLE communications, the same dongle can also be reconfigured to work with a number of other standard communication protocols.

220248-01

Questions or Comments?

Do you have any questions or comments about this article? Email the author at mathias.claussen@elektor.com or contact Elektor at editor@elektor.com. You also can watch Mathias on the monthly Elektor Lab Talk (www.elektormagazine.com/elt) livestream on YouTube, where can ask your questions live!

Magic RGB LED Cube



Hardware Design Around an RP2040

By Mohamed Belkhir Sayari (Tunisia)

Want an eye-catching RGB LED Cube for your desk or workbench? Would you enjoy building your own "magic" cube based on a circuit featuring a Raspberry Pi RP2040? If so, this 6 sides × 5 × 5 LED cube is for you.

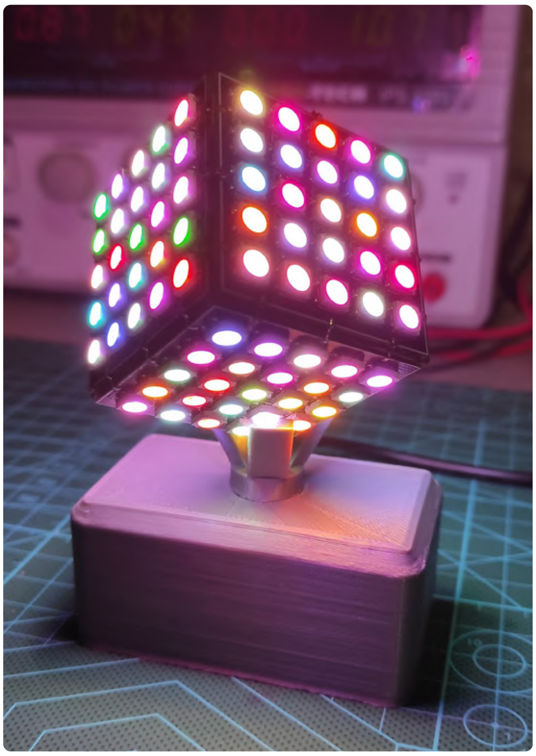


Figure 1: The RGB LED Cube.

You can create your own gadget around your own circuit board design. All it takes is to simply follow some tips to achieve a professionally made device. Let's jump straight to the main topic. I will share the story of how I made an RGB LED Cube that is powered by the Raspberry Pi RP2040 microcontroller (**Figure 1**). I will explain how to put an RP2040 into action, and I will present a basic setup for making this amazing MCU run our devices.

RP2040-Based RGB LED Cube Design

Since early 2021, the global electronics components shortage has made it difficult to get inexpensive yet efficient MCUs. But I found that the RP2040 can help engineers during these tough days due to its compact size, ease of use, high performance, and cheap price. When I started the RGB LED Cube, the MCU was fairly new and there weren't many detailed projects built around it, so the project offered me my first opportunity to deal with such an MCU [1].

I wanted to find the most appropriate power and programming setup for my RP2040, so I randomly chose to build a 5 × 5 LED matrix controlled by the MCU. But soon thereafter the single matrix idea turned into a six-piece connected matrix that formed a cube.

First things first. It is important to know how to deal with this MCU, and you must understand its requirements. For that, refer to the detailed schematic (**Figure 2**). The schematic is divided into four blocks: the MCU, power, USB-C, and memory. Let's now consider each block.

The Microcontroller's Setup

The first block labeled as *RP2040* in Figure 2 shows the basic electronics components needed to be set around the MCU, and you can see that it has many capacitors included! I recommend that you consider each used part in this section to avoid any surprises when you put your hardware design into action. These capacitors are mainly 100-nF decoupling capacitors, and it has to be set for each power line of the MCU.

These bypass capacitors will filter the power supply noise caused by the other circuit's elements since this power noise will be shunted through the capacitors. These capacitors also provide another aspect of use when acting as a local charge supply which helps the inner circuits of RP2040 during considerable voltage drops caused by a sudden increase of current demand.

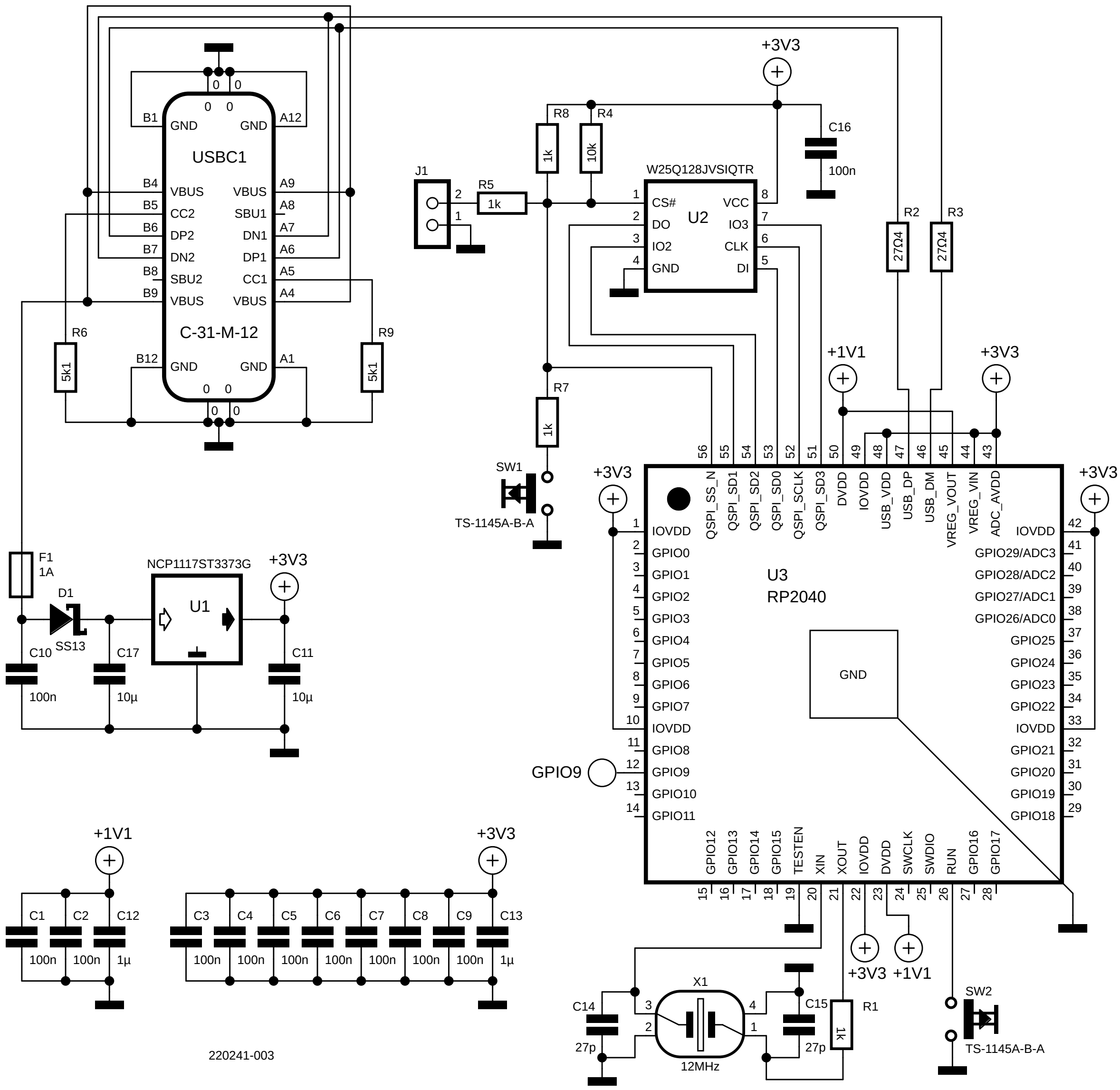


Figure 2: The RP2040 and the basic hardware setup.

Now you understand the importance of the decoupling capacitors. But this is not all of it! During the PCB design process (**Figure 3**), make sure you keep the capacitors as close as possible to your MCU's power pins. This will ensure a high filtering effectiveness.

The RP2040 has its own internal oscillator, so basically it does not strictly require an external oscillation source. Since this internal crystal is not as stable as needed, an external 12-MHz oscillator is recommended (**Figure 4**). We used the X322512MSB4SI oscillator. (Remember that we picked up the most available parts on the market, but you still can make your own selection.) The two main factors to consider about crystals are the load capacitance (LC) and the Equivalent Series Resistance (ESR). You can get the necessary parameters from the part's

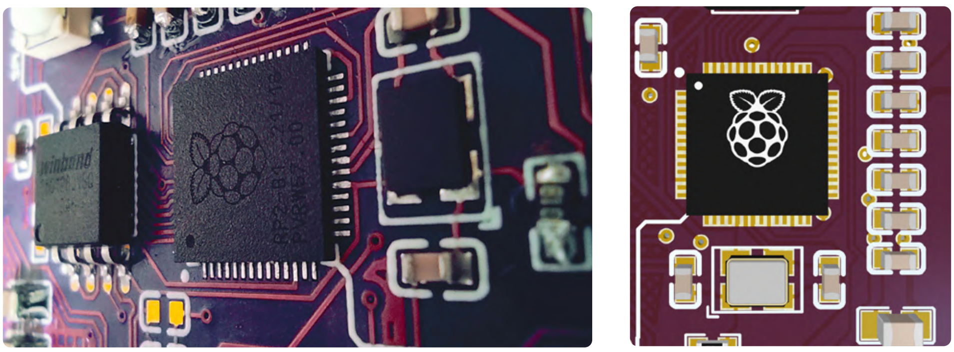


Figure 3: Decoupling capacitors and crystal positioning.

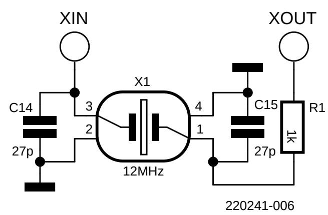


Figure 4: Crystal resonator setup.

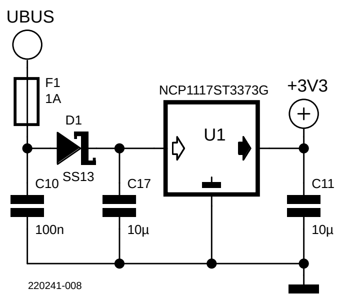


Figure 5: The 3.3 V voltage regulator setup.

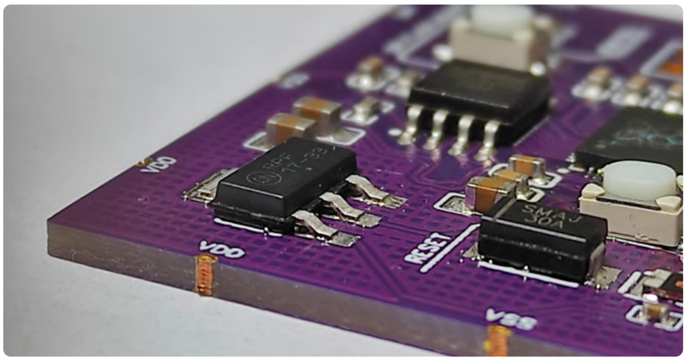


Figure 6: The voltage regulator NCP111733T3G.

datasheet. The device that we used had to have a load capacitance of 20 pF. In order to satisfy such a need, we placed two capacitors (C15 and C14) in parallel to the XIN and XOUT of the crystal. I see that you are asking a question here! How can you determine the appropriate capacitor value? The desired load capacitance (LC) value combines C15 and C14 through a simple formula:

$$LC = \frac{C14 \times C15}{C14 + C15}$$

Since C14 and C15 have the same value, then LC = C14/2, so we considered to use a 27 pF for C14 and C15. Yes, I know that 27/2 = 13.5 pF less than 20 pF, but do not forget that you will use a printed circuit board (PCB), which will add extra capacitance through its tracks and this could reach up to 7 pF. So, in total, we have 20.5 pF capacitance, which is close enough to 20 pF.

The ESR factor is also available through the manufacturer's datasheet. In our case, it is set to 80 Ω, so I have found that such value along with a 1-kΩ resistor (R1) is sufficient to prevent the crystal from being over driven.

Power Management

This was the first time I got to deal with a microcontroller that requires a 1.1 V. But I had no worries about that requirement since the MCU already has an internal Low Dropout Voltage Regulator built specifically to provide 1.1 V for the device. It had to be connected to a voltage source, and here I provided 3.3 V for the RP2040 through an external voltage regulator (**Figure 5**).

The NCP1117 is a commonly used voltage regulator, and it is available in several output voltage values, so make sure that you are using the 3V3 one defined as 33T3G. I used a resettable fuse (PTC) alongside the power line input (UBUS) to protect my circuit against overcurrent faults.

The compact size of this device (**Figure 6**) makes it suitable for our small PCB design. Keep in mind that we are designing this board to build a small Magic LED Cube, so basically, we need to distribute all our electronics components around a compact PCB layout of, at most, 45 mm × 45 mm. This is the main reason why we are picking up small component packages. But if you are willing to design hardware around RP2040 for a different purpose, then you are free to select bigger components.

USB C Connector

No need for a USB-to-TTL converter here because the RP2040 MCU provides an internal full-speed USB interface. So all it takes is just picking up a USB connector of your choice and connecting the USB_DP and USB_DM of RP2040 to the USB D+ and D- of your connector. I used a USB type C connector (**Figure 7**).

You can remove those pull-down resistors (R9 and R6) because the MCU does not need any external pull-up or pull-down resistors, but since this is the most appropriate USB C setup that we can come up with for all our projects, we will keep them as long as it has as a small size. So, I picked up a resistor package size of 0402 to avoid any clutter on the PCB layout. This certainly provided an easier track routing. Using this connector, the circuit board could act as a Host or

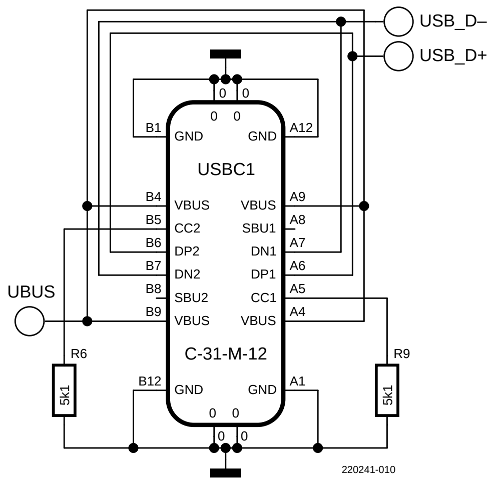


Figure 7: USB C setup for RP2040.

a Device depending on the boot mode.

The connector's placement on the board has to be slightly moved away from the board's edge because we have to assemble the cube pieces through their edges. Basically, we need a free surface of approximately 1 to 2 mm around the board edges.

Flash Memory (Program Storage)

Comparing to the classic AVR and ARM microcontrollers that we used to deal with, the RP2040 needs an external storage memory where we could place the program code and make the RP2040 Boot on that code and execute it. Talking about booting and code execution then we are talking about high frequency data transfer from the storage chip to the MCU and this is the reason behind the selection of a Quad Serial Peripheral Interface (QSPI) flash memory chip to store the program code (**Figure 9**). These flash memories could have several storage capacities. In our case, we are specifically using the W25Q128JVSIQTR device (**Figure 8**), which has a storage space of 128 Mbit (16 MB), and this is the maximum space memory that the RP2040 could handle through its data bus.

The QSPI pins should be connected directly between the flash memory chip and the RP2040, and these two parts had to be placed in closely on the PCB layout to keep the tracks between them as short as possible and to prevent any data transmission errors caused by nearby components interferences.

I remind you that when we detailed the USB C setup I mentioned that the RP2040 could be used as a host or as a device depending on the boot mode. This is specifically depending on the QSPI_SS pin connection during the memory bootup (Figure 8). And here comes the use of resistor R4 acting as a pull-up of 3.3 V and putting the chip select pin (CS#) at the same voltage as its own 3.3-V supply pin as the device is powered up. When we turn on our device, the QSPI_SS pin of RP2040 will automatically pull up, but the pin state stays unknown during the pin state switching. This resistor (R4) will ensure the pin state and set the memory to operate properly to feed the program code to the RP2040 for execution.

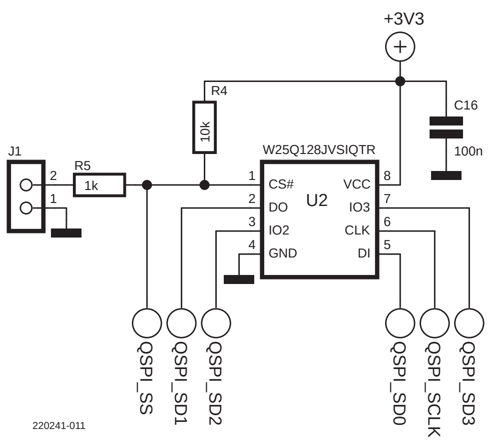


Figure 8: QSPI flash memory setup.

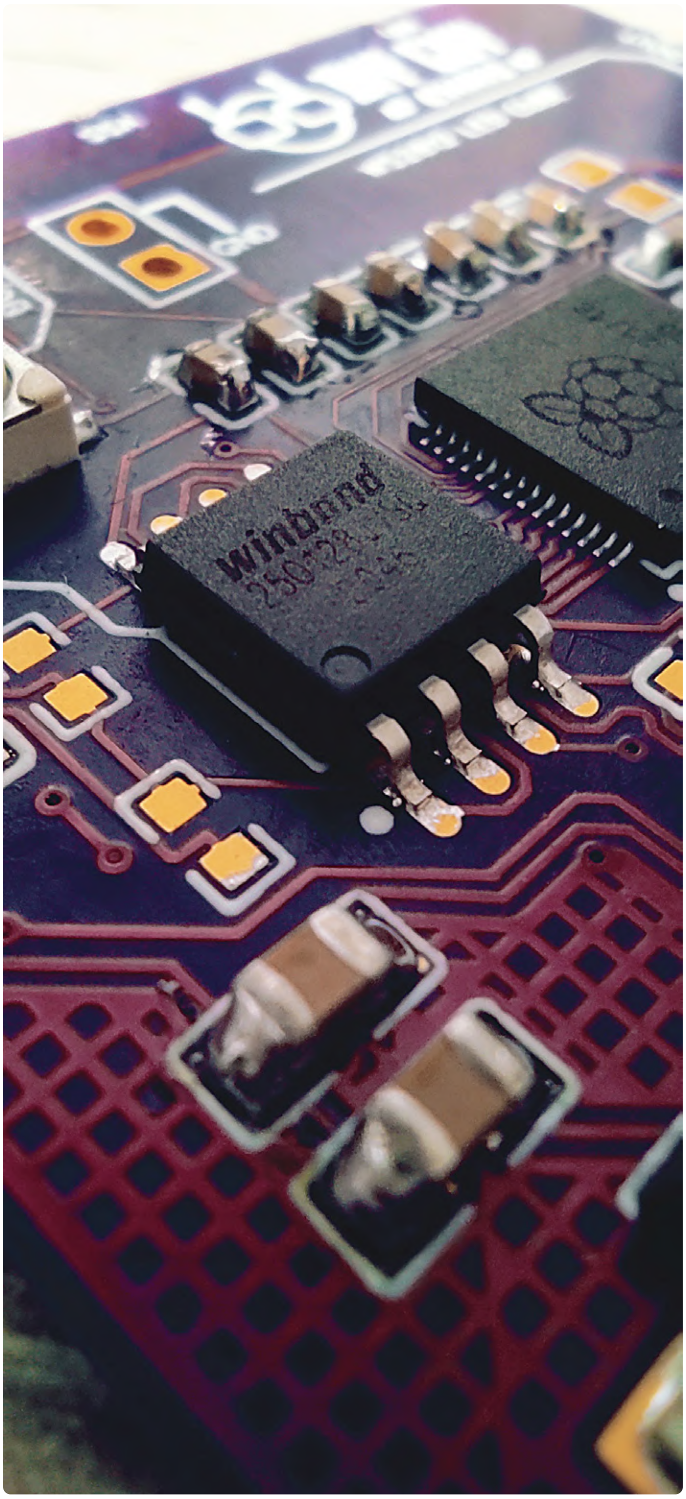
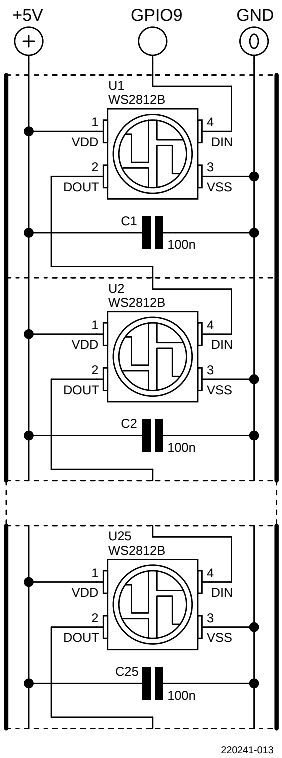


Figure 9: W25Q128 real part size.



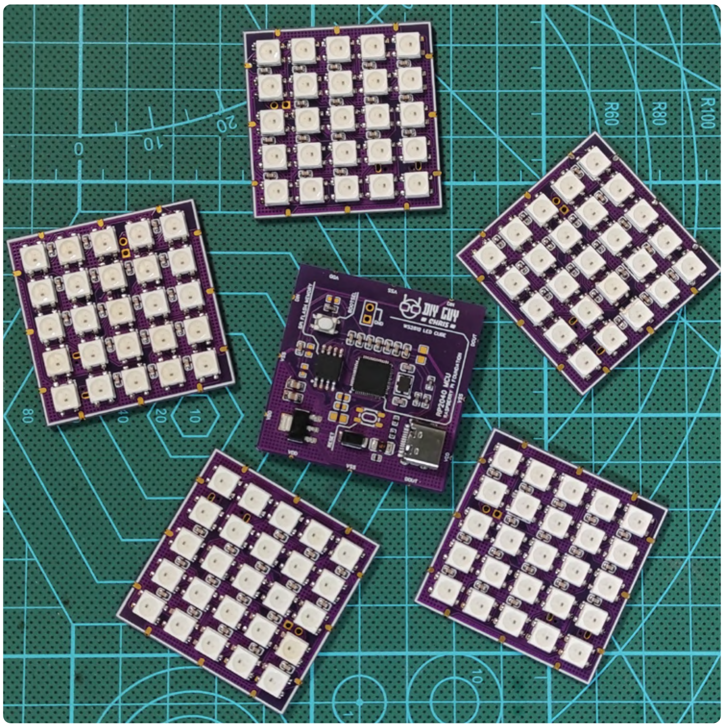


Figure 10: The LED circuitry and finished boards.

Regarding the second boot mode (storage device mode), this is where the use of the jumper (J1) comes in by pulling down the QSPI_SS pin when we power up the circuit board. The memory will be recognized as a USB storage device and then the program could be copied directly in it. A reset of the circuit power will switch the boot mode and then the RP2040 will execute the copied program.

More About the Journey!

The bottom side of this PCB contains a 5 × 5 RGB LED matrix connected in series (**Figure 10**), so if we consider connecting six pieces of this circuit board, we end up with a total of 150 LEDs. How can we control them all using this RP2040?

Actually, you will use one pin to control these addressable LEDs. Have you ever heard about addressable LEDs (in our project, we used the WS2812B)? These are LEDs with an integrated controlling device and a data signal reshaping amplification drive circuit all in one single housing package of 5 mm × 5 mm in size. Brightness data and light color will move from the RP2040 control pin all along the LEDs connection to update each LED's controller.

We built this circuit board to test our hardware design capabilities with an RP2040 MCU. Since it has worked for this Cube project, this hardware setup will certainly work for your custom projects. Just follow the recommendations and tips about the necessary components and you will get your device up and running without any big surprises.

All files associated with this project (schematics, Gerber files, RP2040 software, and STL-files for 3D printing) can be downloaded from [1].

220241-01

About the Author

Mohamed Belkhir Sayari — "DIY GUY Chris" on YouTube (www.youtube.com/MEGADAS/) — is a Tunisian electronics R&D engineer with the spirit of true innovator. He treats each project as a learning adventure and an opportunity to demonstrate his skills.

Questions or Comments?

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

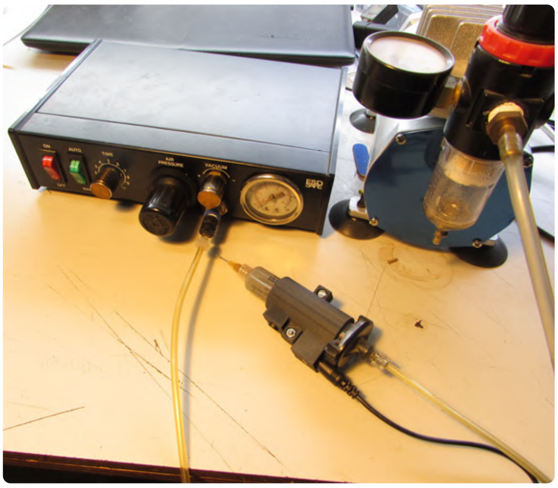
WEB LINKS

[1] DIY GUY Chris, "How to Build Magic RGB LED Cube (With Code and Files)," March 23, 2022: www.youtube.com/watch?v=A0OtHySzadk

Auto On/Off for Solder Paste Compressor

By Luc Lemmens (Elektor)

Do you often forget to switch off the air compressor of your solder paste dispenser? This simple project may be just what you need.



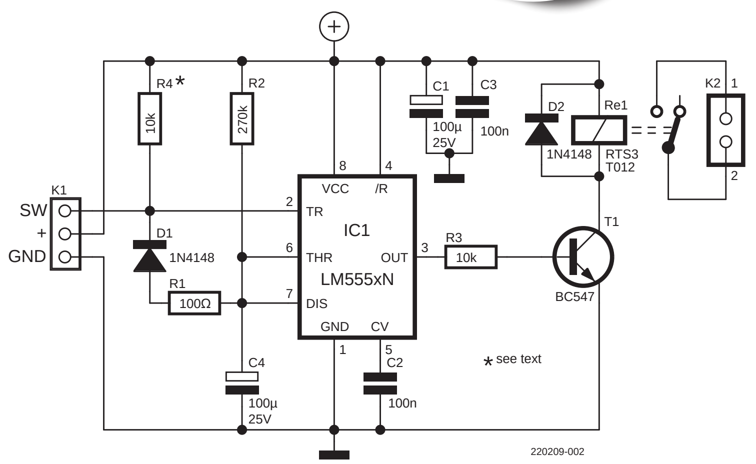


Figure 1: The circuit diagram. A 555 does the trick.

In my home lab, I have been using a dispenser for applying solder paste for several years. The header picture shows the kind of device I mean. It regulates the compressed air from a small 130 W airbrush compressor so that exactly enough paste gets on the solder islands of a circuit board to solder SMD components properly. The compressor has its own pressure switch, which automatically turns it on and off at a preset air pressure level.

Why This Project?

When you stop operating the dispenser, the compressor should stay quiet: with perfect air hoses, couplings, and valves, the system remains pressurized. But in practice there is always a leak somewhere — so every now and then, the compressor switches on again. The good thing is that this reminds you to switch the compressor off. The bad thing is that even the most silent compressor is noisy, so you'll probably have to walk up to it to turn it off. But if the dispenser can control the output of the compressor, it should also be able to control its input (i.e., the compressor's connection to the grid).

The combination of compressor and dispenser works very simply: every time the user operates the hand or foot switch of the dispenser, an electromechanical air valve is opened. This enables airflow from the compressor to a syringe that contains the solder paste and doses the amount of paste coming out of the needle. In the project presented here, the switch will also trigger a monostable multivibrator that controls a relay that powers the compressor. The monoflop is retriggerable and its on-time is longer than the "normal" interval between two key presses when you're applying solder paste on the PCB. It remains switched on for about 30 seconds after the dispenser was last operated. After that time, the compressor is powered off but is immediately powered on again as soon as the hand or foot switch is pressed again.

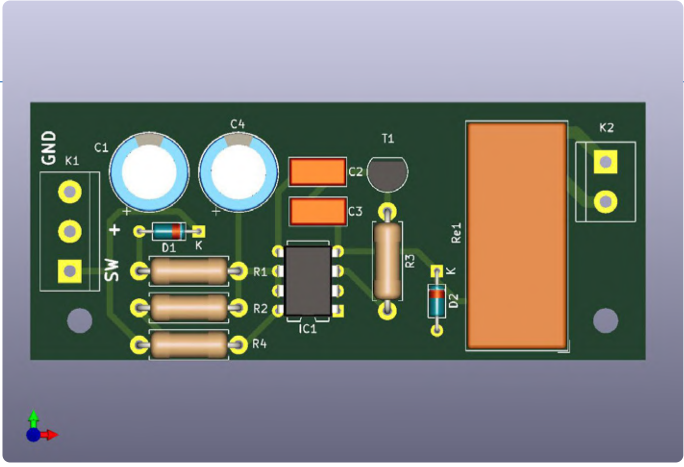


Figure 2: 3D view of the PCB.

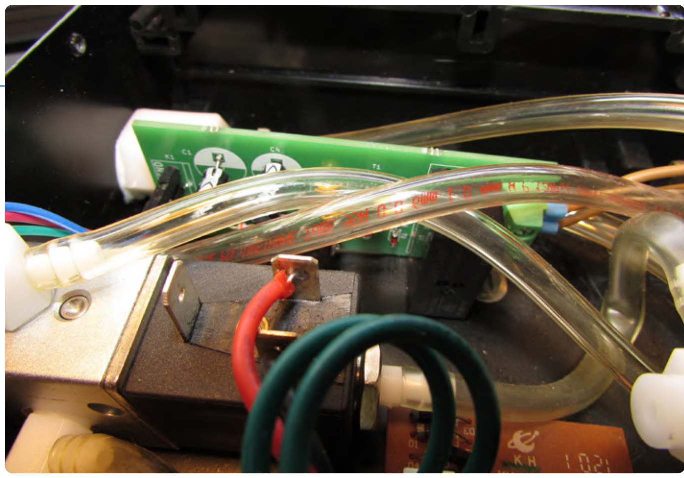


Figure 3: The board mounted between the dispenser's air hoses.

Classic 555 Circuit

My dispenser is easy and safe to modify: there is a transformer inside and all electronics are low voltage, galvanically separated from the power grid. I haven't reverse-engineered the internal electronics. For this project, the only important things were the on-board 7812 voltage regulator (that can also be used to power some additional hardware) and the switch input, which apparently has a pull-up resistor to +12 V (the trigger switch pulls it to GND). Easy enough to connect the simple circuit shown in **Figure 1**, which can be recognized as a standard application of a 555 as retriggerable monostable. When the trigger input (pin 2) is pulled low, the output (pin 3) goes high for approximately 1.1 x R2 x C4 seconds — in this project half a minute.

A second trigger (before the 555 times out) discharges capacitor C4 via R1/D1 and restarts the timing interval. Resistor R4 is an optional pull-up resistor for testing the circuit or using the circuit in other applications, it is not needed when connected to the dispenser.

Building the circuit

Unplug the power cable before opening the enclosure of the dispenser! The PCB (**Figure 2**) is a quick-and-dirty design I made with standard KiCad libraries, nothing fancy. The design and Gerber files can be downloaded from [1]. It was made to fit inside the enclosure of my dispenser; the space is a bit tight but the board is relatively small. **Figure 3** shows the board mounted and connected. The GND and SW terminals (K1 pin 3 and pin 1, respectively) are connected to the jack socket for the hand/foot switch on the back panel, the +12 V (K1 pin 2) is directly soldered to the output pin of the 7812 voltage regulator (**Figure 4**). The exact hardware inside the dispenser may be different in other types or brands and I cannot guarantee that my solution will work in all of them.

To connect the compressor, an IEC C13-style power outlet was added to the back panel of the dispenser, the wiring is shown in **Figure 5**. Note

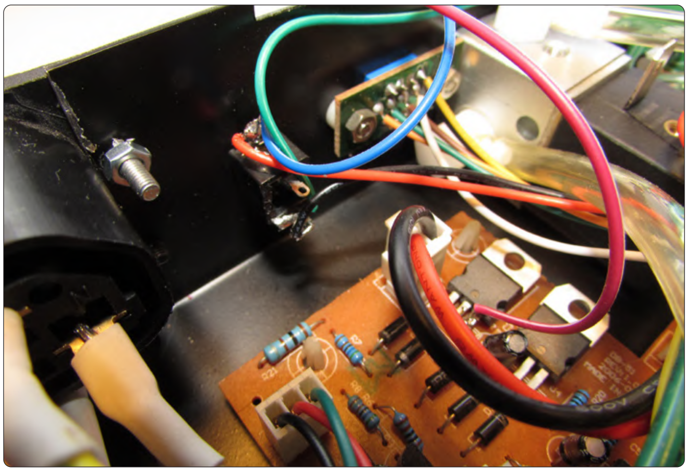


Figure 4: Power supply and switch connected.

3D Print Your Own Handpiece

When you buy a solder paste dispenser like the one I have, you'll probably get both a hand and foot switch as accessories. I prefer the hand switch, but it is part of a handpiece designed for 150 cc syringes, and too large for the 30 cc packaging in which most solder pastes are sold. On the website of Adafruit [2], I found files to 3D print your own handpiece for these smaller syringes, but, in my opinion, this design is too fragile to last long. Fortunately, the Rhino design file was also included in the download, and I changed it to a sturdier version. I didn't put too much time in it, though, it does the job but can be optimized for a better fit of the switch and connector inside the case. The original white transparent handpiece in the picture looks to be about the same size but is actually much wider.

The black 3D printed version that fits the 30 cc syringe in black.

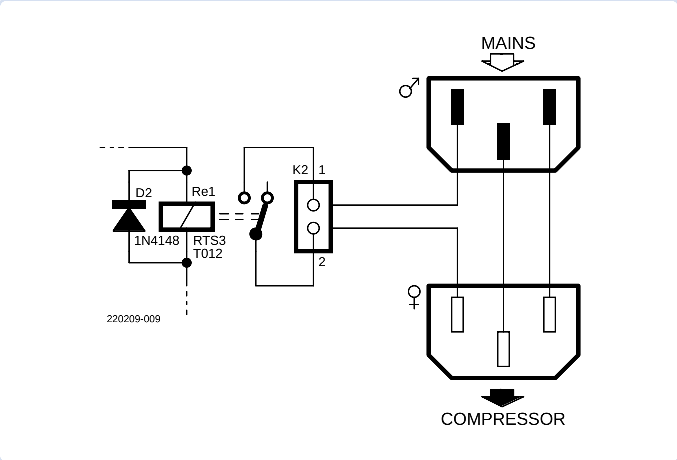


Figure 5: Wiring for the power outlet.

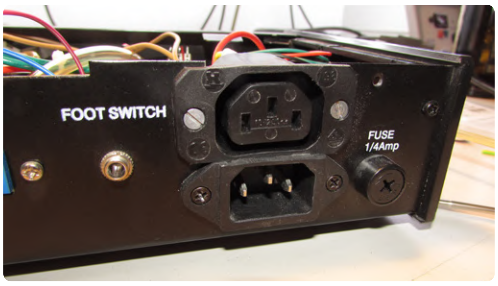


Figure 6: Power outlet for the compressor.

that the original connections on the power inlet (the male connector) remain intact, only the wires that need to be added to this connector are shown in this illustration.

Figure 6 shows the power outlet installed for the compressor, of course, its original power plug must be replaced with a matching C14-style plug.

Note that the relay used here will only survive for small airbrush compressors like the one I have. The huge in-rush current of heavier compressors when they are switched on will kill almost any standard relay contacts, or weld them together. I had the idea for this circuit for a long time and now that it is ready, I wonder why I waited so long. It is so much easier that I do not have to switch the compressor on and off separately. It saves a socket and — most important — annoyance!

220209-01

Component List

Resistors
R1 = 100 Ω
R2 = 270 k
R3,R4* = 10 k

Capacitors
C1,C4 = 100 µ/25V radial
C2,C3 = 100 n

Semiconductors
D1,D2 = 1N4148
T1 = BC547
IC1 = NE555

Miscellaneous
K1 = 3-way terminal block, pitch 200 mil
K2 = 2-way terminal block, pitch 200 mil
Re1 = relay 12V DPDT (Schrack RTS3T012)
IEC C13 power outlet (e.g. Bulgin Limited PX0675/63)
IEC C14 plug (e.g. Bulgin Limited PX0686/WH)

* = see text

About the Author

Luc Lemmens started working for Elektor in March 1990 after his studies at the Technical University Eindhoven. He has many interests, which means that he knows a little about a lot of topics in electronics. Of course, he has also written or edited software in a wide range of programming languages — and, especially in his early days at Elektor, in assembly language. Nowadays, he usually limits himself to the Arduino IDE, which is perfect for most simple projects. In his spare time, Luc likes to play with pinball machines, especially repairing and restoring both modern electronic machines and electromechanical ones (with relays and stepper units).

Questions or Comments?

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

Disclaimer

Unplug the dispenser before opening the enclosure. Elektor and the author accept no responsibility for damage arising from the use or installation of this circuit. Opening the case of the dispenser will void the warranty!

WEB LINKS

[1] This project's downloads: www.elektormagazine.com/220209-1
[2] Original 3D Handpiece Design Files: https://learn.adafruit.com/3d-printed-solder-paste-dispenser-hand-switch

Elektor **Video Content**

Livestreams, Webinars, and Courses for Engineers and Pro Makers

By The Elektor Editorial Team

Elektor community members are always looking for new projects, engineering tutorials, and technical insights, which is why we produce a wide variety of video content in addition to our regular magazine editions. Join our livestreams, webinars, and courses to level up your engineering capabilities.

Elektor **Engineering Insights**

For six decades, Elektor has been known for its printed magazine covering all kinds of electronics. To bring you even more, up-to-the-minute content, there's the **elektormagazine.com** website and the weekly E-Zine(subscribe at **www.elektormagazine.com/elektor-newsletter**). But text and pictures aren't everything: we also produce a lot of video. For example, at embedded world 2022, we had two cameramen and three editors on assignment to report on the latest products and to interview experts. Check out our Elektor YouTube channel (**youtube.com/elektorim**)!

Since spring, we've added additional regular sources of inspiration for you: editorial webinars, courses (**www.elektormagazine.com/elektor-academy**), and two monthly live shows on YouTube. All of this is streamed directly to your PC, tablet or smartphone, and you can also ask questions of the presenters and experts via a chat function.

Elektor Webinars

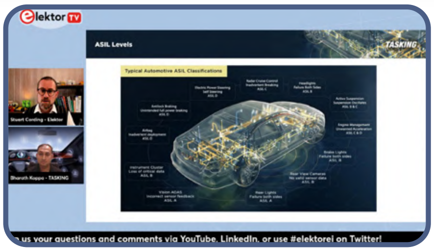
You can register for the editorial webinars at **www.elektormagazine.com/webinars**. The Elektor editorial team conducts a webinar for each of the regular bimonthly issues, focusing on the main topic of that issue. For example, the main topic for the issue in your hands is "Wireless," so be on the lookout for the webinar, which will take place on October 13 at 16:00 CET.

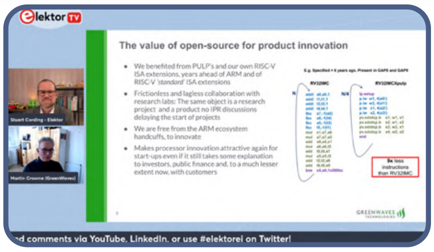
For the two YouTube live shows (see below), you don't have to register at Elektor, but the live character and chat option are the same. For the chat function, you have to be logged into YouTube.

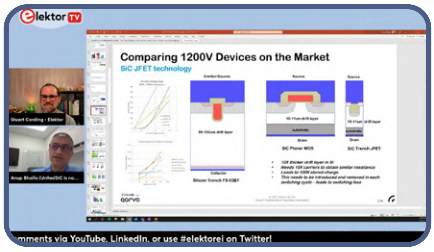
Elektor Engineering Insights

Being stuck at home during Covid for two years without access to trade shows, Elektor's Stuart Cording wanted to find a way to recreate the technical discussions he typically had at exhibitions and congresses. With a wealth of industry contacts and the support of the Elektor team, he decided to bring his show, Engineering Insights, to life. Streaming live every month on Elektor TV on YouTube, LinkedIn, and Twitter, Stuart's guests share their knowledge on every conceivable technical topic, from the simple to the cutting edge.

To date, Elektor Engineering Insights has interviewed those developing new low-power devices based upon the RISC-V processor along with a company that builds the tools required to simulate custom instruc-







tions on this core. We've also looked at wide bandgap technology, examining the advantages of gallium nitride (GaN) and silicon carbide (SiC) devices over traditional silicon MOSFETs. And, for those wondering whether their invention could be patented, Stuart chatted with the European Patent Office to better understand the application process, and spoke to a Portuguese solar-energy developer to learn from their experience of applying for a patent.

"Over the years, I've learned so much from my peers in the industry, visiting their offices, their labs, and meeting them at trade fairs," Stuart says. "The restrictions imposed by Covid were devastating, essentially cutting off the exchange of knowledge and ideas that trade fairs and congresses offer. I really hope that my show helps to complement such events and is seen by the engineering community as a valuable resource to learn about the latest technologies, software, and development approaches."

More info at:
www.elektormagazine.com/eei

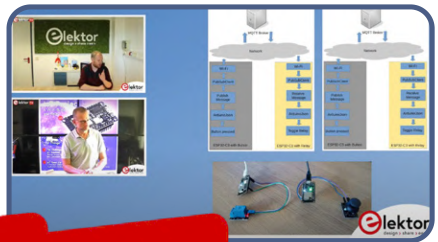
Elektor Lab Talk

Mathias Claußen and Jens Nickel are both Elektor editors. Mathias is our software expert in the Elektor Lab, and Jens is the magazine's editor-in-chief. Both are enthusiastic about electronics and programming, even in their private lives. They are always interested in the latest gadgets, new electronic techniques and highlights from the software world. Don't miss the shows where they talk about the latest Elektor projects and hot products or just talk shop about the world of electronics. From time to time, they also invite experts from different fields of electronics as guests. On every show, viewers can also win development boards, books or other products from the Elektor Store!

Elektor Lab Talk streams every month. For a list of upcoming and all past broadcasts, visit **https://youtu.be/alOJrB1mdp8**. Of course, you can watch the shows afterwards "on demand," but then you won't be able to chat with Mathias and Jens. So don't forget to watch live! If you are logged into YouTube, you can press the little bell to get a reminder. Be sure to subscribe to the channel and like our videos.

More info at:
www.elektormagazine.com/elt

220357-01



Bicycle Electrification

Hands-On with an E-Bike Retrofit Kit

By Dr Thomas Scherer (Germany)

A steep hill, my self-respect as an electronics enthusiast, and my advancing age were the reasons I decided to electrify my trusty bike. After an initial version with a front drive motor, which I built many years ago, I have now opted for the right choice: a mid-drive motor. Read on to learn about my experience with this conversion project.

Figure 1: Before: My bicycle with a front hub motor and improvised wiring.

I used to live in Frankfurt. At that time, my cycling life was still reasonable. Pretty much everything there is flat, the same as in Münster (the hotbed of cycling in Germany) or the Netherlands, where the highest mountain has elevation of 322.4 metres — dismissed as a molehill by residents of the Alps. With my bike I could get nearly everywhere in the city faster and more comfortably than with a car. Around 15 years ago, however, I relocated to Bad Vilbel, around 6 km from Frankfurt. Although Frankfurt was still within cycling distance, there was a hill between my home and the big city, and the first time I tackled the hill was an exhausting experience.

After the curses and vow of 'never again', my thoughts quickly turned to electrical assistance. One option would have been to buy a new pedelec, but I had recently purchased a very comfortable bicycle with front and rear suspension and belt drive instead of chain drive, so retrofitting was a more attractive alternative. And as a bonus it satisfied my desire for tinkering. Right after arriving home, I latched on to eBay and ordered the necessary parts for bike electrification. At that time there were only two choices: front drive and rear drive. I didn't want to sacrifice the belt drive, so it had to be a front wheel motor. **Figure 1** shows my bike after the initial conversion, consisting of a thick, heavy 36 V hub motor from China that I ordered together with a controller, brake levers and pannier rack battery.

Reasons for a Mid-Drive Motor

With the hub motor already installed in the front wheel, converting my bike didn't take very long. From the first trial ride I learned two things: first, that the Bad Vilbel hill was no longer a fearsome challenge, and second, that the controller manufacturer hadn't given much thought to German road traffic regulations. The motor had more than enough power, but it quickly drained the battery, and the entire setup was basically not street legal in Germany (see the **box 250 W and 25 km/h**). Fortunately, I was able to identify the controller and track down a tool online that allowed me to limit the motor power and make everything legally compliant. I used a Hall sensor triggered by magnets in the front brake disc to comply with the prescribed speed limit of 25 km/h. A bit of circuitry ensured that the motor was only powered when the bike was travelling at less than 25 km/h. I even managed to gently reduce the power between 20 and 25 km/h.

Actually, I could have been satisfied, but as the saying goes, better beats good. In particular, I was worried that after a while the front fork might break due to material fatigue caused by the stress of the motor torque. The real turning point came from a cycling outing together with my lady friend. The bike she was riding — an older model, ready-made pedelec with a mid drive motor — had distinctly better battery life despite already having 48,500 km on the odometer. This meant that my motor was not especially efficient. What's more, a mid drive motor runs through the gearshift, so it provides a lot more force in the lower gears. The fact that braking energy can be recovered with a hub motor doesn't outweigh the advantages of a mid drive motor, since energy recovery is a relatively minor consideration with a bicycle.

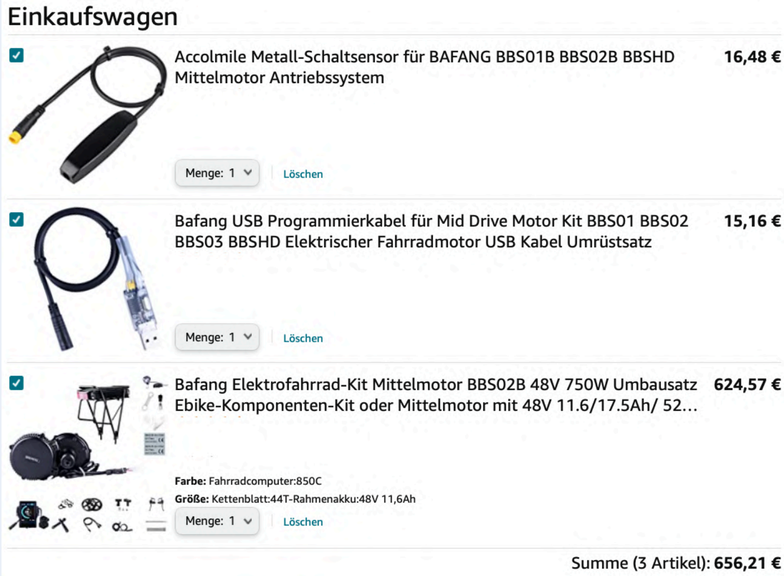


Figure 2: Screenshot of my order for the main components.

Ordered and Delivered

An online search revealed that a pedelec meeting my requirements would easily cost several thousand euros – and rob me of a lot of tinkering pleasure. Once again, the alternative was retrofitting. Aside from a few small manufacturers of retrofit kits, there is only one with a wide range of products, a large user base and reasonable prices: Bafang. Motors from this Asian manufacturer are available with various power ratings and operating voltages. The type BBS01 motor has a rated power of 250 W and an operating voltage of 36 V, making it the ideal motor for the German situation. However, ordering online is a bit tricky: this type is also available for 48 V and with a rated power of 250 W, 350 W or even 500 W. A kit containing nearly all the parts (motor, display, crank arms, chainwheel, brake levers, wiring, miscellaneous small parts and white gloves ;-)), but without a battery, can be obtained from very many sources for less than €500. You also need a suitable battery (with prices starting at €150) and some tools, and then you're ready to go.

Figure 3: The scope of delivery excluding the battery: (a) display with mount, (b) chainring, (c) crank arm puller, (d) motor, (e) thumb throttle, (f) left spacer ring, (g) USB/TTL adapter, (h) gear shift sensor, (i) gloves, (j) cable harness, (k) LED headlamp, (l) battery connector, (m) brake levers, (n) chainring guard, (o) crankset bearing wrench, (p) left crank arm, (r) right crank arm, and (s) motor mount.

Other motor types are also available, such as the BBS02 with a rated power of 750 W at 48 V (sometimes labelled as 500 W) and the BBS03 (also known as BBSHD) with a colossal rated power of 1 kW. Is this actually allowed in Europe? Amazingly, the answer is yes, because the laws are concerned with actual power rather than potential power — and the actual power is determined by the motor controller and can be set using a tool. The main advantage of the larger motors is that they have a higher-performance controller, which can squeeze more (peak) juice out of the motor. I therefore opted for the BBS02 motor. Its peak power should be more than enough to hold its own against ready-made pedelecs.

In order to configure the parameter settings for the motor controller, you need a USB-to-TTL serial adapter. Virtually all types with a CH340 IC, and some with an FTDI IC, are suitable. The CH340 types with a matching mini connector are inexpensive, so I ordered one at the same time.

Another required component is a gear shift sensor, since changing gears under high torque is hard on a gearshift. This sensor allows the motor to be briefly switched off during a gear change.

The next question was the battery capacity. I thought 550 Wh should be enough, since my previous 36-V battery had only 400 Wh. The final question was the choice of display. The various suppliers offer kits where you can choose from different displays with different features and different prices. I decided on the

Figure 4: The 48-V frame battery: (a) charger, user guide, keys and battery, (b) left side view of the battery, and (c) bottom view of the battery and frame mount.

Figure 5: The crank arm puller at the top and the wrench at the bottom were included in the scope of delivery. I had to order the lockring wrench in the middle separately.

Figure 6: The inner mechanism of a seven-speed gear hub, regreased after removing the back-pedal brake.

commonly used 850C type because it has a colour LCD and five support levels.

Figure 2 shows a screenshot of my order from a fairly large dealer. Thanks to an opportunely available discount, the total amount including shipping was just under €630. I was satisfied. **Figure 3** shows what was delivered. In **Figure 4** you can see the frame battery with the charger and mounting bracket for attaching the battery to the frame in place of a water bottle holder.

Does It All Fit?

If you are contemplating this sort of retrofit, you should check a few things before pressing the Order button. A Bafang mid drive motor is inserted into the bottom bracket from the right and secured by a threaded ring on the left. For this the width (or length) of the bottom bracket on your bike should be between 68 and 77 mm. If it is a bit narrower than 68 mm, you can compensate with spacer rings. If it is a bit wider than 68 mm, you will also need 6 mm washers for the motor mount on the left side. In addition, the inside diameter must be at least 33.5 mm and should not be much larger. This means you need to measure everything first. In my case, it fitted nearly perfectly.

It's also important to order the right brake levers at the same time, because they switch off the motor when you brake. In many cases you can choose between levers for rim brakes and levers for hydraulic disc brakes, with no difference in price.

When browsing through the immense amount of online content on Bafang motors, I encountered several reports on the risks of operating 48 V motors at a slightly higher voltage. There are batteries with an additional cell (14 cells instead of 13), which means they have a nominal voltage of 52 V. A fully charged 48 V battery can have an output voltage as high as 54.5 V under no-load conditions. A 52 V battery can therefore apply nearly 59 V to the motor controller terminals, and apparently the motors don't like this in the long run. So remember: 48 V is enough.

I actually have a good collection of tools, but I'm not a passionate bicycle mechanic, so I didn't have a suitable lockring wrench (shown in the middle of **Figure 5**) for removing the retaining rings of my crankset bearings. So I bought a low-cost wrench. The required crank arm puller was already included in the delivered items.

A final aspect is that with Bafang mid drive motors you can choose from chainrings with 42 to 46 teeth, which affects the gear ratio. I opted for the standard chainwheel with 44 teeth. Offset versions with fewer teeth are also available from accessory shops, if you want to put together a mountain bike with especially high drive force.

Installation

As a preliminary remark, a detailed description of the installation process would require a much larger article and is in any case certainly not necessary. There are an enormous number of reports on this topic available online, as well as many YouTube videos dealing with all the mechanical aspects. The kit also comes with instructions (in English), which I found helpful. I can't even recommend any particular online sources, since it's really worthwhile to read up on this before you start. That's because the various reports and instructions have their individual strengths and weaknesses, and in some cases rather unique interpretations.

The basic procedure: Remove the left and right crank arms and the crank axle, bearings and chainwheel. Insert the motor from the right and secure it on the left with the threaded ring, and attach the chainwheel and chainwheel guard on the right. Then fit the new crank arms and the old pedals. Next, exchange the brake levers and mount the display and thumb throttle on the handlebar. Finally, fit the battery and connect the wiring harness. Apply a bit of Loctite to threaded joints where necessary to ensure they stay tight. If you are used to working on bikes, you can get this done in an afternoon if everything fits.

My bicycle, however, presented a number of challenges, so I spent several days on the

250 W and 25 km/h

The legal situation of a DIY pedelec (also called e-bike25) varies considerably from one place to the next. In the USA you can more or less do what you want, but European residents are bound by regulations to a certain extent. As might be expected, the regulations in Germany are especially strict [1]. In Germany there are four criteria that must be met if you want to avoid violating the law:

1. Power limited to 250 W
2. Speed limited to 25 km/h
3. Ride assistance only when pedalling ('moped mode' is not allowed)
4. Start assistance only up to 6 km/h without pedalling

These criteria are far from precise. The law refers to continuous power and does not say anything about peak power. As a consequence, in some situations (e.g. riding uphill at low speed) even e-bikes from renowned manufacturers deliver hundreds of watts more than the allowed average value. The situation with regard to speed-dependent assistance is also unclear. Is a gradual decrease from 250 W to 0 W between 20 and 25 km/h necessary, as is sometimes assumed, or is it sufficient to abruptly reduce the motor power just before 25 km/h? And how large is the tolerance on the 25 km/h speed limit? Can you reasonably expect a tolerance of 3 km/h above the prescribed limit, as allowed for motor vehicles captured by speed cameras?

Only points 3 and 4 are more or less clear. Any thumb throttle mounted on a bike can definitely only accelerate up to 6 km/h without active rider assistance, so it is permissible.

And any electric vehicle marketed in the EU requires a EU declaration of conformity and a CE mark, but you can issue these yourself.

conversion, working carefully. At first I thought I could somehow make the front belt sprocket fit on the new drive motor. Unfortunately, that simply wasn't possible, so I switched from belt drive to chain drive. That's easier said than done. My existing eight-speed Shimano gear hub was not really suitable for conversion to chain drive, so I needed a new gear hub. First I tried a low-cost seven-speed gear hub from a discarded bike. It had a back-pedal brake, which is a safety risk and less than convenient. This led to my decision to dismantle the gear hub and remove the back-pedal brake (**Figure 6**). This turned out to be possible and I managed to do it with a bit of online help. However, it was hard to take for my hands, which are used to working with clean electronic components instead of greasy parts.

The front wheel with the hub motor had to be removed and replaced by the old front wheel, but that wheel was equipped with a modern hub generator that was no longer necessary. Powering bicycle lighting directly from the battery is now allowed, and the wiring harness even has a separate connector for this that provides 6 V at 3 W. So I put the front wheel with the hub generator on eBay and bought a suitable front wheel without a generator from eBay.

A good friend — the son of a master bike mechanic and brother of a bicycle shop owner — warned me that a normal gear hub would eventually wear out from the stress of the motor torque. Accordingly, I had a look at what is available for this purpose. The absolute top is undoubtedly the Rohloff Speedhub [2]: 14 speeds, high efficiency, very sturdy thanks to good German engineering – all doubtless very nice, but more expensive than my bike including the conversion kit. And I didn't actually need 14 gears; 'less is more' seemed to be appropriate here. Another option was the distinctly lower-cost but sturdy Nexus 5E five-speed gear hub from Shimano [3], which is designed specifically for pedelecs and is sturdier than most. This was good enough for me, so I ordered one. The next challenge came quickly: the gear hub had to be spoked into a rim. There again the right approach was to first read up, and I discovered a world that was totally new to me. The previously mentioned friend remarked that spoking a rim is a demanding task even for a trained bike mechanic, but that only spurred my ambition. **Figure 7** shows the new gear hub in a new rim with new spokes, all in my alignment jig made from a one-hand clamp, some cable ties and a vintage vice. With this I manged to achieve a runout of about ±0.05 mm sideways and ±0.15 mm radial. If you have a derailleur setup, you don't need to bother with this sort of arrangement, and you're welcome to laugh at my efforts.

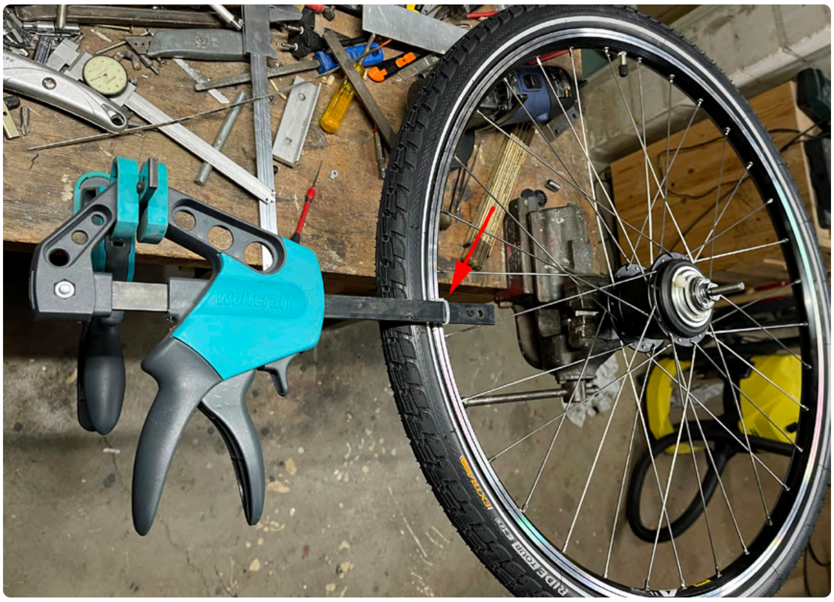


Figure 7: Improvised spoke fitting jig. The vice came from my grandfather.

The battery location also needed further consideration. Previously I had a battery screwed onto the pannier rack, but I no longer liked that approach. A frame battery can usually be mounted on the two threaded

Figure 8: Motor and left mount. My bottom bracket was 70 mm wide, so the mount needed stainless steel washers (arrow) with 6 mm holes.

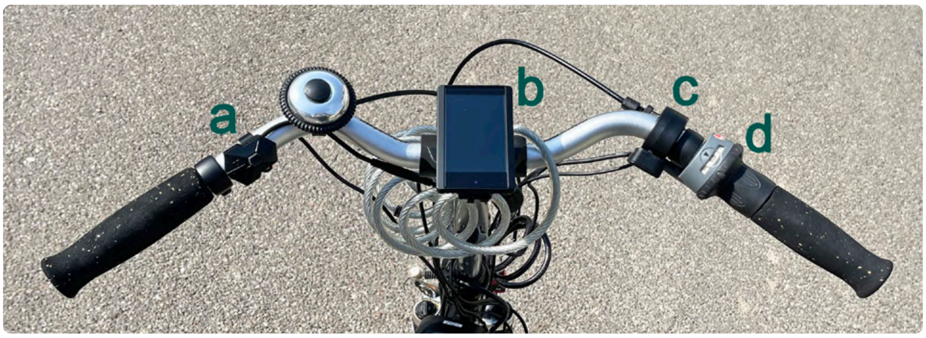


Figure 9: New handlebar fittings: (a) control buttons, (b) display, (c) thumb throttle (start assistance), and (d) shift grip before conversion to the Inter-5E.

Figure 10: After: The fully equipped e-bike, still with the old seven-speed gear hub. But it rides!

bushes in the frame intended for attaching a water bottle holder. The position of these threaded bushes on my bike was less than ideal, so I made a mounting adapter from a stainless steel sheet measuring 400 × 35 × 3 mm and screwed it to the frame with a layer of rubber underneath.

That sums up the special features in my case; I hope your situation is easier. **Figure 8** shows the left side of the installed motor. **Figure 9** shows the new fittings on my handlebar. The finished overall result can be admired in **Figure 10**.

Settings

As my teenage years are a ways in the past, I'm not especially excited by driving with illegal vehicles. In addition, the allowed maximum speed of 25 km/h is enough for me, especially in city traffic. My experience shows that my average speed now is distinctly higher than before without electrical assistance, presumably thanks to the assistance. Accordingly, the next step was to set suitable parameter values so that my DIY e-bike complied with European and German regulations.

This requires a suitable software tool in addition to the previously mentioned USB/TTL adapter. The most advanced and least error-prone option is the Bafang Configuration Tool V2.0 from Stefan Penov, which can be downloaded from his website [4] complete with source code, instructions and example setups. Unfortunately, a lot of contradictory information about programming the Bafang controller can be found online, and the instructions for the tool are not entirely adequate. If you follow my procedure, however, it should be easy for you to not only make your motor legal but also configure it to meet your needs.

The first step is to plug the USB/TTL adapter into a USB port on your computer. If you are running Windows 11 (as I am), the matching driver for the CH340 IC will be installed automatically. If not, you can download a current version of the driver from [5]. For FTDI you can find a suitable driver on the manufacturer's website. As you can see from the Device Manager screenshot in **Figure 11**, on my computer a virtual serial port was set up on COM3.

When you launch the Bafang Configuration Tool, the window shown in **Figure 12** is displayed. If you now disconnect the display from the motor, you can insert the green plug of the USB/TTL adapter into the green socket (leading to the motor) and supply power to the motor via the switch on the battery. The LED on the adapter should light up. After selecting the right virtual port on the right under *Communication Interface*, you can click *Connect* and then click *Read Flash* as the first step. This reads the default configuration of the motor controller. It is advisable to use the File menu to save this with a meaningful name, so you always have a backup if you somehow mess up the parameter settings.

Basic

The basic settings are shown on the *Basic* tab in the window. A battery with BMS normally switches off the battery if the voltage is too low. Under *Low Battery Protection* you can additionally enter a value. With a 48 V battery and a minimum voltage of 3 V per cell, the switch-off value is 39 V; with a 36-V battery, the appropriate value is 30 V.

The maximum current under *Current Limit* determines the peak power drawn from the battery, and in combination with other parameters also indirectly determines the continuous power. You should never enter a higher current here than what the controller can handle. Its maximum value is displayed at the bottom right. For a 36-V BBS01 with a rated power of 250 W, the limit is 15 A. This corresponds to 540-W power consumption. After subtracting various losses, this leaves at most 70% at the rear wheel. The peak output power is therefore around 380 W. In combination with other parameter settings, this puts you on the safe side. With the 48-V BBS02, 25 A is possible. According to Adam Ries, this results in a gross peak power of 1.2 kW, with 840 W actually available assuming an overall efficiency of 70%. That's much too much for my taste. I therefore reduced the peak current to 15 A for an acceptable real peak power of around 500 W at the rear wheel — and my controller will be grateful for

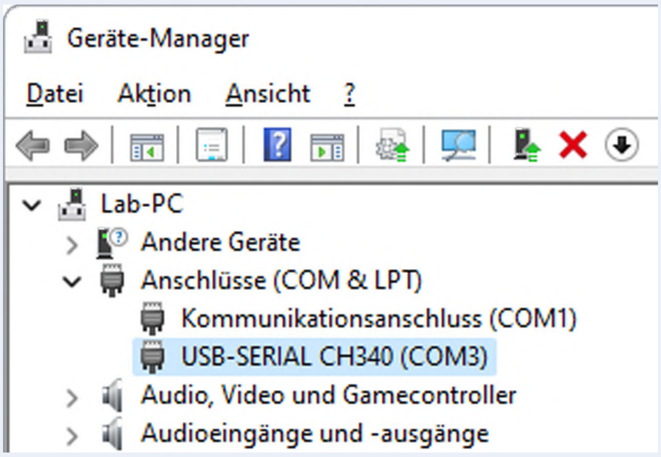


Figure 11: In the Windows Device Manager window the USB/TTL adapter shows up with virtual port COM3.

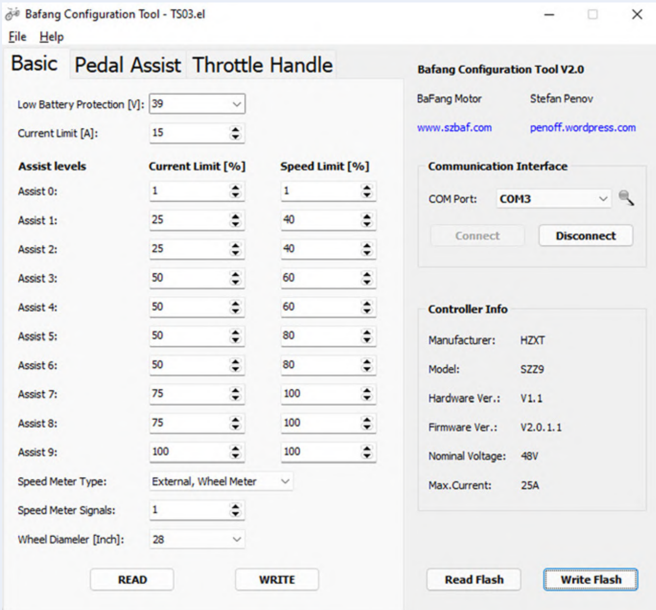


Figure 12: The **lower size** *window with the* **Basic** *tab selected, showing my parameter settings.*

this. Some ready-made pedelecs with Bosch motors have significantly higher settings. If you need to set the current limit for a BBSHD motor, you can figure it out for yourself. Its controller can handle 30 A, but the battery also has to be able to supply this if you want to fully utilise the possible 1.5-kW peak power consumption. Of course, in that case it won't be legal.

Next come the *Assist Levels*. Along with level 0, the controller supports nine additional levels. Most displays are actually small bicycle computers and only allow five additional levels. However, some do support nine levels or have a configurable number of levels. For all displays, the rule is that for *Assist 0* you should enter '0' for *Current Limit* and *Speed Limit* unless you want start assistance via a thumb throttle, in which case you should enter '1' as shown here. My display supports five levels. Unfortunately the instructions available online do not always agree on which levels should be configured for displays with fewer than nine levels. Levels 1, 3, 5, 7 and 9 are often recommended for a five-level display, but I have also seen 1, 2, 4, 6, 9 and other variants. In my case none of these worked properly, but the values in Figure 12, where each level except the last is duplicated, did work. The current limit percentage settings relate to the *Current Limit* setting, and similarly the speed percentages relate to the maximum speed set on the next tab. With this configuration, at level 3 (corresponding to level 5/6) I ride at up to

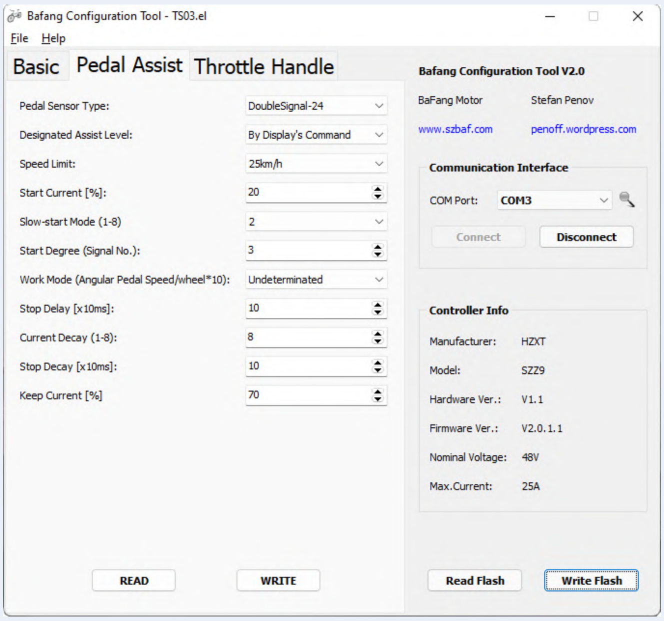


Figure 13: The tool window with the Pedal Assist tab selected. Make sure you set the speed limit here!

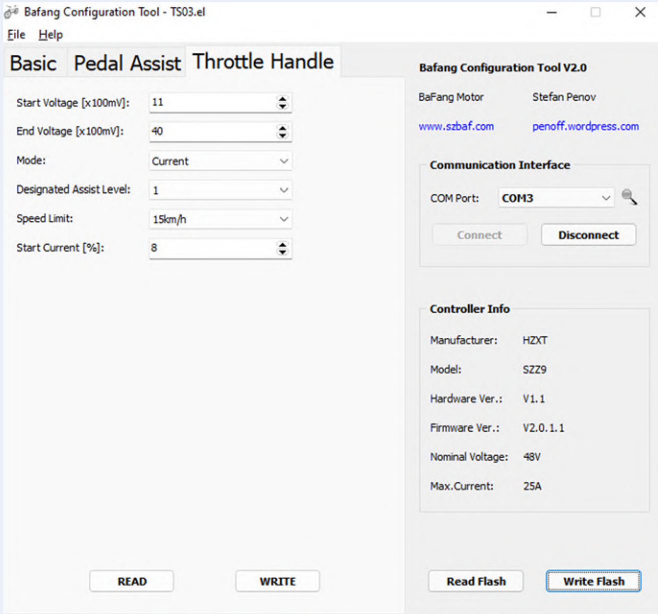


Figure 14: The start assistance parameters are set on the Throttle Handle tab.

half power for speeds up to 20 km/h if the maximum speed is set to 25 km/h. Of course, these values are not cast in stone and can be adjusted if desired. At level 1 (corresponding to level 1/2) I usually use start assistance, so there the setting is only 40%.

For *Speed Meter Type*, you should always select *External, Wheel Meter*. Enter '1' for *Speed Meter Signals* if you have fitted only one magnet on the spokes to trigger the rpm sensor. Enter the wheel diameter in inches for the *Wheel Diameter* parameter.

Pedal Assist

This tab (**Figure 13**) contains other important parameters. *Pedal Sensor Type* should be set to exactly what is shown in the figure. The *Designated Assist Level* parameter is set for control by the display, which is the only way to switch levels using the handlebar buttons. Now things start to get interesting: For *Speed Limit* you should always enter a maximum speed of 25 km/h. It is also possible to fetch this value from the display, but that allows user manipulation while riding. You can take it from me: the police do not look kindly on tricks like this — and nowadays police officers certainly know how to navigate e-bike displays. Accordingly, you should set the speed limit as a fixed value in the controller.

Start Current sets the current level for starting. Values greater than 30% result in very sudden acceleration when starting. At higher gear settings, this can overload the controller. A value of 20% (corresponding to 3 A or 150 W) gives a smooth and gentle start. *Slow-start Mode* defines the start behaviour; higher values are more abrupt. *Start Degree* sets the number of internal pedal signal pulses required to start the motor. Lower values result in faster starting, but a value of '1' should be avoided as otherwise the bike may start off backwards. Apparently nobody really knows what the *Work Mode* parameter does, so you should leave it as is.

Stop Delay is the time delay in 10 ms increments after which the motor switches off after you stop pedalling. The *Current Decay* parameter determines how much the current is reduced when a specific high cadence (pedalling rate) is reached. This ensures that even at 25 km/h you cannot exceed 250 W continuous power by pedalling very fast. I selected a high value for this. *Current Decay* sets the point where power reduction starts, and *Keep Current* defines the percentage value of the resulting reduced current. It's still not entirely clear to me which value is right for me.

Throttle Handle

This tab is only relevant if you have fitted a throttle grip or a thumb throttle. In Germany, here you must ensure that the maximum possible speed does not exceed 6 km/h when the throttle is actuated without pedalling. First you have to set the minimum and maximum voltage values under *Start Voltage* and *End Voltage*. A narrow voltage range leads to a more abrupt response to the throttle. The next parameter is *Mode*, with the options *Current* and *Speed*. I find *Current* more natural and gentler.

Next you have to set the start assistance level under *Designated Assist Level*. In my case, this is set to level 1. I specified a new *Speed Limit* of 15 km/h. Maybe you remember that a maximum speed of 40% was set for level 1. Referenced to 15 km/h, this gives exactly the allowed 6 km/h. In countries where more is allowed, you can also enter other levels and speeds. The last parameter here is *Start Current*. This parameter also only relates to start assistance and should be fairly small. With 10% of the maximum current set under *Basic*, the bike starts off very nicely.

Saving

To save the values entered on each tab to the controller, click the *Write* button. Clicking *Write Flash* writes all the values to the controller at the same time. Then you can immediately make a test ride to see the effect of the altered settings.

Time for a Test Ride

If you are inspired by my experience and want to personally electrify a bicycle in this way, you should always wear a helmet on the first few rides and wear a helmet after changing any parameter setting later on. In my case nothing unusual has ever happened, but you never know. A mid drive motor like this can develop a lot of force in the lower gears. Unlike mid drive motors from other manufacturers (e.g., Bosch), the Bafang motors do not have a torque sensor in the pedal crank. This means that the assistance is not proportional to the applied force, but instead depends on the assist level and the pedal cadence. I got used to this fairly quickly. In terms of performance and torque, my bicycle can hold its own against commercial products.

If you have gotten as far with your upgrade as described here, the next step is a test ride. I hope you find it very enjoyable!

210726-01

RELATED PRODUCTS

- **Joy-IT VAX-1030 Wireless Multifunction Meter (SKU 19199)**
  www.elektor.com/19199
- **PeakTech 6227 DC Switching Power Supply (0-60 V, 0-6 A) with colour LCD & 2x USB (SKU 19323)**
  www.elektor.com/19323
- **Velleman VTSS220 Temperature-controlled Soldering Station (SKU 19865)**
  www.elektor.com/19865

Questions or Comments?

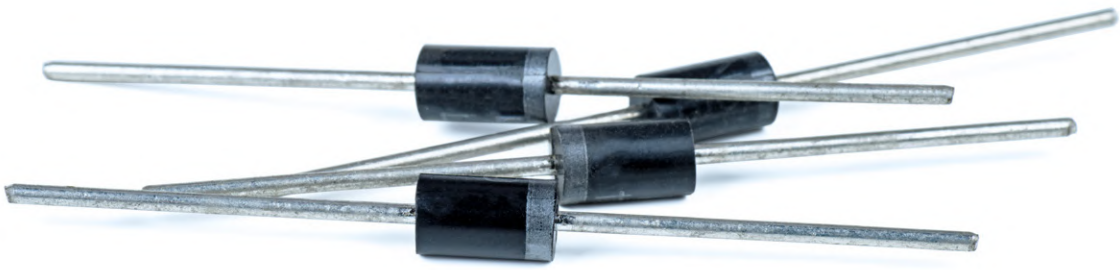
Do you have any questions or comments about this article? Contact Elektor at editor@elektor.com.

WEB LINKS

[1] German Pedelec Act: www.gesetze-im-internet.de/stvg/__1.html
[2] Rohloff Speedhub: www.rohloff.de/en/products/speedhub
[3] Shimano Nexus Inter-5E: https://tinyurl.com/bdzkh48k
[4] Bafang Configuration Tool V2.0: https://penoff.me/2016/01/13/e-bike-conversion-software
[5] Driver for CH340 (USB/TTL): www.wch.cn/downloads/CH341SER_ZIP.html

Starting Out in **Electronics**

Multiplying Voltages



By Eric Bogers (Elektor)

In the previous installment, we finished with a simple rectifier circuit that we used to generate a symmetrical power supply voltage (that is, the same magnitude of positive and negative voltage). A nice property of this circuit is that we can also use it as a voltage doubler.

In **Figure 1** we recall the schematic for the rectifier that generates a symmetrical power supply voltage. Across the top resistor, a positive voltage will appear, and across the bottom one a negative voltage of the same magnitude – measured with respect to the common node of these two resistors. This is, of course, extremely useful if we (for example) need to power a couple of opamps.

But wait a moment! We are not at all obliged to use this node of these two resistors as a common connection! If we do not do that, then we have across the two resistors (roughly) double the transformer voltage. This circuit is, therefore, a voltage doubler.

Let's rearrange the components in the circuit of Figure 1 a little bit – see **Figure 2**. This is also a voltage doubler. During the negative half periods of the input AC voltage, capacitor C1 is charged to the (singular) peak value of the AC voltage. Then during the positive half cycle, the transformer output voltage is added to this voltage across C1, so that capacitor C2 is charged via diode D2 to double the peak value of the AC voltage.

Of course, the voltage across C2 will never be equal to twice the peak value of the AC voltage. Firstly we have to subtract the forward voltage drops of the two diodes, and secondly, C2 will be discharged by the resistor (that is, the load).

This kind of circuit can be used, among other applications, to charge a capacitor to a high voltage and use it (as is done in stagecraft) to ignite a pyrotechnic display.

Figure 3 shows a circuit that triples the voltage. During the positive half period capacitor C1 is charged to the singular peak voltage. This gives the opportunity to charge C2 during the negative half period to twice the peak value. During the positive half periods, the (singular) peak value is also added to the double peak voltage of C2. This is then finally used to charge capacitor C3 via D3.

Finally, in **Figure 4** is drawn the schematic of a voltage quadrupler. During the negative half period, C1 is charged via D1 to the singular peak value of the AC voltage. During the positive half cycle, the peak value of the transformer AC voltage is added to the voltage across C1; this is used to charge capacitor C3 via D2 to twice the peak value. During the negative half period, capacitor C2 is then charged via D3 to three times the peak value, and then finally (again during the positive half period) capacitor C4 is charged via D4 to four times the peak value.

We could continue this little game — in theory, a voltage multiplier can be designed that multiplies the peak of the input AC voltage by a very substantial amount. But this not only requires a considerable number of components, but the efficiency of the circuit quickly diminishes also: diode multipliers are cursed with an appreciable internal resistance. Under load, the voltage collapses very quickly. If we require a somewhat load-capable high voltage, then it is better to use a transformer. Use that to transform the voltage up and then subsequently rectify it.

Other Diode Circuits

Before we plunge into Zener diodes and other interesting diode variants, we will first take a quick look at a few other frequently occurring diode circuits.

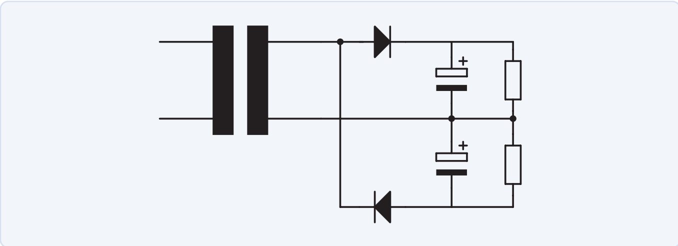


Figure 1: This is how we generate a symmetrical power supply voltage.

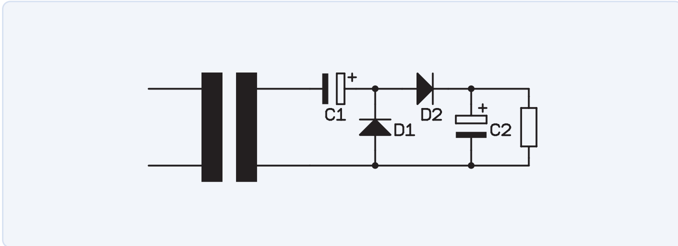


Figure 2: The circuit is also a voltage doubler.

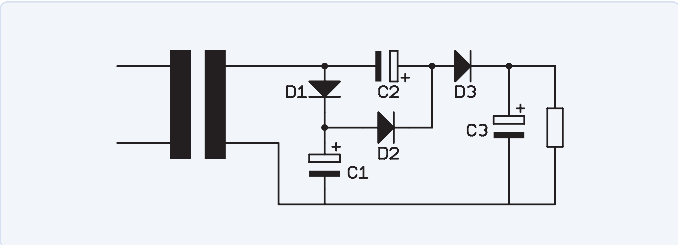


Figure 3: Voltage tripler.

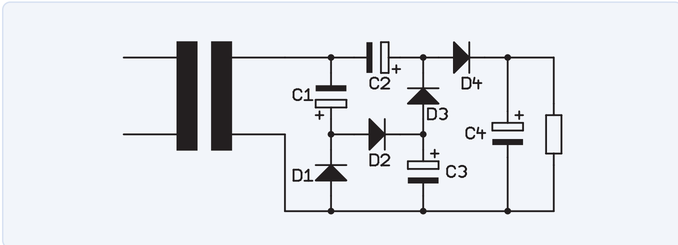


Figure 4: Voltage quadrupler.

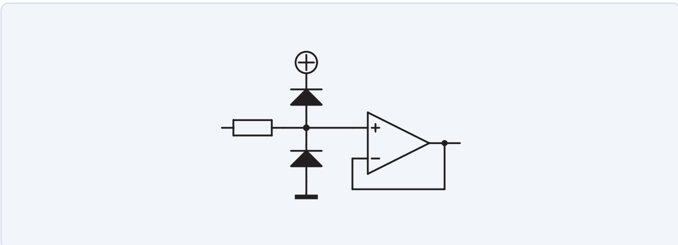


Figure 5: Limiting of the input voltage.

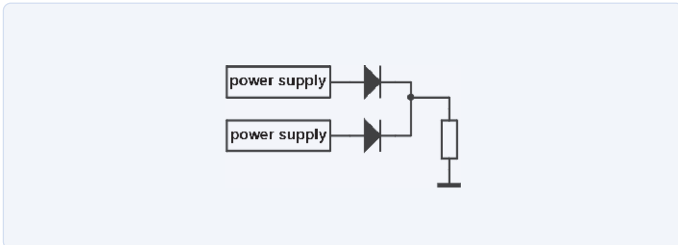


Figure 6: Parallel connection of two main power supplies.

The schematic symbol on the right of **Figure 5** is an operational amplifier, or opamp. These will be extensively dealt with later in this series. For now, we only need to know that opamps will fail if the voltages applied to their inputs are (considerably) higher than the positive power supply voltage or lower than the negative supply voltage. To prevent such mishaps, we can add diodes from the input to the positive and negative power supply rails (as is drawn in Figure 5), which will then divert input voltages that are too large to the power supply.

With this circuit, it is important that the current is limited using a series resistor (on the left in Figure 5). The circuit of Figure 5 is intended for an asymmetric power supply voltage; when a symmetric power supply voltage is used, the anode of the bottom diode is not connected to the ground but to the negative power supply voltage instead.

In stagecraft, mixing panels are generally powered from two power supplies: if one fails, then a second one is on hand to take over the power supply duties — "the show must go on", after all. These power supplies can be connected to different, separately fused circuits of the mains power system.

Now it is not a particularly good idea to connect two DC voltage sources 'just' in parallel; this has to be done via diodes. The source with the highest voltage supplies the current for the load, while the other voltage source (the backup supply) is not loaded.

When both power supplies are operational and supply the same voltage (this should be the case — they are, after all, regulated and compensate for different input voltages), then the total current required by the load will be reasonably well divided equally between the two power supplies.

Power supply circuits for mixing panels generally supply a number of different voltages, each of those will have to be combined with its own diode network (as is drawn in **Figure 6**).

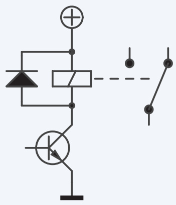


Figure 7: Freewheeling diode.

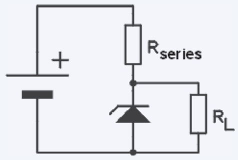


Figure 8: Basic application of a Zener diode.

An inductor, when the current is switched off, generates a peak voltage that can easily reach a value of a few hundred volts – and that is more than sufficient to strike a fatal blow to semiconductors. The polarity of this voltage peak is the opposite of the normal power supply voltage, so that we can, without any problems, short-circuit that with a so-called freewheeling diode, as is drawn in **Figure 7**.

In this figure, the inductor is the coil of a relay; the component at the bottom is the transistor used to turn the relay on and off. It is this transistor that is protected from voltages that are too high by the freewheeling diode.

The Zener Diode

When an ordinary diode is used in the reverse direction, it will start to conduct above the maximum allowable breakdown voltage (and generally give up the ghost). The voltage at which a normal diode breaks over amounts typically to a few hundred volts and generally varies widely between individual examples.

A Zener diode, on the other hand, has a breakdown voltage that (normally) is somewhere between 1 V and 100 V. Furthermore this is a reasonably accurately determined parameter.

In comparison with an ordinary diode, a Zener diode does not fail when it starts to conduct in the reverse direction (of course, you must limit the current that flows in the reverse direction to a reasonable amount). This property of a Zener diode makes it suitable for regulating higher voltages.

A Zener diode must always – as in **Figure 8** – be used in combination with a series resistor; this resistor limits the current through the diode.

That is it for now; in the next installment we will continue with the Zener diode and a few other interesting family members of the diode.

220256-01

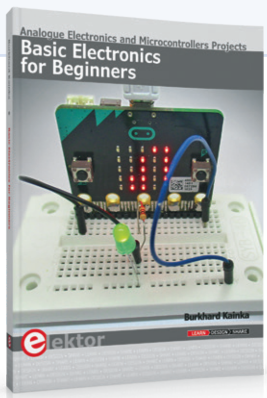
Editor's Note: The series of articles "Starting Out in Electronics" is based on the book *Basiskurs Elektronik*, by Michael Ebner, which was published in German and Dutch by Elektor.

Questions or Comments?

Do you have any questions or comments prompted by this article? Send an email to the editor of Elektor via editor@elektor.com.

RELATED PRODUCTS

- **B. Kainka, *Basic Electronics for Beginners*, Elektor 2020. (SKU 19212)** www.elektor.com/13950
- **B. Kainka, *Basic Electronics for Beginners* (E-Book), Elektor 2020. (SKU 19213)** www.elektor.com/18232



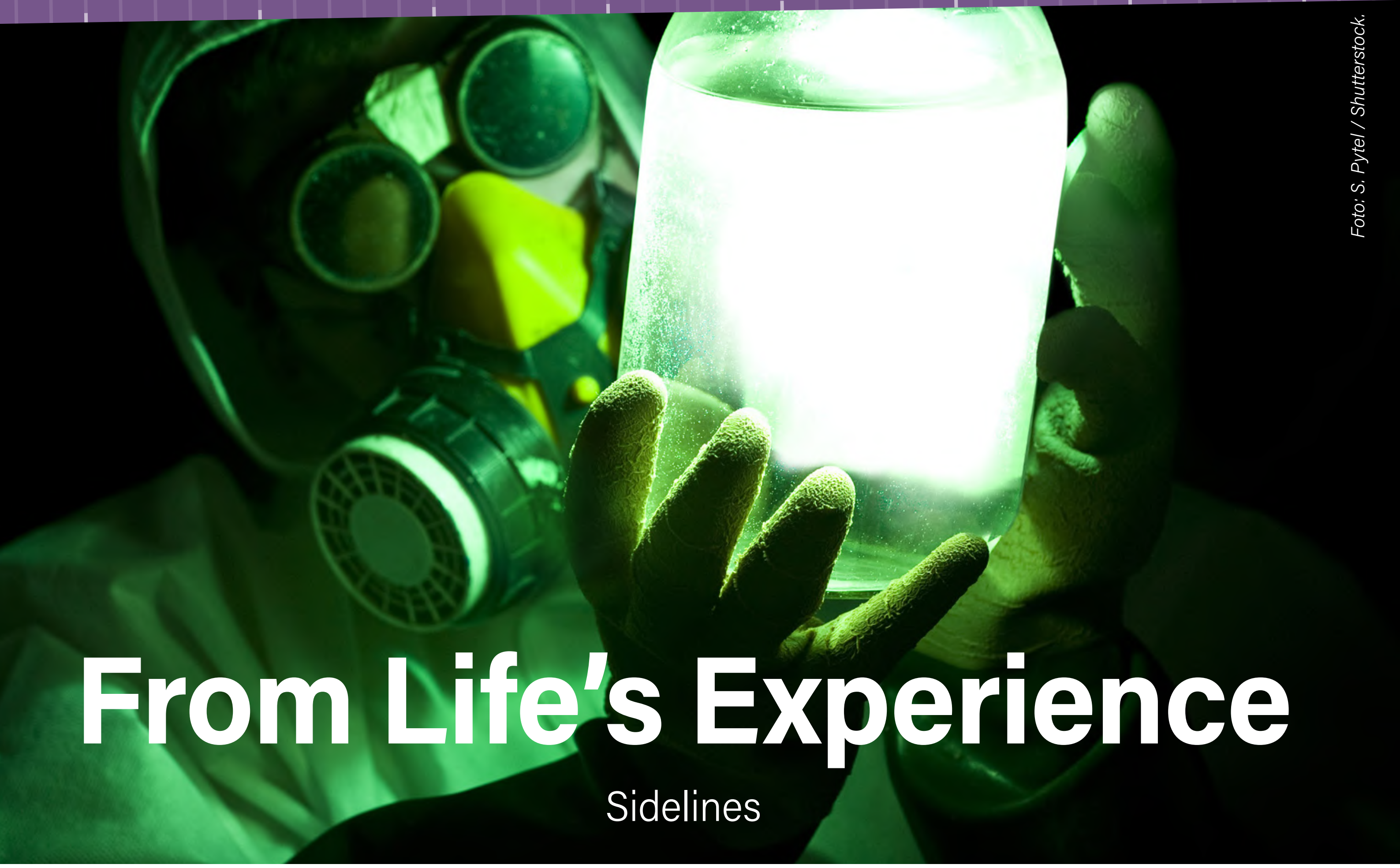


Foto: S. Pytel / Shutterstock.

From Life's Experience

Sidelines

By Ilse Joostens (Belgium)

If you work with electronics professionally and you would also like to make the step to product development, then you need to be a jack of all trades. While for many electronics enthusiasts the focus is mainly on the electronics itself, I see electronics more as part of a larger whole and a means to achieve a particular goal. A more holistic approach therefore.

Throughout the years I have come into contact with various technologies and fields of study, and I have had countless interesting interactions. Naturally a number of eccentric individuals also made their appearance and every now and then exciting or even esoteric subjects came up. Emailing at ungodly hours about the allotropes of plutonium is not something that occupies the average mortal, to give a random example. On more than one occasion, a small amount of knowledge about electronics proved to be extremely handy in disciplines that are not immediately related to electronics, and that closes the loop. Did I not just say that everything is connected to everything else?

Radioactive Man

It so happened that the editor at Elektor received a bulky reader's letter about a year ago, related to the article published in January 2020 about the extendable environmental monitoring system [1]. Although the correspondence with the author of the letter was initially quite stiff, we got to know each other better during the following months. The man turned out to be not only a retired aviation engineer but also a descendant of an old noble family. His father was a nuclear physicist and not only did he have a number of patents to his name, but in the '40s had also contributed to a doctoral thesis related to "the element with alpha radiation with a 1.8 cm range" — plutonium, that is, for the attentive listener. On the final page of the thesis was a drawing of an invention, partially constructed from asbestos, to chemically concentrate the "element" concerned. Well, asbestos and plutonium, what could possibly go wrong? Those were carefree times. Compared to this, the chemicals on the European REACH SVHC list are more suited for wimps. We exchanged quite a few emails around nuclear related subjects and so I came up with the idea of exploring the reception of a university hospital with a small scintillation counter. Just about every time, I encountered radioactive patients and from the gamma spectrum it often revealed what treatment or examination they have had. For example, the 511 keV emission line (positron annihilation) from a PET scan. What medical confidentiality? Each time I automatically thought about the comic hero *Radioactive Man* by Morty Mann, and I wondered whether the people at reception and the staff at the cafeteria inadvertently received a higher dose than the effective allowable limit of 1 mSv/year for the general public.

A scintillation counter [2] is something completely different from the classic Geiger-Müller counter tube and is not only much more sensitive to gamma radiation, but is also much quicker in terms of response time. Unfortunately, the smaller dosimeters with a scintillation crystal come mainly from the country that currently is not considered that friendly. Fortunately, you can buy silicon photo-multiplier chips for very reasonable prices from any of the well-known electronics wholesalers, and my hands are already itching to cobble something together myself. An obstacle is that I do not know where I can obtain small and, especially, affordable scintillation crystals in reasonable quantities. I will therefore continue to dream quietly for the time being.

Psychotronics

My (not that nuclearly active) father used to have an analog mirror reflex camera, and I have to confess that at a much younger age, I was much more involved with photography as a hobby than with electronics. And even now I do a lot of photography and this is very handy because professional product photos of my electronic creations simply improve the sales thereof. The other way around, there are countless applications of electronics within photography, and apparently you can make fantastic photos using high voltage. I therefore managed to acquire the book *Psychotronica* by Matthijs van der Veer and will dip a little into Kirlian photography. Just too be clear, this is about making beautiful photographs and not about the more nebulous subjects that are also extensively covered in that book. By the way, the American Gordon Kirkwood makes sublime photographs using a Marx generator [3] and I quiver at the idea of trying that for myself. I fear that the neighbors will be less appreciative of all the banging and flashes of light.

High voltage I already use for woodworking, for which I built a machine with a flyback transformer and ZVS driver so that I can safely burn Lichtenberg figures in wood (Figure 1) [4] and with all that photography I was also somewhat engrossed in sublimation printing for wall decorations and of course printed utensils such as coasters and mugs. Every time I do sublimation printing of aluminum plates, I think that this is a fantastic method for making front panels for electronics projects. Prior to sublimation, machine the holes and notches with a CNC milling machine and Bob's your uncle!

Figure 1: Lichtenberg generator (Photo: Ilse Joostens).

Roll Up Your Sleeves!

There is more to life for an electronics enthusiast than electronics alone. Therefore dare to experiment with various 'sidelines' besides electronics and give your creativity and imagination a free run. Think outside the box, dare to question the accepted values and rules, be a little more artistic and especially, dare to combine different technologies with each other. The end result will be that much more beautiful and give a great deal more satisfaction than the umpteenth circuit on a breadboard or a circuit board that lies about but never sees an enclosure. Don't hesitate, just do!

220294-01

WEB LINKS

[1] Ilse Joostens, "Extendable Environmental Monitoring System," Elektor 1/2020: www.elektormagazine.com/magazine/elektor-139/56998

[2] Wikipedia: Scintillation counter: https://en.wikipedia.org/wiki/Scintillation_counter

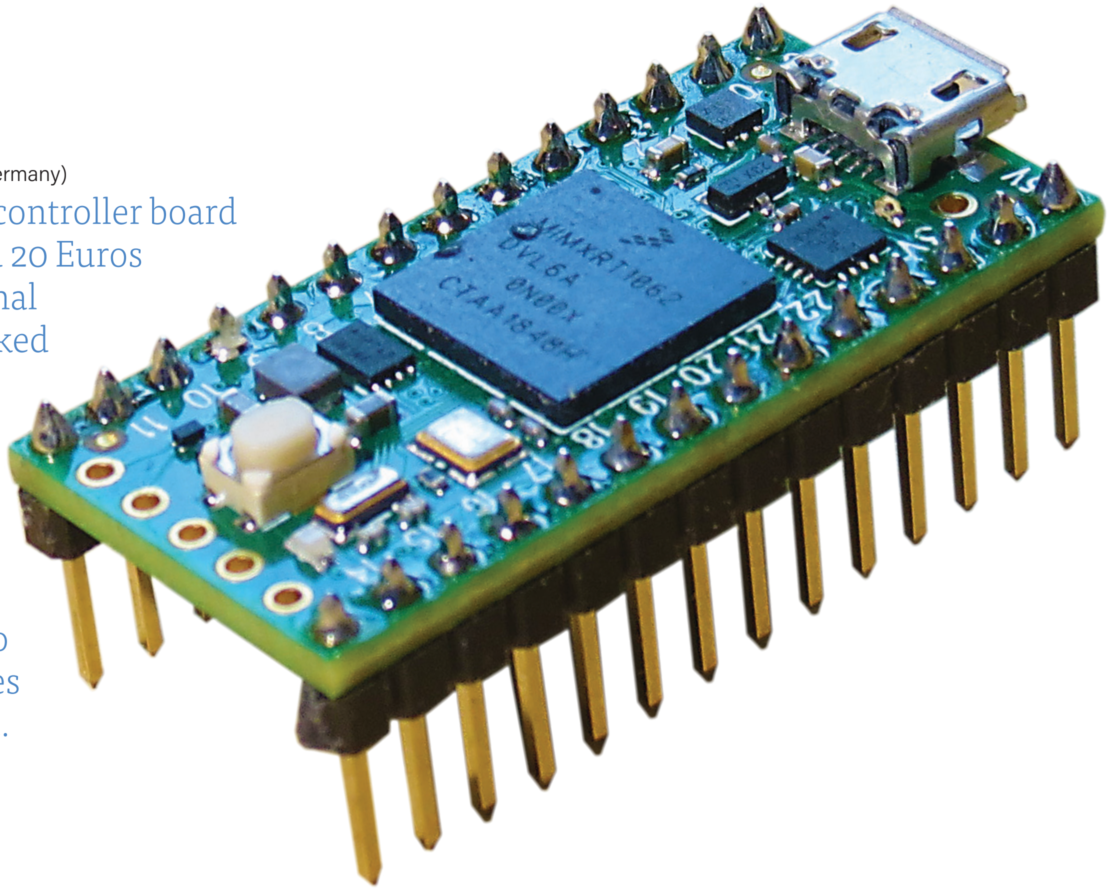
[3] Gordon Kirkwood photography: http://gordonkirkwood.com/photography

[4] Burning Lichtenberg figures in wood: www.youtube.com/watch?v=xmZuidC5qUY

Teensy 4.0
Why Is This Board So Fast?

By **Prof. Dr. Martin Oßmann** (Germany)

The Teensy 4.0 microcontroller board is available for around 20 Euros and delivers exceptional performance. It's clocked at a remarkable 600 MHz, but that's only half the story. Here we probe the board and undertake some simple experiments to find out the techniques used to make it so fast.



The IMX-RT1062 processor from NXP uses an Arm Cortex-M7 CPU and its architecture is closer to that of PC processors than to an AVR microcontroller. Teensyduino can be used as the programming environment, which is largely compatible with Arduino. In this article, however, we will program partly in C or inline assembler.

Pin toggling

The first thing we're going to do is toggle a pin as quickly as possible. To both warm up and provide a comparison, we will first do this using an Arduino Nano clocked at 16 MHz. In terms of the clock rate alone, the Teensy 600/16 is 37.5 times as fast. To toggle the on board LED connected to pin 13 of the Arduino we can execute the code as shown in **Listing 1**.

The intention was to switch the LED on and off for a microsecond. The LED is, however, on for 3.5 µs and off for 3.9 µs. There are two reasons for this. Firstly the `digitalWrite` command has a relatively long execution time (2.5 µs) and, secondly, the Arduino needs to process its own tasks between the calls to `loop`, which require another 0.4 µs. In order to toggle faster, you can and must program 'closer to the hardware', as in **Listing 2**.

With this program, a loop cycle only takes 2.66 µs, which corresponds to six machine cycles at a clock frequency of 16 MHz. To explain why 6 cycles are required, you can look at the associated assembler in **Listing 3**.

The loop consists of three commands. The commands `sbi` and `cbi` respectively set and clear the LED bit. The `rjmp` instruction makes sure the commands are continually executed in an infinite loop. The execution times of all the individual instructions can be found in the AVR technical documentation. Each instruction takes two clock cycles, corresponding to the timing measurements we made.

For this AVR processor it's hardly possible to make any changes to the program so that the output pin toggles any faster. For comparison, we will now turn to the Teensy 4.0 board to see how fast we can make it toggle.

We will first program it in the standard 'Arduino style' (**Listing 4**). One cycle of this

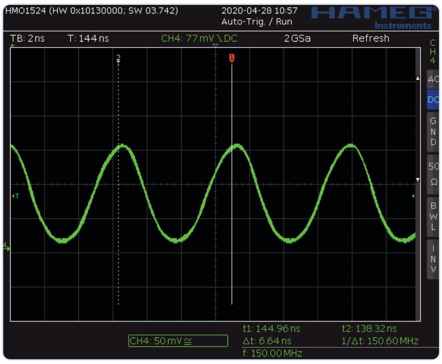


Figure 1: Pin toggling at 150 MHz.

infinite loop takes 135 ns. Compared to the Arduino, this is relatively fast in absolute terms, but this still takes 18 clock cycles which is considerable. The execution of the `digitalWrite` function is, again, relatively slow and the internal 'losses' encountered between the calls to `loop` also seems to be considerable.

When we take a programming approach that is closer to the hardware (**Listing 5**) the time for one pass through the `while` loop is 6.66 ns, i.e. four cycles.

The signal at the LED output pin is shown as the waveform in **Figure 1**. Since the signal has a frequency of 150 MHz it uses the full bandwidth of my oscilloscope, which is struggling to record the fast signal edges and makes the waveform look more like a sine wave than a square wave. In order to get a more precise measurement of the on and off times of the LED, the CPU clock was reduced to 100 MHz. The on and off-time of the LED are both two cycles long (corresponding to 20 ns with the 100 MHz clock). To find out which commands the CPU executes in this loop, let's look at the assembler in **Listing 6** again. The loop consists of three commands (as with the Arduino). Two 'word store' instructions (`str.w`) set and reset bit 3. The loop ends with a branch instruction (`b.n`). Now, it would be interesting to know how many cycles each individual instruction requires, but the Arm documentation does not provide any information about this. This is deliberate because the actual number of cycles depends on many constraints. Thus we should really just look at the overall performance. However, as engineers we are curious and still want to investigate using this simple example. It seems that the execution of the `str` command takes two cycles. You can confirm this by using **Listing 7**.

Listing 1: 'Arduino style' pin toggling.

```
void setup() {
   pinMode(led, OUTPUT);
   }
void loop() {
   digitalWrite(led, HIGH); //3.5 us high time
   delayMicroseconds(1);
   digitalWrite(led, LOW); // 3.9 us low time
   delayMicroseconds(1);
   }
```

Listing 2: Hardware-close pin toggling using the Arduino AVR.

```
#define ledBit 5
 #define ledDDR DDRB
 #define ledPORT PORTB
void setup() {
   cli() ;
   ledDDR |= _BV(ledBit)    ; set output
   while(1){                ; loop time 2.6648 MHz = 6 cycles
     ledPORT |= _BV(ledBit) ; 2 cycles on
     ledPORT &= ~_BV(ledBit); 4 cycles off
     }
   }
```

Listing 3: Assembler listing of the pin toggle routine.

```
342: f8 94   cli           ; cli() ;
344: 25 9a   sbi 0x04, 5   ; ledDDR |= _BV(ledBit) ;
                           ; while(1){ // 2.6648 MHz = 6 cycles
346: 2d 9a   sbi 0x05, 5   ;   ledPORT |= _BV(ledBit) ;
348: 2d 98   cbi 0x05, 5   ;   ledPORT &= ~_BV(ledBit) ;
34a: fd cf   rjmp  .-6     ; jump to 0x346
```

Listing 4: Teensy programming using 'Arduino approach'.

```
int led = 13;

 void setup() {
   pinMode(led, OUTPUT);
   }
void loop(){
   digitalWrite(led,1) ;
   digitalWrite(led,0) ;
   }
```

Listing 5: Hardware-close pin toggling using Teensy.

```
void setup() {
   pinMode(13, OUTPUT);
   cli() ;
   while(1){  // cycleTime 150 MHz = 4 cycles
     CORE_PIN13_PORTSET = CORE_PIN13_BITMASK ;  // on: 2 cycles
     CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK; // off 2 cycles
     }
   }
void loop(){
   }
```

Listing 6: A fast pin-toggling loop in assembler.

```
8c:    f8c2 3084     str.w    r3, [r2, #132]  ; set GPIO pin 13
90:    f8c2 3088     str.w    r3, [r2, #136]  ; clear GPIO pin 13
94:    e7fa          b.n      8c <setup+0x10> ; branch endless loop
```

Listing 7: A sequence using contiguous set/clear instructions.

```
void setup() {
   pinMode(13, OUTPUT);
   cli() ;
   while(1){  // cycleTime 150 MHz = 4 cycles
     CORE_PIN13_PORTSET = CORE_PIN13_BITMASK ;
     CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;
     CORE_PIN13_PORTSET = CORE_PIN13_BITMASK ;
     CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;
     CORE_PIN13_PORTSET = CORE_PIN13_BITMASK ;
     CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;
     . . . jeweils ein set/clear Paar
     }
   }
void loop(){
   }
```

Listing 8: The finite toggle loop.

```
while(1){
   for(int k1=15 ; k1>0 ; k1--){
     CORE_PIN13_PORTSET = CORE_PIN13_BITMASK;
     CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;
     }
   delay(100) ;
   Serial.println("test5\n") ;
   }
```

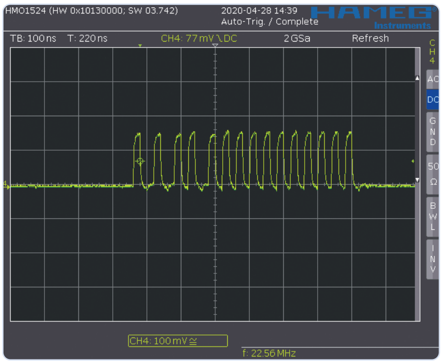


Figure 2: The LED pin toggle timing is not consistent.

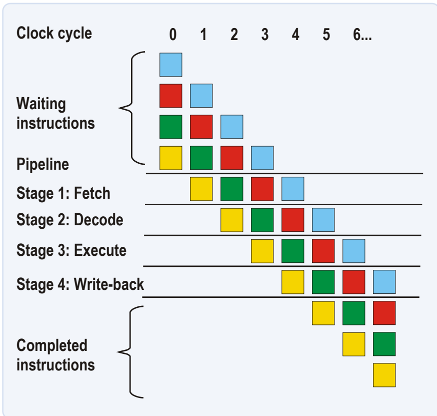


Figure 3: A four-stage pipeline.

Each additional pair of `set/clr` instructions in the execution sequence does not seem to change the frequency of the signal on pin 13. This means that each pair of `set/clr` instructions extends the loop by four cycles. In the earlier example, the two `str` instructions took up four cycles, which is the total time required for the while-loop. It would also seem that no additional CPU time is necessary for the execution of the jump instruction. This seems incredible at first, but we will find out why this is later. As a result, the Teensy board is actually even faster, requiring just four clcok cycles compared to the AVR which uses six.

Let's add to the confusion by executing the `while` loop in **Listing 8**. To make sure the voltage waveform at the LED pin looks more rectangular, we're exectuing the code with a clock frequency of 100 MHz again. The code toggles the LED pin 15 times, pauses, outputs some text using `printLn`, then starts over again.

The voltage at the LED pin then looks like that in **Figure 2**. Amazingly, the duty cycle of the on and off times are not constant. At the beginning, the loop is sometimes longer, sometimes shorter. It appears that the instructions don't have a consistent execution time. Towards the end the cycle time of the loop is four cycles again, exactly as above with the infinite loop. This is all the more astonishing as the loop now contains one more instruction, namely `k1--`. However, this does not seem to need any additional processor time, so it does not affect the loop timing.

Jump prediction

A technique known as 'branch prediction with speculative execution' provides an initial explanation for the observed high loop speed. To understand how this technique works, we first need to look at how a CPU executes instructions. The first method to improve throughput is to use instruction pipelines. The instructions that are currently being executed are in a pipeline. Each instruction is processed in several stages. For example, if Instruction 1 is in stage 4, Instruction 2 is being processed in stage 3, Instruction 3 in stage 2, and Instruction 4 is in stage 1 (a four-stage pipeline as shown in **Figure 3**). The instructions are fetched from the memory into the pipeline and the memory access unit has to ensure that the pipeline is always filled. To do this, it always fetches the next instructions into the pipeline.

Now, there are situations in which the contents of the pipeline have to be 'thrown away' because they are incorrect. This is the case when a (conditional) jump is encountered, causing a deviation from the current execution of the linear instruction sequence . Such *pipeline stalls* can also arise when the stages of a pipeline depend on a result that another stage must calculate first. In order for the pipeline to be correctly filled in the event of conditional jumps, the CPU would need to know whether a jump is being executed or not. It cannot know for sure, but the CPU can try to predict it. In our example (**Listing 9**), the branch condition in the loop means that the jump is taken 14 times and ignored just once at the end when the loop counter `k1` is zero. During the first run through the loop, the CPU observes what the jump is doing. It notices that the jump is taken the first time through the loop and assumes that the next time it encounters the branch instruction the jump will again be taken.

The pipeline is therefore always loaded with the `str` instruction after the jump instruction. The the CPU is always correct when the program begins executing, but not at the end when a stall occurs. This technique is called *branch prediction with speculative instruction execution*. Since the pipeline is usually filled correctly, the jump instruction no longer wastes any CPU time. This explains the program behavior of **Listing 9**. During the first pass, the branch prediction will not be correct, but overall it leads to an extremely fast loop execution. One consequence of this is that a cycle-accurate calculation of the execution time of a piece of code during a branch prediction is very difficult because the behavior of the prediction depends on many things. Now we know why there are no cycle times specified for individual instructions of M7-type Arm processors. Good branch prediction is a crucial architectural feature of high-performance CPUs. Since modern CPUs often have relatively extensive pipelines, a stall is quite 'expensive' and a correct jump prediction is essential. Today's *branch-predictors* can achieve a hit rate of over 98%! Smaller processors and microcontrollers (such as AVR controllers for example) often do not employ branch prediction, but pipelines are now more or less standard.

Next, let's look at the timing of some more complicated pieces of code. Our first program calculates a scalar product in a loop (**Listing 10**). The calculation is performed using the statement `skp += x[k]*y[k]`. Prior to this instruction we set the LED pin 13. We then reset the pin immediately after the calculation is finished. The on-time of pin 13 should be precisely the execution time of our test instruction.

Amazingly, we hook up a scope to pin 13 and measure that the on-time is just two cycles long. This time period is just the time needed to set the GPIO pin, so the test instruction has been ignored. Something odd is going on here.

To get to the bottom of the problem let's take a look at the equivalent assembler listing (**Listing 11**).

Interestingly, there is only a single instruction between the port instructions. The compiler moved the accesses to the variables `x[k1++]` and `y[k2++]` prior to the port set instruction. This means we can't measure the timing of interest by looking at the LED signal. The compiler just doesn't always generate the code we expect. Instructions are sometimes also rearranged if this is correct in the context of the code to achieve better performance. When you need to make measurements of a processor's activity, it's necessary to make sure the compiler has not rearranged or 'optimised' the code.

Incidentally, you can see the total number of cycles that have elapsed in the `ARM_DWT_CYCCNT` variable and use that to measure the timing. In our example I measure the total time for the loop (0–1000). The

Listing 9: Assembler code version of the finite toggle loop.

```
ca: 230f       movs    r3, #15             ; k1=15
cc: 3b01       subs    r3, #1              ; k1--
ce: f8c5 4084 str.w    r4, [r5, #132]      ; set pin 13
d2: f8c5 4088 str.w    r4, [r5, #136]      ; clear pin 13
d6: d1f9       bne.n   cc <test5()+0x10>   ; if !=0 springe nach cc
```

Listing 10: Test sequence using scalar product calculation.

```
cyclesStart = ARM_DWT_CYCCNT ;
 skp=0 ;
 for(int k=0 ; k<nn ; k++){
   CORE_PIN13_PORTSET = CORE_PIN13_BITMASK;
   skp += x[k]*y[k] ;
   CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;
   }
 cyclesStop = ARM_DWT_CYCCNT ;
```

Listing 11: Assembler listing of the scalar product.

```
for(int k=0 ; k<nn ; k++){

a0: ecf3 6a01 vldmia    r3!,             ; x[k1++]
a4: ecb1 7a01 vldmia    r1!,             ; y[k2++]
a8: 42a3      cmp       r3, r4           ; k1==1000 ?
aa: f8c2 0084 str.w     r0, [r2, #132]   ; set pin 13
ae: eee6 7a87 vfma.f32  s15, s13, s14    ; skp += x[k]*y[k] ;
b2: f8c2 0088 str.w     r0, [r2, #136]   ; clear pin13
b6: d1f3      bne.n     a0               ; branch on not equal
```

Listing 12: Addition of 100 values.

```
int NN=100 ;
CORE_PIN13_PORTSET = CORE_PIN13_BITMASK;
 sum=0 ;
 for(int k=0 ; k<NN ; k++){
   sum += fun1(k) ;
   }
 CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK;

 int fun1(int x){
   return 100*x+x*x+32 ;
   }
```

Listing 13: Assembler code version of the addition loop.

```
1ea: f8c6 3084 str.w  r3, [r6, #132]    ; set pin13
1ee: 4602      mov    r2, r0
1f0: f8c5 9000 str.w  r9, [r5]          ; sum=0x000C9CB6H
1f4: f8c6 3088 str.w  r3, [r6, #136]    ; clear pin13
```

Listing 14: A complex loop.

```
void loop(){
   int xx=42 ;
   int nn=256 ;
   int k ;
   int m=0 ;
   int vv=0 ;
   while(1){  // 5 cycles, 9 cycles if dualIssueDisabled
     cyclesStart = ARM_DWT_CYCCNT ;
     for( k=0 ; k<nn ; k++){
       CORE_PIN13_PORTSET = CORE_PIN13_BITMASK; // led-1
       CORE_PIN13_PORTCLEAR = CORE_PIN13_BITMASK; // led-2
       xx *= 105529 ;
       vv += m & 0x1234 ;
       m +=17 ;
       }
     cyclesStop = ARM_DWT_CYCCNT ;
    ...
```

Listing 15: Assembler version of Listing 14.

```
// r9=105529 ; r3=m ; r5=vv ; r6=xx

1d8: f241 2234  movw  r2, #4660       ; r2=0x1234
1dc: f8c8 7084  str.w r7, [r8, #132]  ; set pin 13
1e0: fb09 f606  mul.w r6, r9, r6      ; xx *= 105529 ;
1e4: 401a       ands  r2, r3          ; r2= m & 0x1234 ; r2=1234h r3=m
1e6: 3311       adds  r3, #17         ; m +=17 ;
1e8: f8c8 7088  str.w r7, [r8, #136]  ; clear pin 13
1ec: 4299       cmp   r1, r3          ; abbruchbedingung r1 <> m
1ee: 4415       add   r5, r2          ; r5 += m & 0x1234 ; r5=vv
1f0: d1f2       bne.n 1d8 <loop+0x1c> ; loop weitermachen
```

Listing 16: Dynamically allocated memory in Teensy RAM2 memory area.

```
uint8_t *RAM2buffer ;
 RAM2buffer=(uint8_t *)malloc(NN) ;
cyclesStart = ARM_DWT_CYCCNT ;
   int sum=0 ;
     for(int k=0 ; k<NN ; k++ ){
        sum += RAM2buffer[k] ;
       }
   cyclesStop = ARM_DWT_CYCCNT ;
```

number of cycles obtained divided by 1000 then gives the number of cycles per loop. In this case the loop is seven cycles long. The next exercise is to measure the total time it takes to execute the loop (k = 0–100) in the program shown in **Listing 12**. We have positioned `PORTSET` and `PORTCLEAR` instructions directly before and after the loop while the function value `fun1(k)` is summed in the loop.
Incredibly, we measure a time of 5 ns (corresponding to 3 cycles) for the entire loop. But that can't be. A look at the equivalent assembler code version in **Listing 13** provides an explanation.

Compiler optimisations

The compiler has just placed two simple instructions between the port set and reset instructions and ignored the loop entirely — the entire loop structure is nowhere to be seen in the assembler listing. If we analyze this more closely, we find the compiler eliminated the loop and replaced it with the assignment `sum=826550` (= 0x000C9CB6H). The compiler has actually calculated the resulting value in advance and replaced 100 passes through the loop with an assignment. This shows that optimising compilers are now able to optimise relatively complex code quite effectively. The executed code can therefore differ selectively from the source code. Again, you have to be careful when measuring execution times so that you don't end up comparing apples with oranges.
Now, let's run the loop shown in **Listing 14**. The Teensy board uses five cycles for each pass of the loop.
That seems relatively short. So if we look at the assembler **Listing 15** again, we can see that the loop consists of nine instructions. The Teensy Board manages these nine instructions in just five cycles, which at first seems barely credible. This level of performance is, however, achieved because the Teensy processor is a 'Superscalar CPU with dual issue'. In the case of a superscalar CPU, a number of the functional units (adders, etc.) are duplicated several times, allowing the CPU to process several subtasks from several instructions at the same time. 'Dual issue' in this context means that two instructions from the first part of the FIFO are simultaneously transferred, or issued, to the superscalar units. The pipeline can be seen in **Figure 4**. Two instructions can be sent from the issuing unit to the subsequent units at the same time. If we switch off the 'dual issue' feature, the loop above needs nine cycles for the nine instructions, which is almost twice as long as when 'dual issue' was enabled. This example shows how successful and important such optimisation techniques can be.
Now we will examine the performance using a realistic application, namely by implementing a fast Fourier transform or FFT. First, we take a 128 point FFT. Using the Arduino this routine takes about 50 ms, but on the Teensy 4.0 it only requires 23 µs. This works out at a factor of 50,000 µs / 23 µs =

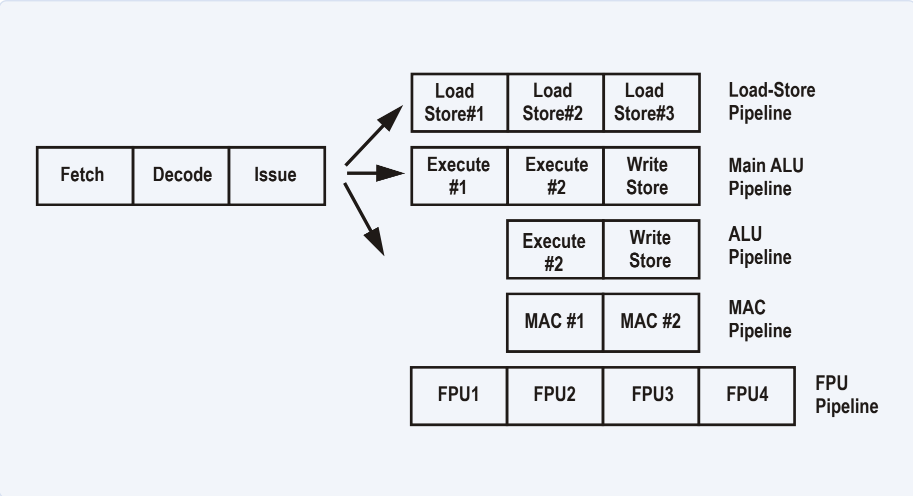
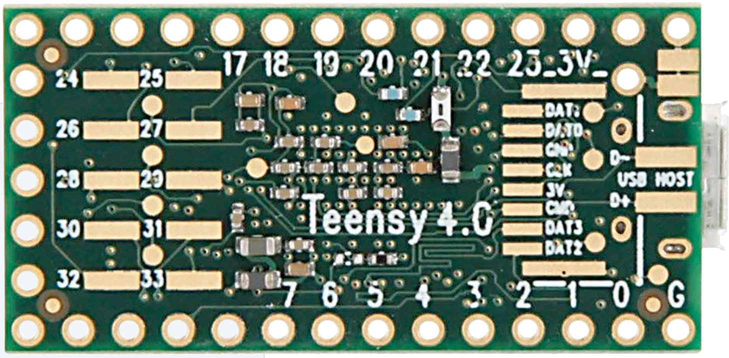


Figure 4: Dual-issue pipeline.

CONCEPTS USED BY THE IMXRT1062-CPU

- High clock rate
- Optimising compiler
- Speculative execution
- Branch prediction
- RISC Architecture
- Pipeline
- Superscalar architecture
- Dual-issue
- Instruction cache
- Data cache
- Floating point unit



RELATED PODUCTS

- **Teensy 4.0 Development Board**
  www.elektor.com/teensy-4-0-development-board
- **Teensy 4.1 Development Board**
  www.elektor.com/teensy-4-1-development-board

2,200 times faster! Of this, the difference in clock rate only accounts for a factor of 600 MHz / 16 MHz = 37; the remaining factor of 2200/37 = 60 is achieved with assistance of the floating point unit and other architectural features inside the Teensy. Since these various architecture tricks (branch prediction, dual issue, etc.) can also be switched off, it is possible to determine more precisely how much influence they really have.

The branch prediction produces an acceleration of 21%, while the dual-issue concept provides a 41% boost. This shows that two instructions are superscalar and processed relatively often. In our example, the instruction cache does not provide any acceleration but that's because the code is in fast memory. If the code were to be put in the relatively slow flash memory, and cachine were to be switched off, the routine would be slowed by a factor of 7. When the code is resident in flash, instruction caching provides a factor of 7 boost. Since data is stored in fast RAM, the data cache does not provide any additional improvement. Memory areas dynamically allocated via `malloc()` are located in a slower memory area (RAM2). When you access data there with the cache switched off, the program (**Listing 16**) slows down by a factor of 7 because RAM2 is accessed via a slower bus.

With this we come to the end our tour through the concepts of processor performance enhancement. The concepts implemented in Teensy are listed once more in the **text box** accompanying this article. Perhaps someone will investigate the same thing using the more recent versions of the Raspberry Pi that are clocked at 1.5 GHz. There are still some techniques, such as *out of order execution*, *paging memory management unit*, *L2- and L3-caches*, *simultaneous multithreading* or *register renaming* features as found in some PC-like CPUs that can provide an extra performance boost.

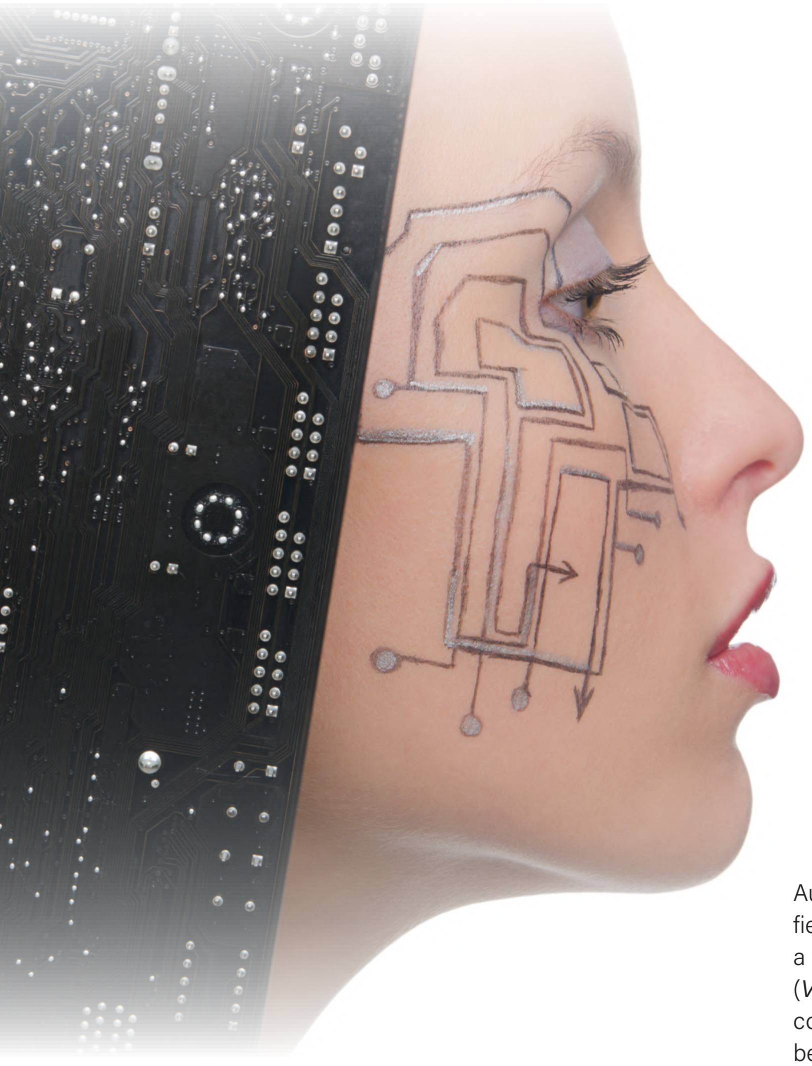
200190-03

Questions or Comments?
The author looks forward to any questions or comments you may have regarding this article. You can reach him at ossmann@fh-aachen.de.

Contributors
Idea, development and text: **Prof. Dr. Martin Oßmann**
Translation: **Martin Cooke**
Editing: **Stuart Cording**
Layout: **Giel Dols**

Audio Power Amplifier Simulation with TINA

The Try-Before-You-Build Approach



By Dogan Ibrahim (United Kingdom)

Elektor's Summer Circuits (see last edition) are traditionally small projects, always inviting to experimentation and enhancement. While you can test most of them by soldering or breadboarding parts on a Sunday afternoon, consider running a simulation of these circuits in virtual operation. Here we show the use of TINA or TINACloud to comprehend, test, and analyze a simple audio power amplifier without the risk of solder fumes, smoke, or blown fuses.

Audio power amplifiers are essentially the final stages of audio amplifiers. They are designed to faithfully amplify power (*P*). By contrast, a voltage amplifier is designed for optimum and maximum voltage (*V*) amplification. The outputs of audio power amplifiers are usually connected to loudspeakers. **Table 1** shows the main differences between a voltage amplifier and a power amplifier.

Audio power amplifiers are available in, and specifically designed for, different configurations usually called Class A, Class B, Class AB, Class C, etc., with the following main characteristics:

- **Class A**: Here, the transistor operating point is near the center of the supply voltage. The transistor conducts during the complete 360° phase cycle of the audio signal. The main advantage of Class A amplification is that the output signal is virtually distortion-free. However, the circuit suffers from poor efficiency.

Table 1: Differences between voltage and power amplifiers.

Parameter	Voltage amplifier	Power amplifier
Coupling	Usually, R-C	Usually, transformer
Input voltage	A few mV	Usually, 2 to 4 V
Collector current	A few mA	More than 100 mA
Load resistance	Several kilo-ohms	4 to 20 Ω
β	More than 100	4 to 30
Voltage gain	high	low
Power output	low	high

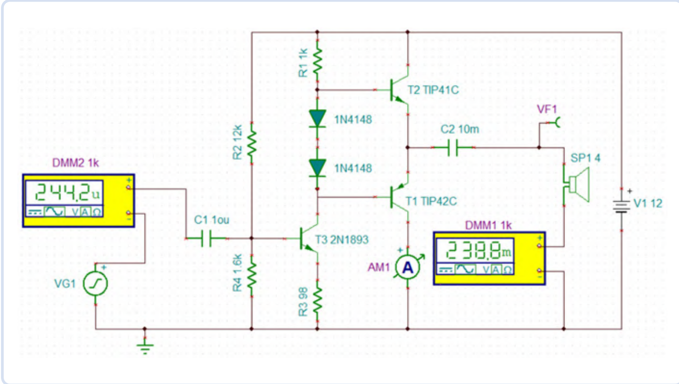


Figure 1: TINA-generated schematic of the Class AB audio power amplifier.

- **Class B**: In Class B operation, two transistors are used with their operating points located at the cut-off point. As a result, one transistor amplifies the signal over one half of the audio input waveform, while the other transistor amplifies during the other half. Because of cut-off biasing, the quiescent current is zero when there is no input and, therefore, no power is dissipated in this situation. Class B configuration suffers from inherent distortion, especially at the crossover point of the input signal.
- **Class AB**: This configuration is similar to Class B, but the distortion is improved by applying a small bias above the transistors' conduction thresholds. Class AB is the most commonly used audio power amplifier configuration.

Simulation of a Class AB Audio Power Amplifier

Figure 1 shows the circuit schematic of a typical Class AB amplifier, consisting of a pair of matched NPN and PNP (i.e., complementary) transistors. The circuit diagram is generated with TINA or TINACloud for the purpose of simulation. Assuming you have TINA or TINACloud installed, you can draw the schematic yourself or import the file *sim14* from the program download available at [1]. Once on that page, first marvel at the author's (e-)book, and then scroll to *Downloads*. Next, click on *Software_Circuit Simulation with TINA Design Suite & TINACloud* (767.94 kB). Save the file locally, unzip it, save the result in a suitably named folder, and then locate the simulation file called *sim14*.

A 4-Ω virtual loudspeaker is connected at the output of the circuit. Two virtual ammeters are connected to measure the current (RMS, root-mean-square) at the input and output of the circuit, respectively. The components used — again, they're virtual — are listed in **Table 2**.

The operation of the circuit is as follows. Two diodes are used to supply DC bias to the transistors. T3 provides current through these diodes. The quiescent output voltage is set at approximately $V_{CC}/2$. T3 acts like a small-signal, common-emitter amplifier and buffer, driving the bases of T1 and T2. At the positive half of the input signal, T1 is driven, while T2 is held OFF. Similarly, at the negative half of the input signal, T2 is driven, while T1 is held OFF.

Table 2: Components used for: sim14 — Class AB amplifier simulation.

Component	Value (TINA units)
VG1	400 m sine wave, f = 1k
C1	10 µ
C2	10 m
R1	1 k
R2	12 k
R3	98
R4	1.6 k
D1, D2	1N4148
T1	TIP41C (NPN)
T2	TIP42C (PNP)
SPI	4 Ω speaker
V1	12 V battery

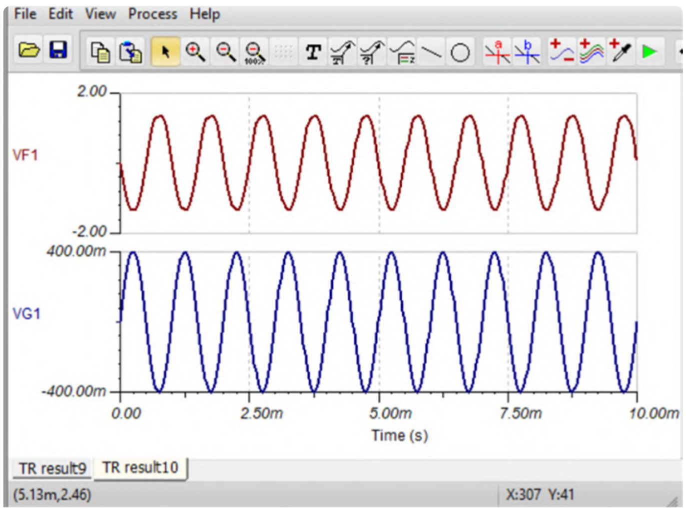


Figure 2: Input and output voltages on TINA's virtual two-trace 'scope.

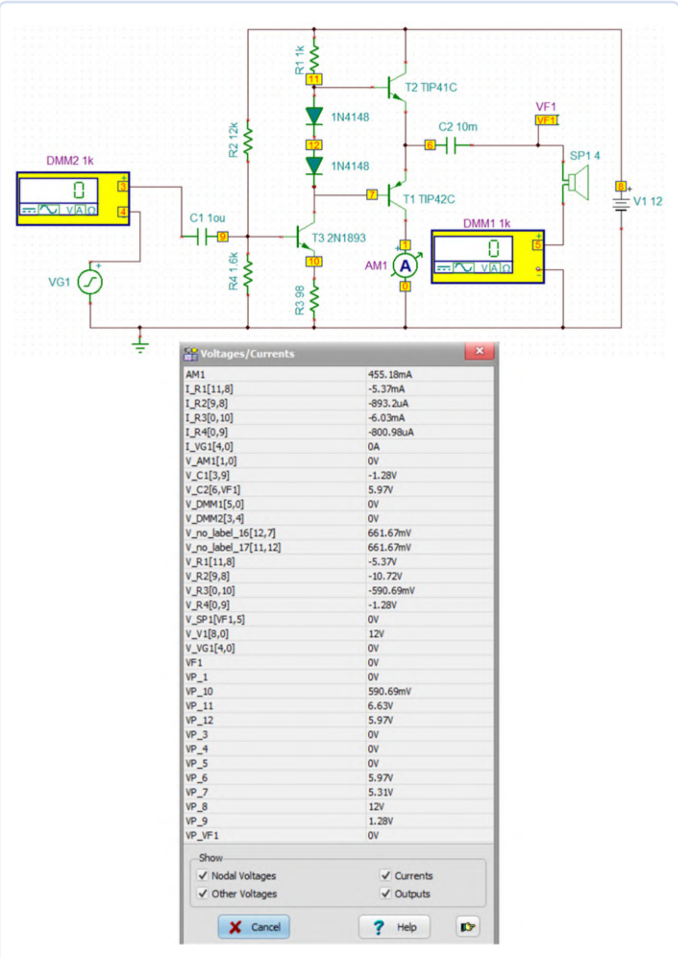
The TINA Simulation

Firstly, let's look at the input and output waveforms. Click *Analysis Transient*. Set the *Start* and *End* display to 0 and 10m. **Figure 2** shows the waveforms on the virtual oscilloscope where the ammeter 'readouts' have been deleted. The input voltage (bottom curve) is 800 mV peak-to-peak, and the output voltage (top curve) is about 3 V peak-to-peak, resulting in a voltage gain of less than 4.

Now, let's look at the DC currents in the circuit. Click *Analysis* ➡ *DC Analysis* ➡ *Table of DC results*. **Figure 3** shows the DC voltages and currents in the circuit as measured by the TINA software.

Arguably, it is more interesting to know the AC voltages and currents in the circuit. Click *AC* in interactive mode. The input current (244.2 μA_{rms}) and output current (238.8 mA_{rms}) will be displayed by the two ammeters pictured in Figure 1.

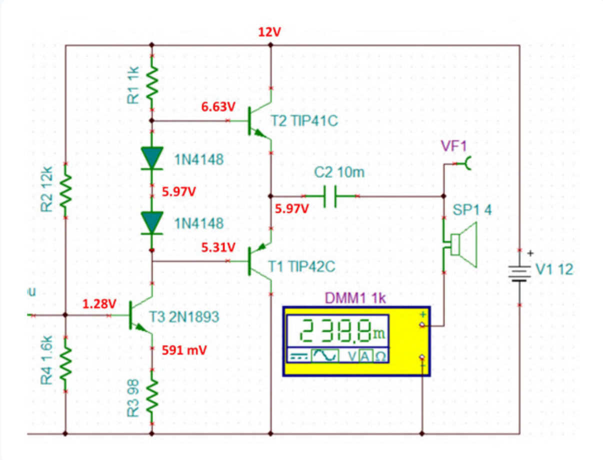
A tabular format of the AC currents and voltages is obtained by clicking *Analysis* ➡ *AC Analysis* ➡ *Table of AC results*. **Figure 4** shows the AC results.



AM1	455.18mA
I_R1[11,8]	-5.37mA
I_R2[9,8]	-893.2uA
I_R3[0,10]	-6.03mA
I_R4[0,9]	-800.98uA
I_VG1[4,0]	0A
V_AM1[1,0]	0V
V_C1[3,9]	-1.28V
V_C2[6,VF1]	5.97V
V_DMM1[5,0]	0V
V_DMM2[3,4]	0V
V_no_label_16[12,7]	661.67mV
V_no_label_17[11,12]	661.67mV
V_R1[11,8]	-5.37V
V_R2[9,8]	-10.72V
V_R3[0,10]	-590.69mV
V_R4[0,9]	-1.28V
V_SP1[VF1,5]	0V
V_V1[8,0]	12V
V_VG1[4,0]	0V
VF1	0V
VP_1	0V
VP_10	590.69mV
VP_11	6.63V
VP_12	5.97V
VP_3	0V
VP_4	0V
VP_5	0V
VP_6	5.97V
VP_7	5.31V
VP_8	12V
VP_9	1.28V
VP_VF1	0V

Figure 3: DC voltages and currents in the amplifier circuit.

A Little Theory



DC voltages of the circuit as calculated by TINA.

1. DC analysis

The direct voltages at various points of the circuit are calculated by TINA as shown in the figure.

The voltage at the base of T3 is given by:

$$\frac{R4}{R4 + R2}(V_{cc} - 1.4) = \frac{1.6}{1.6 + 12}(12 - 1.4) = 1.24\ [V]$$

Overview of in-circuits voltages:

T2 emitter voltage, VE2 = 1.24 – 0.7 = 540 mV

T2 emitter current, IE2 = 0.540 / 98 = 5.5 mA

T1 base voltage, VB2 = VCC – (IE2 × R1 + 1.4 + VE2) = 4.56 V

T2 base voltage,VB2 = VCC – IE2 × R1) = 6.5 V

T1 and T2 emitter voltages, VE1 = VB2 – 0.7 = 5.93 V

Notice that the voltage at the emitters of T1 and T2 is nearly equal to V_{CC} / 2. The theoretical results are very close to the ones calculated by TINA.

2. AC analysis

The AC output power, P_o, of the amplifier is given by:

$$P_O = \frac{{V_L}^2}{R_L}\ \ [W]$$

where V_L is the RMS value of the load voltages.
From the AC analysis, V_L = 0.955 V, and R_L = 4 Ω.
Therefore, P_O = 228 mW.
The input AC power, P_i, is calculated from:

$$P_i = 0.4 \times 0.244 = 0.0976\ [mW]$$

The power gain works out at 228 / 0.0976 = 2333 or 33.6 dB.

Test Before You Build

Both TINA and TINACloud are perfect for a quick test-before-you-build analysis of many electronic circuits. Both programs are highly educational and intuitive and can save a lot of time and money juggling with real components, although those will remain the definite elements to determine the "real life" operation of a circuit when properly assembled. Also properly done and dusted, the simulation with TINA will have a high predictive value of the effects noticed after flicking the power-ON button. If this has you hooked, another, more extensive example of the use of TINA or TINACloud was discussed by the author in an earlier article [2].

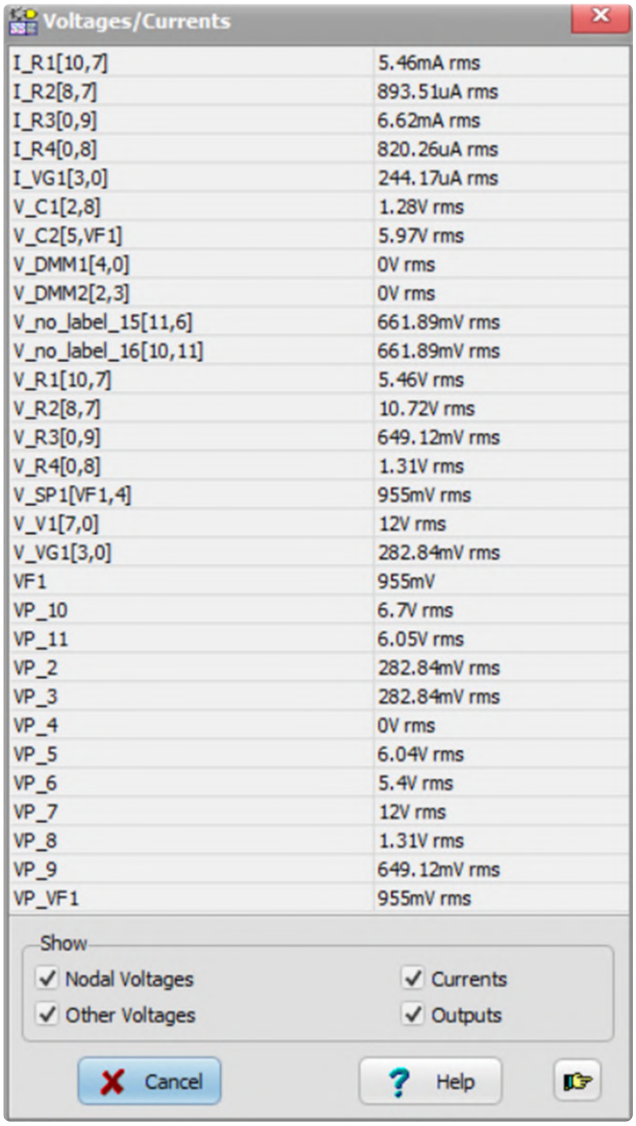
220215-01

About the Author

Prof. Dr. Dogan Ibrahim has a BSc, Hons. degree in Electronic Engineering, an MSc in Automatic Control Engineering, and a PhD in Digital Signal Processing. Dogan worked in many industrial organizations before he returned to academic life. He is the author of over 70 technical books and has published over 200 technical articles on electronics, microprocessors, microcontrollers, and related fields.

Questions or Comments?

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



Voltages/Currents	
I_R1[10,7]	5.46mA rms
I_R2[8,7]	893.51uA rms
I_R3[0,9]	6.62mA rms
I_R4[0,8]	820.26uA rms
I_VG1[3,0]	244.17uA rms
V_C1[2,8]	1.28V rms
V_C2[5,VF1]	5.97V rms
V_DMM1[4,0]	0V rms
V_DMM2[2,3]	0V rms
V_no_label_15[11,6]	661.89mV rms
V_no_label_16[10,11]	661.89mV rms
V_R1[10,7]	5.46V rms
V_R2[8,7]	10.72V rms
V_R3[0,9]	649.12mV rms
V_R4[0,8]	1.31V rms
V_SP1[VF1,4]	955mV rms
V_V1[7,0]	12V rms
V_VG1[3,0]	282.84mV rms
VF1	955mV
VP_10	6.7V rms
VP_11	6.05V rms
VP_2	282.84mV rms
VP_3	282.84mV rms
VP_4	0V rms
VP_5	6.04V rms
VP_6	5.4V rms
VP_7	12V rms
VP_8	1.31V rms
VP_9	649.12mV rms
VP_VF1	955mV rms

Figure 4: AC voltages and currents in the circuit.

WEB LINKS

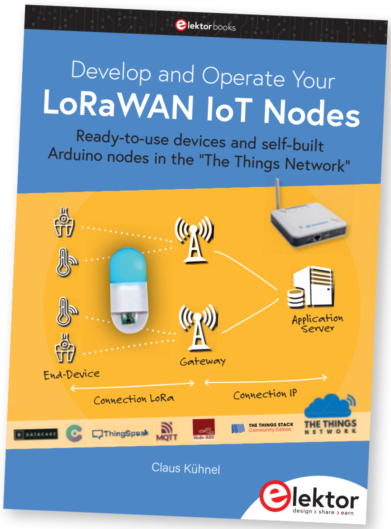
[1] TINA / TINACloud Simulation Files: www.elektor.com/circuit-simulation-with-tina-design-suite-tinacloud
[2] D. Ibrahim, "Circuit Simulation with TINA Design Suite & TINACloud," Elektor 5-6/2022: www.elektormagazine.com/220025-01

Develop and Operate Your LoRaWAN IoT Nodes

Sample Chapter: Dragino LHT65, LDS01, and LDS02 LoRaWAN Modules

By Dr. Claus Kühnel (Switzerland)

From a circuit point of view, a LoRaWAN node combines a microcontroller and LoRa transceiver. Although there are numerous commercial LoRaWAN sensor nodes designed for "industrial use," that should encourage rather than deter the keen experimenter. In this article, *Elektor* book author Claus Kühnel shows the relative ease of making three Dragino sensor modules "talk" within a LoRaWAN.



Editor's Note. *This article is an excerpt from the 224-page book* Develop and Operate your LoRaWAN IoT Nodes *(Elektor, 2022). The excerpt was formatted and lightly edited to match Elektor Mag's editorial standards and page layout. Being an extract from a larger publication, some terms in this article may refer to discussions elsewhere in the book. The author and editor have done their best to preclude such instances and are happy to help with queries. Contact details are in the* **Questions or Comments?** *box.*

Commercially available LoRaWAN (Long-Range Wireless Area Network) devices are available from numerous manufacturers in a daunting variety and at very different prices. You can find a good overview of the range of LoRaWAN end devices at the well-known distributors and on various websites, some of which I am listing here at random:

- *www.thethingsnetwork.org/marketplace/products/devices*
- *https://iot-shops.com/product-category/lora/lorawan-devices/*
- *www.lora-shop.ch/*
- *https://lorawan-webshop.com/shop/10-lorawan-devices/*
- *https://smartmakers.io/iot-sensoren-uebersicht/*

Regarding the ready-made devices presented in the book, I made sure that they are in a price segment acceptable for makers so that nothing stands in the way of the experimental development of these sensors.

Dragino LHT65 Temperature & Humidity Sensor

The Dragino LHT65 temperature and humidity sensor is a LoRaWAN sensor offering long-range transmission. It contains a built-in SHT20 temperature and humidity sensor from Sensirion and a connection for external sensors measuring temperature, soil moisture, tilt, etc.

For example, you can connect a waterproofed DS18B20 with a connection cable for measuring the outside temperature via this external connection (**Figure 1**).

You can find all necessary information about the sensor node in the manual for the Dragino LHT65 at web reference [1]. The connection of additional external sensors is described there too.

The LHT65 has a built-in, non-rechargeable 2400 mAh battery. The expected battery life is more than ten years. The node is fully compatible with the LoRaWAN v1.0.2 protocol. Operation with any standard LoRaWAN gateway is possible. **Table 1** summarizes the technical data of the Dragino LHT65-EU868-E1.

Figure 1: Dragino LHT65-EU868-E1 with an external DS18B20.

Figure 2: EUIs and software keys for the LHT65.

The Dragino LHT65 package includes a sticker with the necessary data for registering the end device at the LoRaWAN server (**Figure 2**) and a special programming cable (**Figure 3**). Keep both in a safe place! It would be best to remember the EUIs and keys for registration and subsequent adjustments.

Simply use AT commands to configure the LHT65. Firmware updates and configuration via serial interface calls for the programming cable connected to the LHT65 via the contact areas on the back of the LHT65.

The LHT65 User Manual contains the firmware update description using an ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 and goes into great detail [2]. That's why I will not cover it here.

The important thing is configuration via AT commands using the programming cable. Due to the realized interface, it needs an FTDI-USB-UART converter (**Figure 4**). A terminal program establishes a connection. Only the lines Rx, Tx, and GND have to be connected between the programming cable and the FTDI-USB-UART converter.

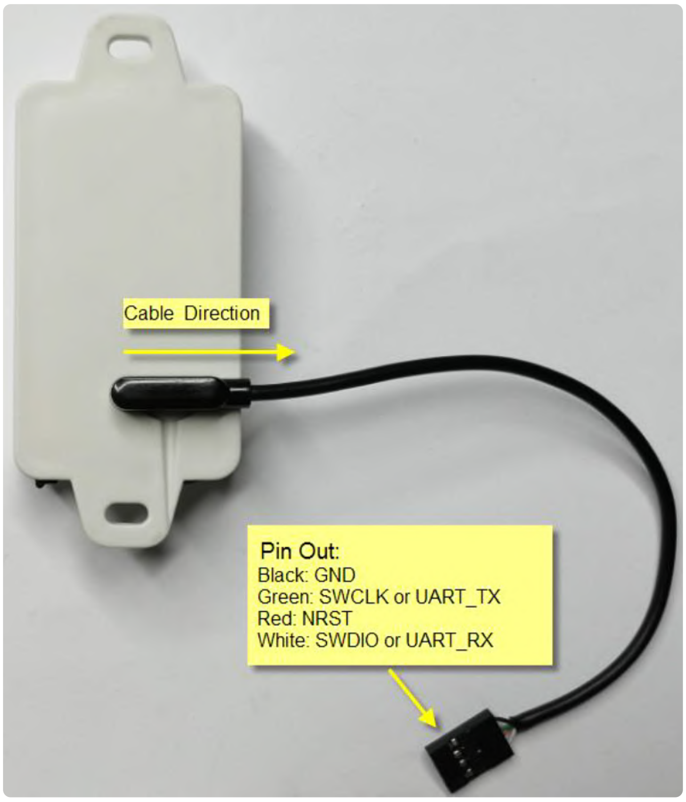


Figure 3: LHT65 programming cable.

Table 1: Dragino LHT65-EU868-E1 – Technical Specifications

Temperature Sensor (internal)	Resolution	0.01 K
	Accuracy	± 0.8 K
	Long-term Drift	< 0.02 K/a
	Measuring Range	−40 °C – +80 °C
Humidity Sensor (internal)	Resolution	0.04% rH
	Accuracy	±10% rH
	Measuring Range	0 – 99.9% rH
	Response Time	< 5 s
External DS18B20 Temperature Sensor	Resolution	0.0625 K
	Accuracy (−10 °C – +85 °C)	±0.5 K
	Accuracy (−55 °C – +125 °C)	±2 K
	Measuring Range	−55 °C – +125 °C

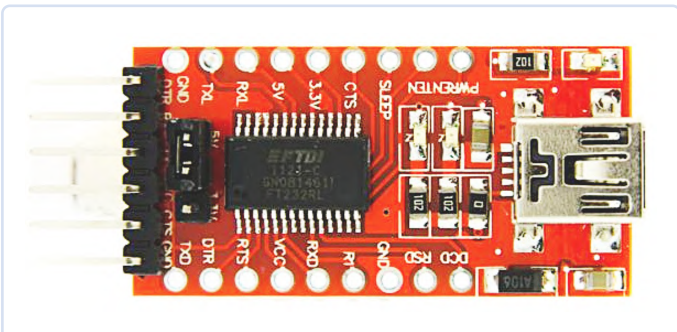


Figure 4: FTDI-style USB-UART cnverter.

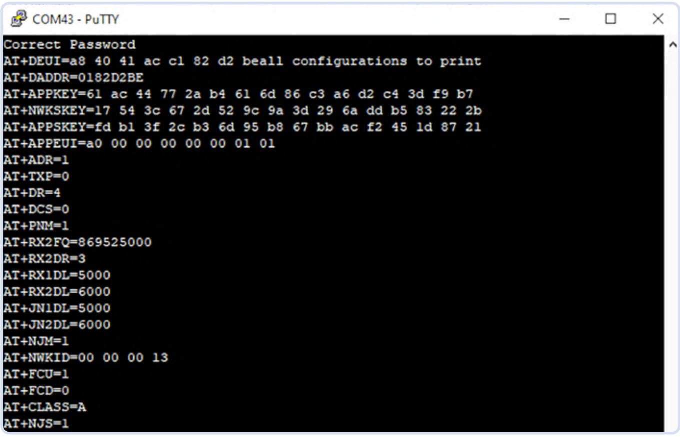


Figure 5: Query of the LHT65 configuration.

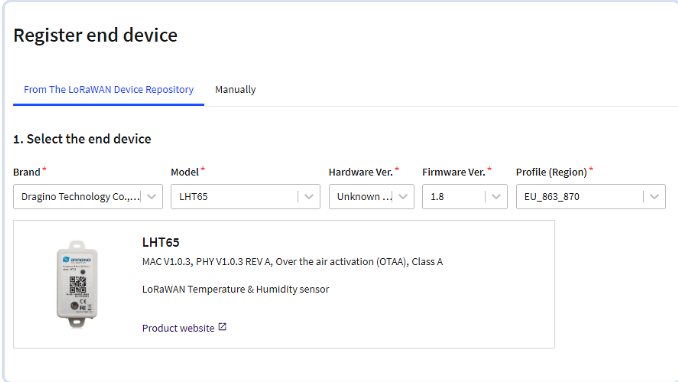


Figure 6: LHT65 registration part 1.

I use PuTTY as a terminal program and connect it to COM43 and 9600 baud. You can view the current COM port in the device manager when connecting the FTDI converter to a USB port.

PuTTY shows the output of the LHT65. To go into the configuration mode, enter the password *123456*. The following `ATZ` command resets the internal microcontroller. Each command must be terminated with ↵(CR/LF).

```
123456 ↵
ATZ ↵
```

You can query the current configuration using the `AT+CFG` command (**Figure 5**).

The upper lines of the output contain the parameters that are important for registration at TTS (CE). If there are no changes, these should match the data on the sticker supplied.

The registration at the TTS (CE) now takes place in a slightly different form since the Dragino LHT65 sensor already has a registration entry in the TTS (CE) and the corresponding data is stored (**Figure 6**).

Entering the EUIs and keys follows the usual method (**Figure 7**).

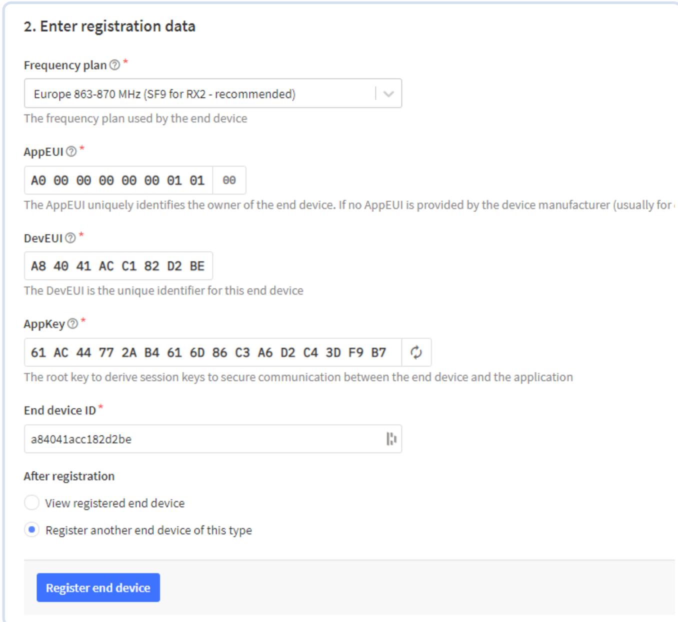


Figure 7: LHT65 registration part 2.

Clicking the *Register end device* button finishes the new device registration at TTS (CE). If there are no errors, the LHT65 sensor node reports its live data after a short time (**Figure 8**).

To receive readable messages in the TTS (CE) console, I entered the JavaScript given in **Listing 1** into the payload formatter in order to decode the payload. The payload formatter decodes data for battery voltage, temperature, and relative humidity measured by the SHT20, and the temperature measured by the external DS18B20.

Figure 9 shows a section of the TTS (CE) console window with the decoded messages of the LHT65. In the beginning, the default transmission interval is 20 minutes.

A downlink message can configure the LHT65. In the Dragino LHT65 manual referenced above, you will find a description of the configuration via AT commands and downlinks.

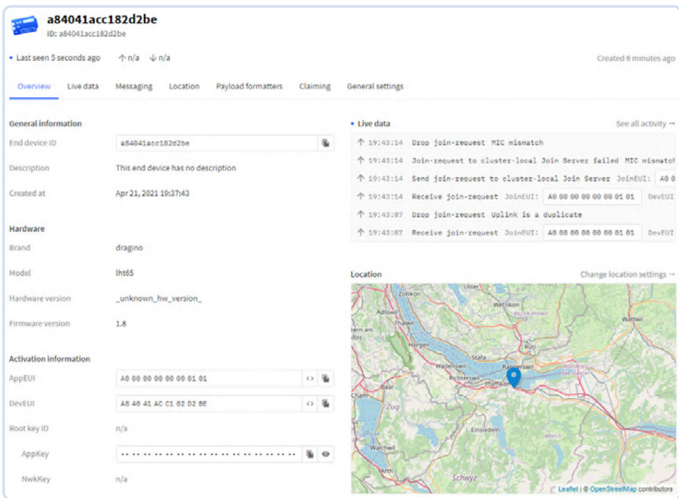


Figure 8: LHT65-supplied live data.

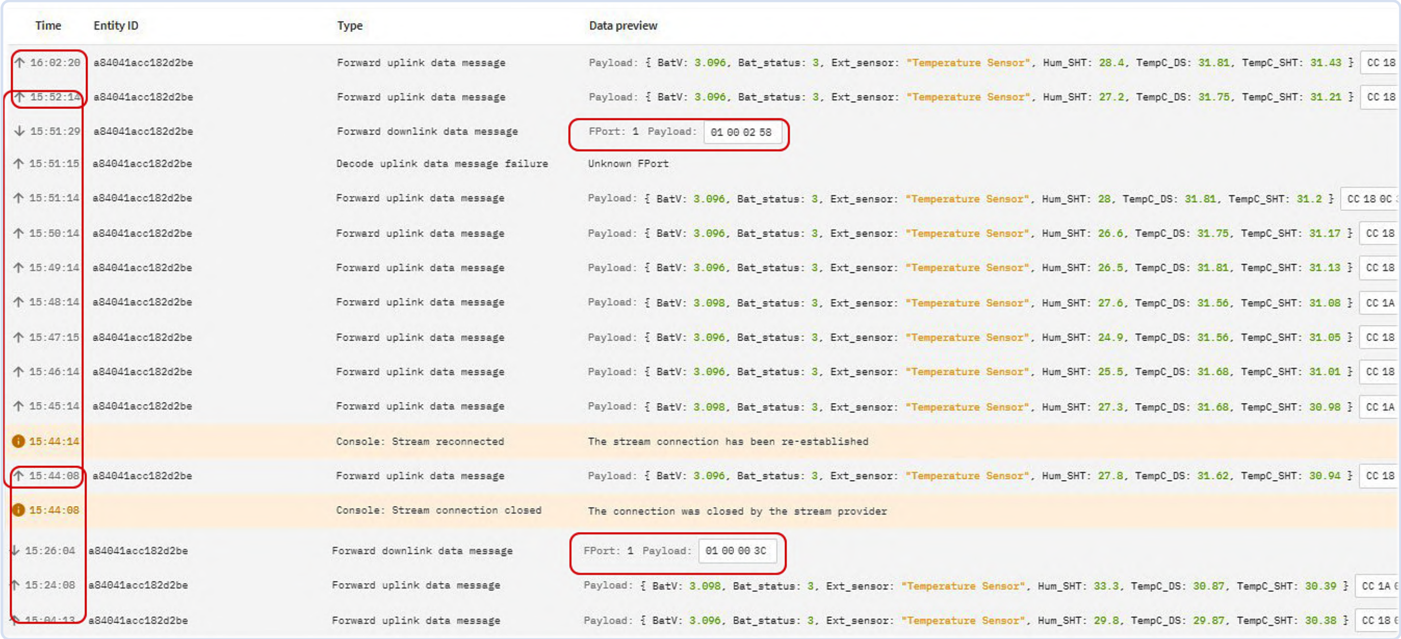


Figure 9: LHT65 — live data and downlink configuration.

The downlink command `01 xx xx xx` configures the interval time. As you can see in **Figure 10**, I have set the interval time to 60 (0x3C = 60_D) seconds with the help of the `01 00 00 3C` command, which is why the following outputs are also displayed appear in this grid. With the command `01 00 02 58` (0x0258 = 600_D) the interval time is finally set to 10 minutes.

As can be seen from Figure 9, the downlink does not take effect until the next uplink. With a LoRaWAN Class A device, a receive window opens after the upload has taken place, and therefore messages can only then be received. In **Figure 10**, '*schedule downlink*' has a literal meaning. Here you can still decide whether the downlink should overwrite an already planned one or whether it should append to one already planned. The configuration via a downlink is very convenient.

Dragino LDS01 Door & Windows Sensor

The Dragino LDS01 is a LoRaWAN door & window sensor. The sensor detects the status as open or closed and sends this information to the LoRaWAN server. Thanks to the compact dimensions of 64 x 30 x 14 mm, the sensor can be positioned almost anywhere.

Listing 1: Javascript for Vbatt, RH, and temperature sensors

```
function decodeUplink(input) {
    var data = {};
    //Battery,units:V
    data.vbat = ((input.bytes[0]<< 8 | input.
    bytes[1]) & 0x3FFF)/1000;
    //SHT20,temperature,units:
    data.temp = ((input.bytes[2]<< 24 >> 16 |
    input.bytes[3])/100);
    //SHT20,Humidity,units:%
    data.humi = ((input.bytes[4]<< 8 | input.
    bytes[5])/10);
    //DS18B20,temperature,units:
    data.extTemp = ((input.bytes[7]<<24>>16 |
    input.bytes[8])/100);
    return {
     data: data
    };
 }
```

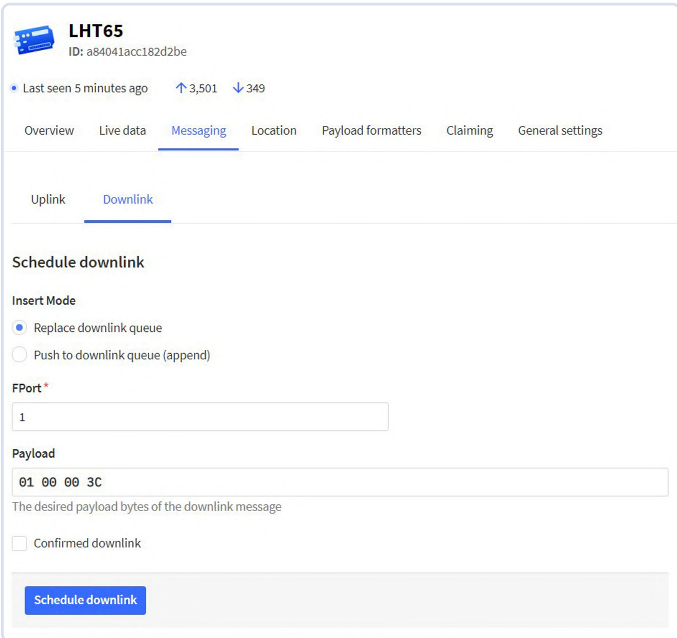


Figure 10: LHT65 configuration via downlink.

Figure 11: Dragino LDS01, closed.

Figure 12: Dragino LDS01, opened.

Figure 11 shows the closed sensor and **Figure 12** the opened one. A reed contact detects these two states. In addition, some data derived from these two states gets placed in the payload.

A type CR2032 battery powers the LDS01. With good network coverage (basis SF7, 14 dB), up to 12,000 uplink packets can be transmitted. A poor network coverage (based on SF10, 18.5 dB) reduces this to around 1,300 uplink packets. According to the manufacturers' design, battery life is up to one year. The user can easily replace the CR2032 battery.

The Dragino LDS01 package includes a sticker with the necessary data for registering the end device with the LoRaWAN server (**Figure 13**). I repeat: keep the sticker safe! It is wise to employ the EUIs and keys for the device registration and make subsequent adjustments later.

The LDS01 uses a serial interface accessible inside the device via a connector strip for configuration. You need an FTDI-USB-UART-converter to connect RX, TX, and GND again (**Figure 14**).

To access the LDS01, you should again use a terminal program (e.g., PuTTY). The baud rate is 115200 bps. The required password is 123456. The registration process for LDS01 is the same as for LHT65 (**Figure 15**). TTS (CE) already "knows" the LDS01.

Figure 13: Dragino LDS01 EUIs and software keys.

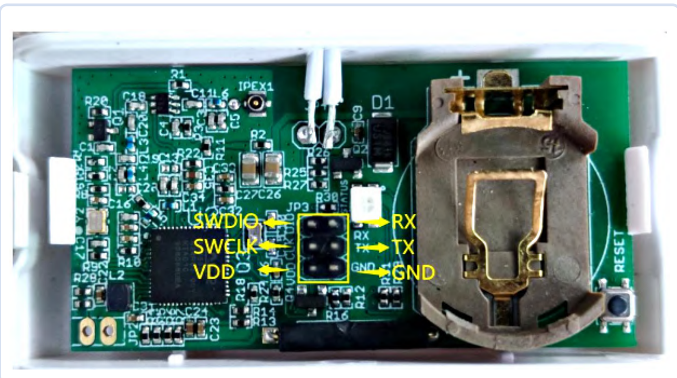


Figure 14: Dragino LDS01 with its case opened.

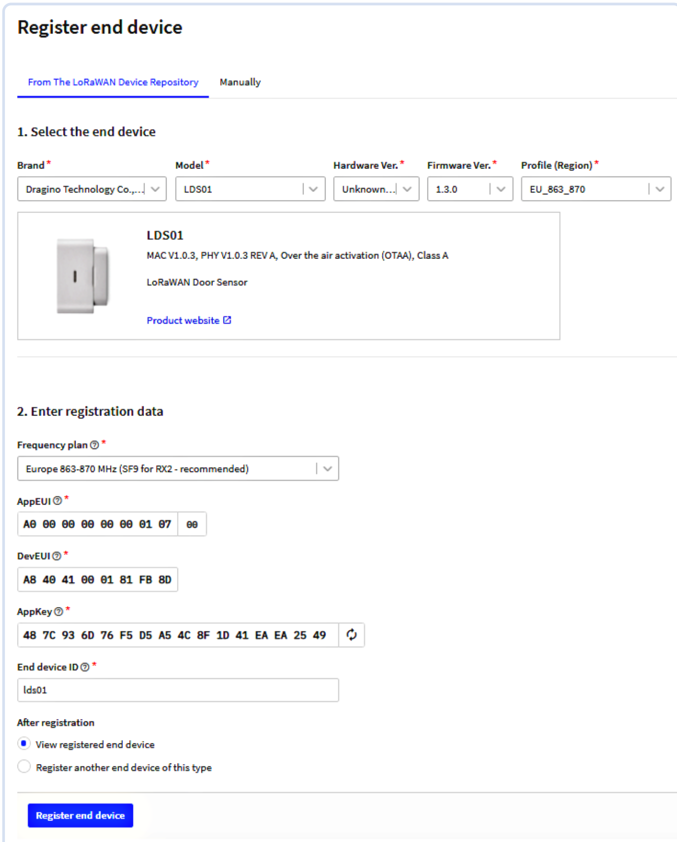


Figure 15: LDS01 Registration.

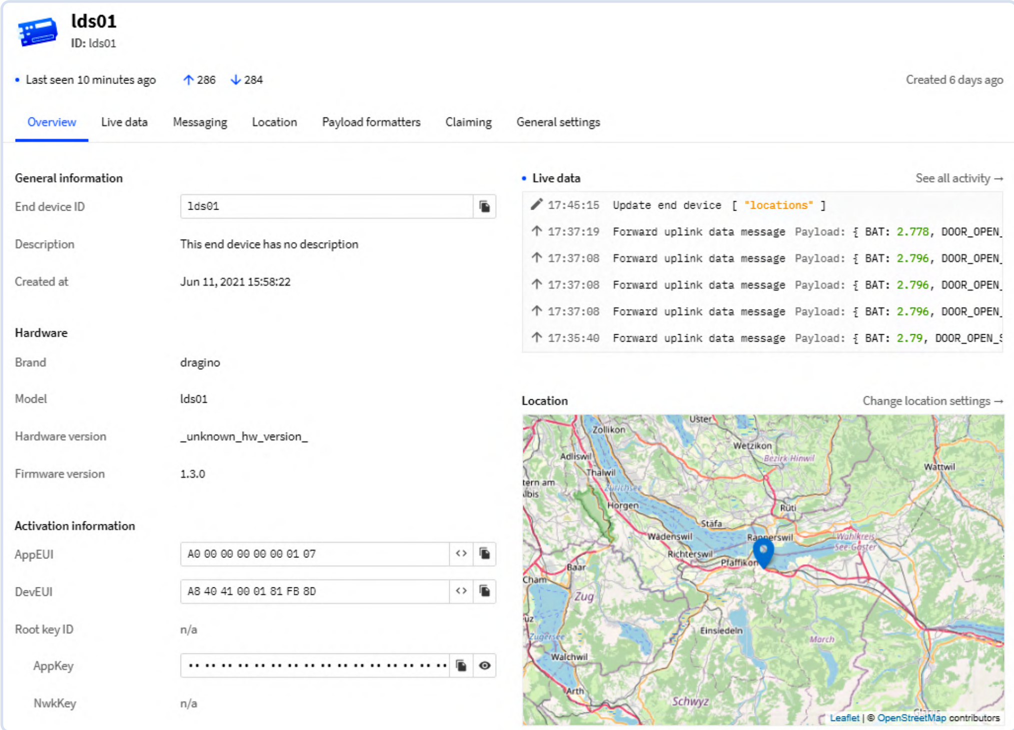


Figure 16: LDS01-supplied live data.

Clicking the *Register end device* button registers the new device at TTS (CE). If there are no errors, the LDS01 sensor node reports its live data after a short time (**Figure 16**).

To find readable data in the TTS (CE) console, I entered the JavaScript shown in **Listing 2** in the payload formatter in order to decode the payload. The payload formatter decodes data for the battery voltage, switch status, number of openings, duration of the last opening, and an alarm bit (not evaluated here).

In addition to using AT commands, a downlink message can configure the LDS01 too. The LDS01 manual describes the AT commands and downlink messages for the configuration of the Dragino LDS01 — you can find it at [2]. For example, the downlink command `A8 xx xx` configures the data rate. As you can see in **Figure 17**, I have set the data rate to *DR5* with the help of command `A8 00 05` to reduce the air time and thus also power consumption.

Listing 2: Javascript for LDS01

```
function decodeUplink(input) {
    var state = input.bytes[0] & 0x80 ? 1:0; //
1:open,0:close
    var voltage = ((input.bytes[0]<< 8 | input.
bytes[1]) & 0x3FFF)/1000;
    var alarm = input.bytes[9]& 0x01;
    var open_times = input.bytes[3]<<16 | input.
bytes[4]<<8 | input.bytes[5];
    var open_duration = input.bytes[6]<<16 |
input.bytes[7]<<8 | input.bytes[8]; //units:min

return {
    data: {
      BAT: voltage,
      DOOR_OPEN_STATUS: state,
      DOOR_OPEN_TIMES: open_times,
      LAST_DOOR_OPEN_DURATION: open_duration,
    },
    warnings: [],
    errors: []
   };
 }
```

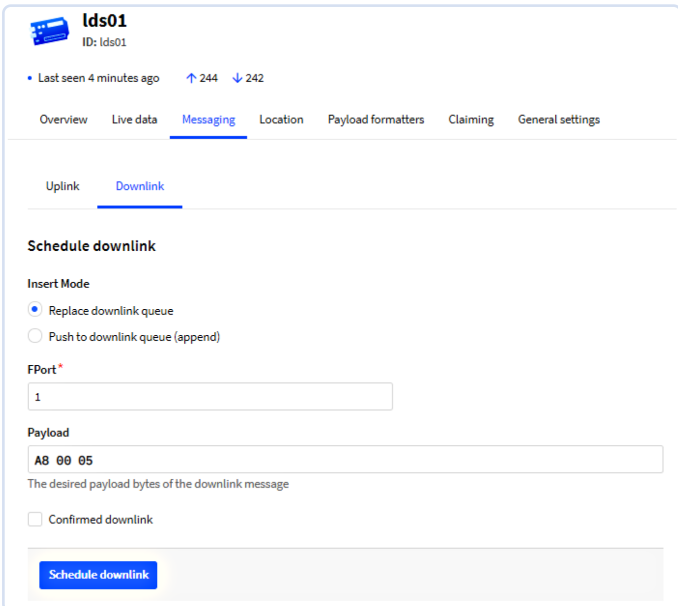


Figure 17: LDS01 configuration via downlink.

Figure 18: The Dragino LDS02.

If the programming of the LDS01 does not work via download, you can fix the firmware using AT commands. The `AT+CRX1DELAY=5` command adjusts the firmware to the conditions of TTS (CE).

The LoRaWAN library, on which the LDS01 and LWL01 sensor code is based, has a fatal error in a binary part of the library. A quick-and-dirty test with version 4.4 of the library shows that Dragino fixed the problem. There is no statement about other software versions. It looks like Dragino will have to release updated firmware for its ASR650x-based products to make them TTS (CE) compatible, see the problem report at [3].

Dragino LDS02 Door & Windows Sensor

The Dragino LDS02 is another LoRaWAN door & window sensor. Like the LDS01, the sensor detects the status as open or closed and sends this information to the LoRaWAN server.

Using two AAA batteries, the dimensions of 69 x 29 x 54 mm are a little less compact than the LDS01 (**Figure 18**). These two batteries are sufficient for around 16,000 to 70,000 uplink packets. When the batteries are empty, the user can replace them with two commercially available AAA batteries.

A set of unique keys for LoRaWAN registration is preloaded on each LDS02. Registration at TTS (CE) is the same as with the LDS01, and the connection is made automatically after switching on.

(220296-01)

About the Author

Dr. Claus Kühnel studied information technology at the Technical University of Dresden, Germany. He has developed embedded systems for laboratory diagnostic devices, among others. In this interdisciplinary field, he came in touch with the Maker scene. Passionate about new technologies around microcontrollers, he is the author of numerous articles and books on microcontroller hardware and software in Germany and abroad.

Questions or Comments?

Do you have any technical questions or comments related to this article? Email the author at info@ckskript.ch or Elektor at editor@elektor.com.

WEB LINKS

[1] Dragino LHT65 Manual: https://bit.ly/35cHPiP)
[2] Dragino LHT65 firmware update description: https://bit.ly/3MWw5oC
[3] Dragino LDS01 Manual: https://bit.ly/3a34ovG
[4] Dragino ARS650x firmware error: https://www.thethingsnetwork.org/forum/t/new-application-v3-otaanotworking/43338/30

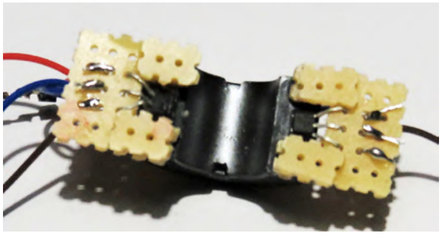
Err-lectronics

Corrections, Updates and Readers' Letters

Compiled by Jens Nickel (Elektor)

DC Current-Sense Clamp

Elektor 5-6/2021, p. 26 (2w00595)



On the second page of the article, there are two formulas: $V = I \times 38.92\ mV/A$ and $V = I \times 39.26\ mV/A$. The results are a factor of 10 too large for the assumed number of turns (n = 1). The correct formulas for n = 1 are $V = I \times 3.892\ mV/A$ and $V = I \times 3.926\ mV/A$.

DC Current-Sense Clamp

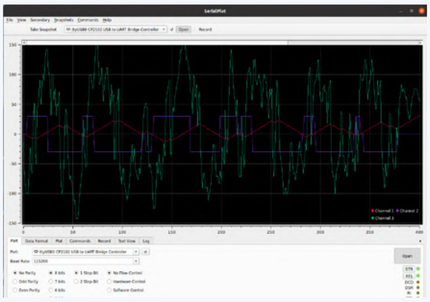
Elektor 5-6/2021, p. 26 (200595)

I found the article on the DC current-sense clamp with a Hall sensor very interesting. That gave me the idea of using a ferrite core with a Hall sensor as a pick-up for detecting voltage pulses in the spark plug cables of vintage motor bikes. At the events of our old-timer club, we want to make noise measurements on the exhaust systems. This requires a fixed engine speed of 2000 rpm. The motor bikes have magneto ignitions, so no low-voltage pulses (6 V or 12 V) are present, and adjusting settings on the engines is not possible. A "voltage-sense clamp" would be ideal for our purpose. The circuit described in the article is intended for measuring current, but the ignition pulses in a spark plug cable have a high voltage (10 to 20 kV) at low current. Can a ferrite core with an A1324LUA also be used for this purpose?
Ardy Notenboom

I think a ferrite ring core is not suitable for sensing the pulses in your ignition cables. The coupling between the high voltage pulses and the coil might distort the pulse waveform. However, it's worth a try. Capacitive coupling would probably be more suitable, such as the arrangement used to trigger stroboscopes (e.g., engine timing lamps). You could have a look at the circuit diagrams of typical stroboscopes.
Martin Ossmann (author of the article)

How to Use Arduino's Serial Plotter

Elektor 3-4/2022, p. 15 (200540)



In the March-April issue, you described the Arduino IDE serial plotter. However, there are better alternatives. One of them is SerialPlot, which I use in my projects. Versions for Windows and Linux can be downloaded at:

https://hackaday.io/project/5334-serialplot-realtime-plotting-software/log/192838-serialplot-v012-release

A sample plot is shown in the illustration.
Walter Trojan

Getting Started with the ESP32-C3 RISC-V MCU

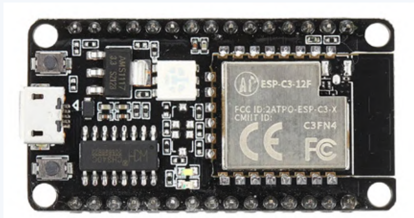
Elektor 1-2/2022, p. 59 (210466)

I have a quick question before placing an order: Does the ESP-C3-12F-Kit in the Elektor store (*www.elektor.com/esp-c3-12f-kit-development-board-with-built-in-4-mb-flash*) already use revision 3 of the ESP32-C3 (i.e., without the bugs described in the article)?
Michael Kasper

The kit in the store has revision 3 or later of the ESP-C3. The revision 2 bugs described in the article are therefore not present.
Mathias Claußen (Elektor)

Your First Steps with an ESP-32 C3 and the IoT

Elektor 5-6/2022, p. 6 (220017)



As an Elektor subscriber, I read the article and then had the idea of putting Node-RED on my Raspberry Pi 3B in order to run MQTT. As a newbie, that immediately led to the first problem: does my Raspberry Pi need its own display and keyboard for this, or can I control it from my PC? Is SSH sufficient for this, or do I need another connection in order to view the Node-RED diagrams?
Hans Kast

Node-RED is entirely controlled and programmed via a browser, which does not have to run on the Raspberry Pi. It is therefore sufficient to install Node-RED on a Raspberry Pi without a monitor or input devices connected; all you need is an SSH link for the installation. For information on installation and operation, you can consult the article *https://www.elektormagazine.com/articles/starting-with-nodered*.
Mathias Claußen (Elektor)

220305-01

5G Just for Me

Gaining Complete Control of 5G Deployments with Private Cellular Networks

By Stuart Cording (Elektor)

While the general public sees 5G as shorter loading times for video content, commercial users are keen to leverage its other advantages: low latency, high reliability, and massive connectivity. However, they are not always ready to place their sensitive data on public mobile networks. Luckily, there are alternatives. Radio spectrum is available to deploy private cellular networks, enabling factories, stadiums, exhibition centers, and airports to take full control, and advantage, of 5G on their sites.

As is often the case, the marketing machinery behind 5G has been marching well in front of the technology's deployment. Those living in the UK could almost believe that Kevin Bacon, actor and brand ambassador for the mobile network operator EE [1], personally delivers customers' new 5G smartphones. But, away from the domain of consumers, 5G is having its moment, flexing the incredible performance baked into its specifications.

While previous improvements to cellular technology focused on data throughput and improved user experience as we moved from 2G to 3G and then to 4G, 5G is slightly different. Yes, the throughput has been improved significantly, known as enhanced Mobile Broadband (eMBB), offering data rates 100 times faster than 4G, but there is more. You also have massive Machine Type Communications (mMTC), enabling up to around one million devices to be deployed in a square kilometer and 10 years of battery life for wireless modems targeting such applications. Then there is Ultra-Reliable and Low-Latency Communication (URLLC). This provides an air interface latency of down to 1 ms together with availability of 99.9999%.

The Role of 5G Capabilities in Applications

While the performance capabilities are impressive, it should be made clear that they are not all fully available simultaneously. Each targets the needs of different markets and industries. eMBB obviously benefits consumers' need for more bandwidth for video. But it is also of interest to the entertainment industry. Cameras can be integrated into the helmets of sportspeople, providing first-person footage from racing drivers or American football players. mMTC allows the tracking of shipping containers from one country to the next

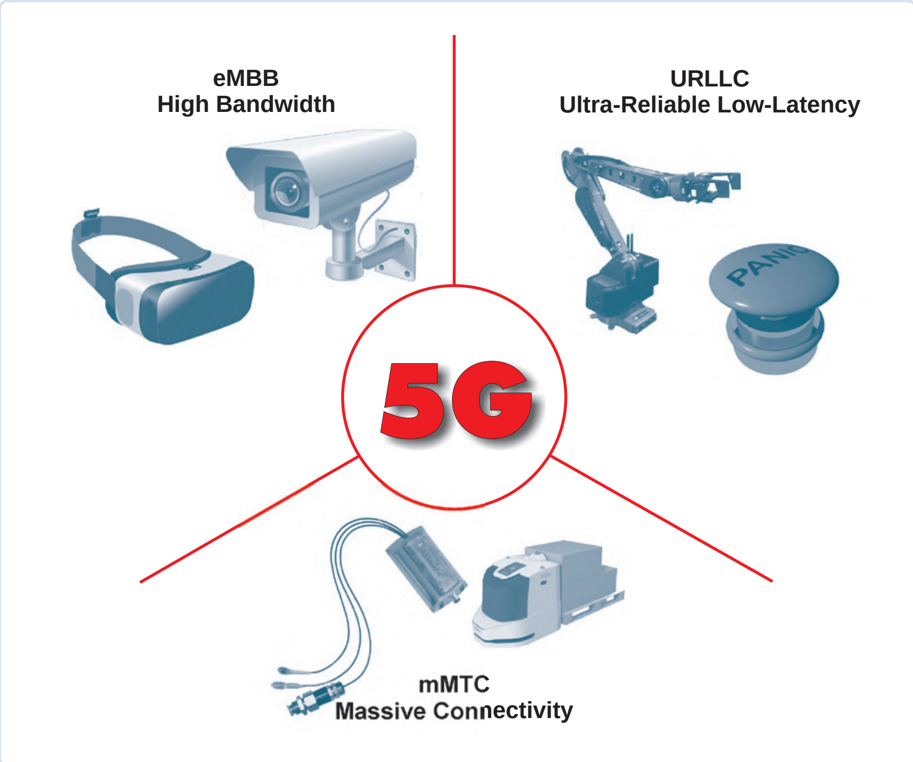


Figure 1: 5G is not just about faster mobile broadband. The specification targets a wide range of commercial, automotive, and industrial needs with low latency, high reliability, and ultra-high bandwidth.

or monitoring air quality with sensors deployed in cities. URLLC can be used in factories, allowing 5G to be part of the control loop for industrial robots and cobots (**Figure 1**). In automotive, both mMTC and URLLC will be used in vehicles. This allows them to share driving data with other vehicles and fixed infrastructure, from low bandwidth telemetry to time-critical data regarding emergency braking.

Some applications, such as automotive, will rely on 5G fulfilling its performance capabilities through the infrastructure installed by mobile network operators (MNO). However, others, such as the operators of factories, harbors, airports, and sports stadiums, have other options. For example, eMBB provides plenty of bandwidth for video but the networks surrounding a football stadium could get overloaded as fans simultaneously share clips of the latest goal. This could mean that television producers lose video quality when sharing the same 5G infrastructure. In a factory, thousands of Industrial Internet of Things (IIoT) sensors regularly sending data, even if they are small packets, could result in massive bills for data each month.

Installing Your Own 5G Infrastructure

To circumvent such challenges, one option is to install your own 5G infrastructure as a private cellular network (PCN). It should be noted that the purpose is not to generate revenue by offering 5G to anyone in the vicinity. The goal is to gain control over the deployment to fulfill the business's needs. This can range from ensuring availability or guaranteeing bandwidth to keeping operational costs under control and increasing security by keeping all data on-site. Furthermore, hardware can be deployed to ensure full coverage over an entire site, both indoors and out.

PCNs are not new. Some industrial sectors, such as mining, have employed 4G LTE PCNs for many years. According to data by Analysys Mason, there are over 1,000 private LTE/5G networks deployed worldwide, performing real-time monitoring in vehicle manufacturing plants, remote monitoring of cranes at ports, and integrated into safety applications at oil refineries [2].

One of the core challenges is acquiring radio spectrum for the deployment. Public spectrum has been auctioned off around the world to MNOs. If an agreement can be found, businesses wishing to implement a PCN could lease some of this spectrum if available. Then there is industrial spectrum that has been reserved in countries such as Japan and Germany. Regulators have set this aside expressly so that industrial and other users can gain the benefits of 5G if they are prepared to invest in the infrastructure needed.

In some countries, shared spectrum is available. Ofcom, responsible for regulating access to the airwaves in the United Kingdom, offers shared access licenses in four spectrum bands for low and medium power operation. Their goal is to enable innovation that improves mobile coverage, tests connected and autonomous transport solutions, and implements smart utility metering [3]. In the United States, band n48 (3550 to 3700 MHz with 5 to 100 MHz channels) has been made available similarly by the Federal Communications Commission (FCC). Known as Citizens Broadband Radio Service (CBRS), it is promoted by the OnGo Alliance [4].

Steps to Deploying a PCN

Whether operation is planned in a licensed or unlicensed band, there is plenty of work required before deploying hardware. According to guidance prepared by Rohde & Schwarz, this starts with rollout preparation [5]. Interference hunting must be undertaken in the relevant frequency bands using a range of equipment, such as spectrum analyzers and handheld and scanning receivers. This can involve simply walking the site or even using drones. Interferers must be dealt with by switching them off or reporting them to local regulators.

With all issues resolved, the PCN hardware can be deployed. However, before use, site acceptance testing must be completed. Signal decoding tests ensure that the 5G system operates correctly at a protocol level and that the signal to interference noise

Figure 2: Embedded SIMs, known as eSIMs, are soldered directly to the circuit board. Devices such as the Infineon SML 97 are 30 times smaller than an equivalent nano SIM card. (Source: Infineon)

ratio (SINR) is within the permitted limits. Download and upload speed and round-trip times enable the team to assess the user experience.

Further testing and site monitoring should also be planned to ensure ongoing performance and coverage, especially as the environment, such as a factory, is filled with equipment. One tool suited to this task is the QualiPoc Android [6], an app from Rohde & Schwarz. It offers a comprehensive range of tests from voice and data quality to RF optimization across all carrier frequencies when installed on a suitable handset. For IoT applications, it can also be combined with NB-IoT modules attached to the smartphone's USB interface to review coverage and functionality within buildings and basements.

Chucking Out the SIM Card

It is fully expected that many devices will need to operate on both PCN and public networks. Additionally, some networks may only be installed to support an event, such as a festival, trade fair, or sports tournament. It would be absurd if everyone using the PCN had to install an additional SIM card to use the network temporarily. This is where the embedded SIM (eSIM) comes in, doing away with fiddly SIM carriers and awkward card trays.

Rather than coming in card format, the eSIM is a component that is soldered directly to the circuit board of the electronic device. Using well-established security protocols, Remote SIM Provisioning (RSP) allows cellular devices to be given access to networks using over-the-air updates.

And while SIMs may look unimpressive from the outside, inside they feature advanced, high-performance 32-bit processors with a wide range of security features to prevent electrical and physical attacks. Accelerators are also included to implement encryption as efficiently as possible using modern algorithms such as DES/AES symmetric and RSA and ECC asymmetric cryptography and Hash engines for message authentication. The application running inside a SIM uses a cut-down implementation of Java known as JavaCard.

The SLM range of solutions from Infineon is one option, with the SLI 97 and SLM 97 [7] targeting industrial and automotive machine-to-machine (M2M) applications, respectively (**Figure 2**). These devices are also offered in M2M standard surface-mount MFF2 packages measuring 5 × 6 mm or chip-scale packages just 2.5 × 2.7 mm in size, a significant improvement over even the nano SIM. STMicroelectronics is another alternative with their ST4SIM-200 family [8]. As well as standard devices for smart things, they also have products suited to industrial-grade applications and AEC-Q100 Grade 2 options for automotive.

Alternatives to 5G PCN

A full PCN may be too much for some businesses to contemplate. In such situations, hybrid models are available. An enterprise may use private radio solutions connected to an MNOs public core. This could provide coverage inside buildings or the required data latency in a factory without having to implement the entire network (**Figure 3**).

One of the challenges of 4G is the lack of ability to reserve any part of the wireless link to provide guarantees of latency, reliability, or bandwidth. Everyone with a connection was treated equally. With the concept of network slicing in 5G, this all changes (**Figure 4**). Network operators can virtualize their traffic, creating connections that offer different performance profiles.

One slice could be dedicated to low-latency communications as used for real-time control of robots. In contrast, another

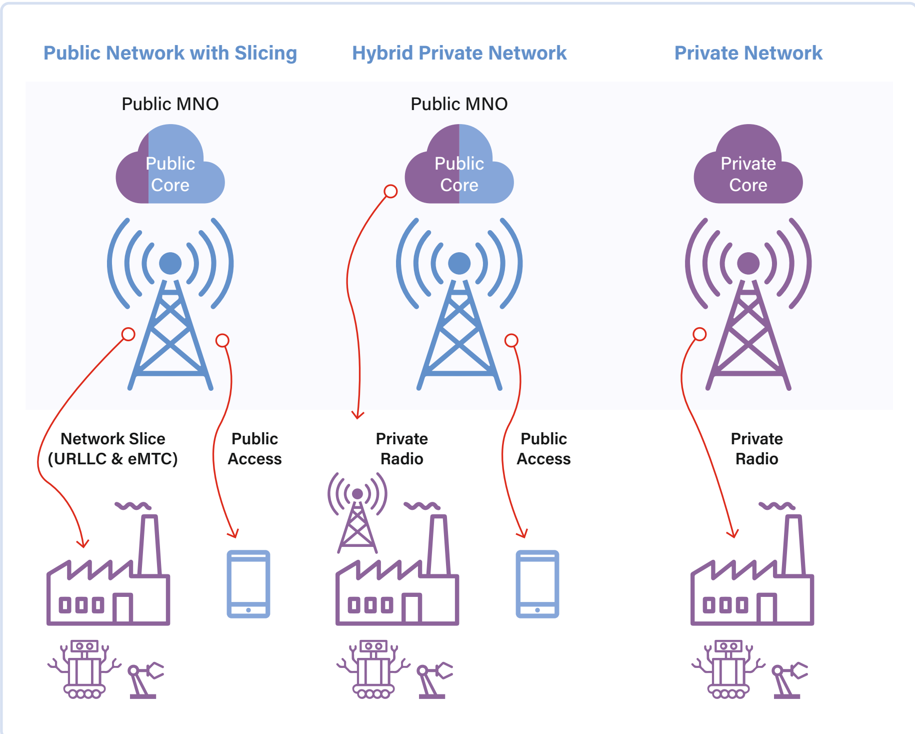


Figure 3: If the investment required for a full private cellular network is too prohibitive, a hybrid approach can be taken by attaching private radio equipment to a public mobile network operator.

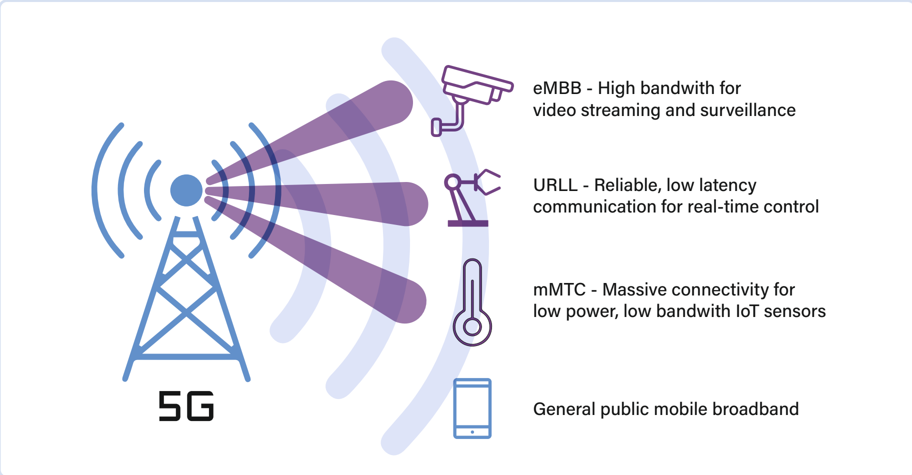


Figure 4: With the introduction of network slicing, mobile network operators can deliver the specific 5G capabilities individual businesses need, while still serving public users with excellent mobile broadband.

slice may provide high bandwidth for video cameras delivering data to an artificial intelligence (AI) vision system. MNOs can thus partition their networks to meet the quality-of-service needs of their customers. If desired, PCNs can also implement network slicing to ensure specific equipment operates over 5G as needed. However, like any resource, the number of slices available is limited.

Of course, the other technology looking to steal the limelight is Wi-Fi 6 (802.11ax). Compared to Wi-Fi 5 [9] (802.11ac), it only offers 37% higher throughput. However, thanks to beamforming, Wi-Fi 6 ensures that all users attain improved data rates as signals are steered towards each user rather than broadcasting in all directions. This should see the throughput per device improve significantly compared to previous generations of the standard. It is also one of the approaches used by 5G to improve per-user data throughput.

There is also Wi-Fi 6E that operates in the unlicensed 6 GHz band. In the United States, this provides 1200 MHz of bandwidth. In other regions, the situation is different. China has reserved the 6 GHz band for 5G, while Europe has reserved just 500 MHz of the band for Wi-Fi, leaving the upper part for 5G [10]. Wi-Fi will continue to play an important role in IoT implementations. However, Wi-Fi wasn't designed for objects on the move that need a smooth handover between access points, such as autonomous guided vehicles (AGV). Additionally, its security and network reliability doesn't match that of 5G. So, while Wi-Fi 6 may offer advantages in the short term before 5G is fully established, 5G is likely to be the preferred choice, especially in commercial and industrial applications.

5G Is on Its Way

While it is easy to laugh at some of the advertising attempting to convert consumers to 5G users, it remains an important reminder of the efforts MNOs are putting into this new technology. Consumers and business users will benefit from the higher data rates on offer. MNOs also benefit from more energy-efficient, configurable networks and the ability to offer differentiated services to customers. But by far the most exciting aspect are PCNs, enabling large commercial users to truly leverage the low-latency, massive connectivity, and high throughput of 5G. And although these networks are private, this doesn't stop traditional MNOs and new players, such as Amazon Web Services [11] (AWS), from getting in on the action in this space too by providing the hardware, software, and backend technology to make them happen.

220297-01

Questions or Comments?

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

WEB LINKS

[1] "EE | 5G. So real you could swear you were there," EE, June 2019: https://bit.ly/3MGa4dH
[2] M. Mackenzie, C. Chappell, "What are private LTE/5G networks and why are they important?," Analysys Mason, February 2021: https://bit.ly/3sY1T4B
[3] "How Ofcom has enabled innovation through spectrum sharing," Ofcom, August 2020: https://bit.ly/3MOjQuq
[4] OnGo Alliance Website: https://bit.ly/3MLcbNu
[5] "Network testing in private networks," Rohde & Schwarz: https://bit.ly/3GfDz3u
[6] "QualiPoc Android," Rohde & Schwarz: https://bit.ly/3wKUFTT
[7] "SLM security controllers optimized for industrial applications," Infineon: https://bit.ly/3wQK8Fl
[8] "ST4SIM: SIM & eSIM cellular connectivity portfolio," STMicroelectronics: https://bit.ly/3wKK8Ik
[9] E. Khorov et al., "A Tutorial on IEEE 802.11ax High Efficiency WLANs," IEEE, 2019: https://bit.ly/3MQpE6A
[10] "The Importance of 6 GHz for 5G's Future," GSMA, May 2021: https://bit.ly/3sZFLqv
[11] AWS Private 5G Website: https://go.aws/3PGu7KZ

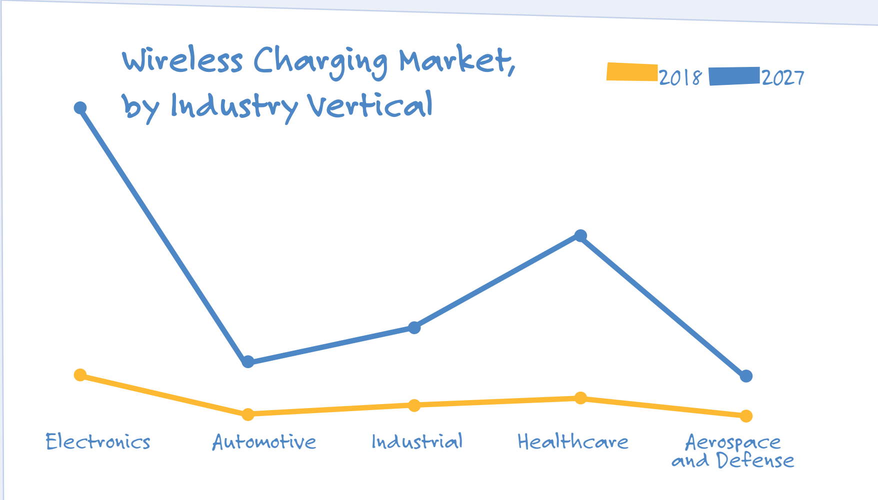
Elektor infographic

By Robert van der Zwan

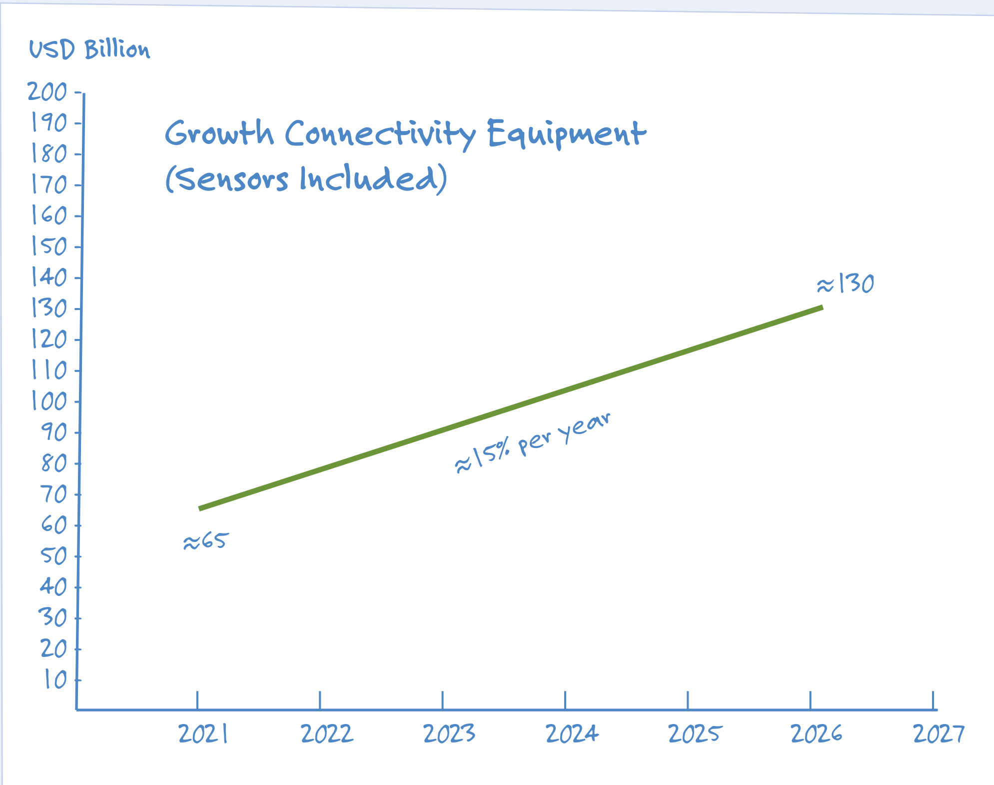
Wireless Charging: Becoming Part of the Family

Wireless communication and wireless charging are hot topics for two reasons. One has to do with market size, the other with market growth. Wireless charging is especially impressive when it comes to growth. 'Sizewise' we are not talking hundreds of billion dollars here, as is the case with wireless communication, but tens of billions. However, let's have a look at the yearly growth rate for wireless charging in a broad sense (= automotive included). This rate is 24.6% between 2021 and 2026, going from USD $4.5 billion in 2021 to $13.4 in 2026. In terms of growth, wireless communication falls behind, although also offering double digit figures.

Sources: (Allied Market Research; MarketsandMarkets)



Drone View Confirms: Everything Points in the Same Direction

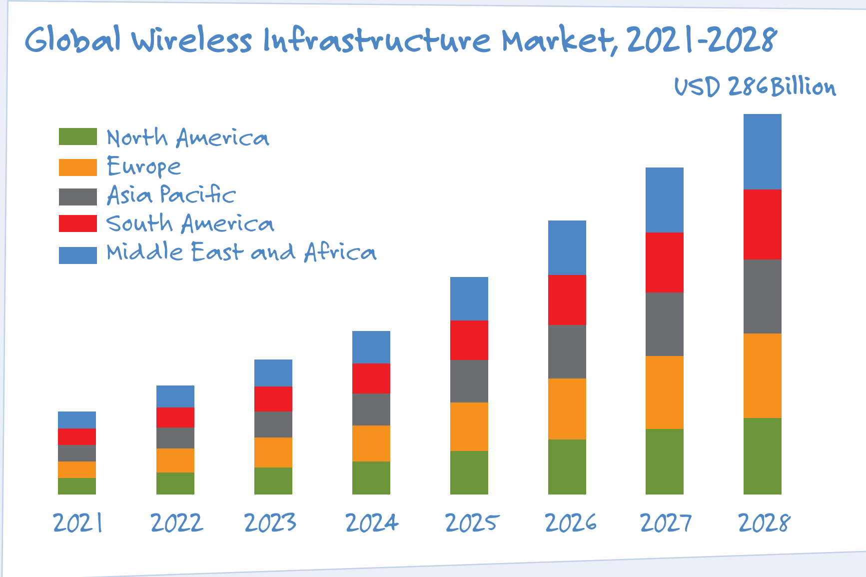


When taking a drone view of all necessary connectivity equipment within mobile devices (including sensors), it becomes clear that every part of society wants to embrace wireless technology. The global market entails so much more than consumers wanting to stay in constant contact while on the go. Municipalities want to build smart cities and companies a reliable 24/7-IoT. Overall, growth in connectivity equipment will indeed be in double digits. Marketing research companies like BCC Research and MarketsandMarkets project a growth of 15% for the coming five years. And yes, that includes a total new market: drones.

(Sources: BCC Research; Global Market Insights; MarketsandMarkets)

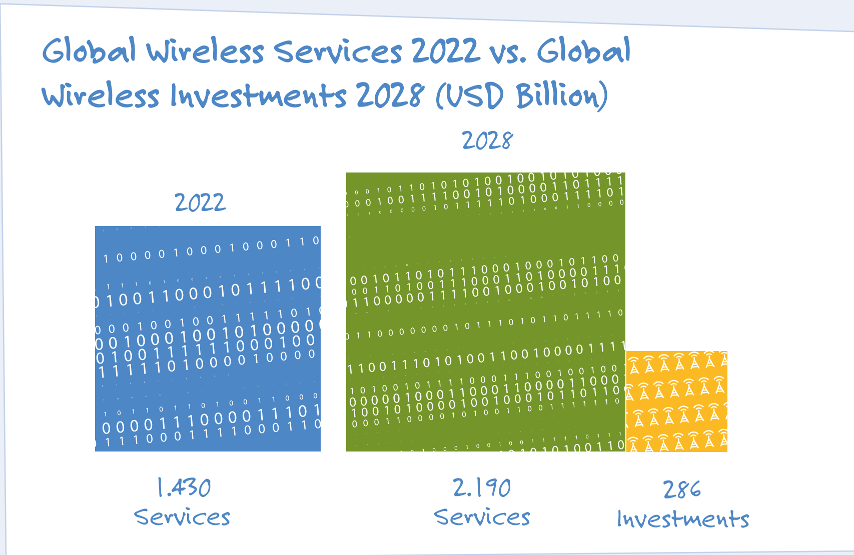
Infrastructure: A Lot Is Already There, Nevertheless...

A pictogram for wireless communication often includes a couple of radio waves representing an antenna of some kind. Antennas are not only the backbones for growth, they also represent growth themselves. For example, more antenna masts are necessary to accomodate 5G technology, which, after all, uses higher frequencies to reach much higher speeds than 4G. That is why the market for wireless infrastructure may grow to USD $286 billion in 2028, according to Data Bridge Market Research. Between 2021 and 2028 this boils down to a yearly growth of 7%. We are not only talking antenna masts here, but also base stations and central hubs.



(Sources: Data Bridge Market Research; Global Market Insights)

Investments? The Market Size Will Take Care of Them



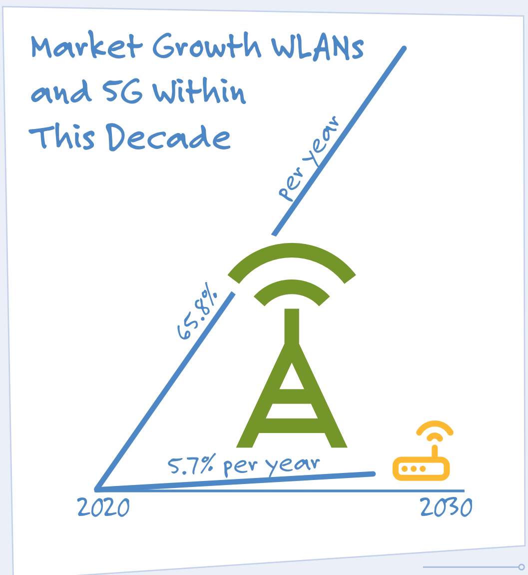
Huge investments in wireless infrastructure pose a huge risk for telecom service providers. Will they be able to get a decent return on their investments? The answer is an indisputable yes. This has not so much to do with the growth rate of wireless services. The growth rate of wireless services will roughly be the same as the investment rate in wireless infrastructure, around +7% between 2020 and 2030. It is the market size that matters. Whereas infrastructure investments will reach almost USD $300 billion in 2028, the global services market surpasses $1.4 trillion in... 2022.

(Sources: Data Bridge Market Research; Future Market Insights; Market Research Future)

In the Long Run It Is the Long Range

It is always tricky to differentiate between long range wireless communication and short range wireless connections. Where does short range stop and long range begin? For matter of clarity, let us suppose that WLANs stand for short range networks based on Wi-Fi 802.11. Furthermore, let us assume that long range networks are driven by the 5G standard. Thus, an interesting picture enfolds when all hardware, software and services are combined (telecom infrastructure excluded). Whereas WLANs will grow about 5.7% between 2021-2028, 5G networks will grow a staggering 65.8% between 2020-2030. Sidenote: 5G can be used for WLANs as well.

(Sources: Allied Market Research; Digi; Fortune Business Insights)



How Does My Device Learn to Transmit?

Applications with Wi-Fi Interfaces

By Dr. Heinz Zenkner
(Freelance Consultant at Würth Elektronik)

As consumer and industrial devices become more connected, the need for easy wireless connectivity is increasing. Typically, this is achieved by implementing a Wi-Fi interface. But how does one go about doing so, and what are the potential pitfalls? This article explains how to create an RF-compliant circuit design, PCB design and layout, how to adapt the antenna, and which components are most appropriate.

Due to the huge rise of Internet of Things (IoT) applications and the decentralization of controls, many developers face the challenge of integrating wireless connections for data communication between the peripheral device such as a sensor, and the central controller. The advantages are obvious: no need to worry about galvanic isolation or laying fixed cables. A highly reliable wireless connection is needed to ensure secure communication even in high-interference environments. A high-reliability Wi-Fi connection needs an RF-compliant design that meets both EMC requirements and signal integrity. The design of the antenna interface, discussed below, includes circuitry, components, layout, and system integration.

Prevent Reflections at the Antenna

Wi-Fi controllers operate digitally and not only generate the required signals at their transmission output (Tx port), but also harmonic interfering signals. Furthermore, interference is caused by mismatches in the transmission path, which should ideally have an impedance of 50 Ω from the transmission output stage up to and including the antenna. To reduce harmonic interference and improve matching to 50 Ω, filters and matching networks are frequently used in the transmit and receive paths of the Wi-Fi interface. The transmission systems matched in the relevant frequency range ensure best possible signal transmission across the complete bandwidth.

Select Suitable Filter Components for RF

In a circuit, components such as capacitors, inductors, and resistors introduce a combination of different impedances. As a result, focusing on the most important component properties is key when using components in filters within the frequency range above 500 MHz.

HF Capacitors Above 500 MHz

Only certain types of capacitors are suitable for high-frequency applications such as the capacitors from the WCAP-CSRF series (No. 885392005010). Their electrical parameters shown in the datasheet are in **Figure 1** with the corresponding specifications for resonance

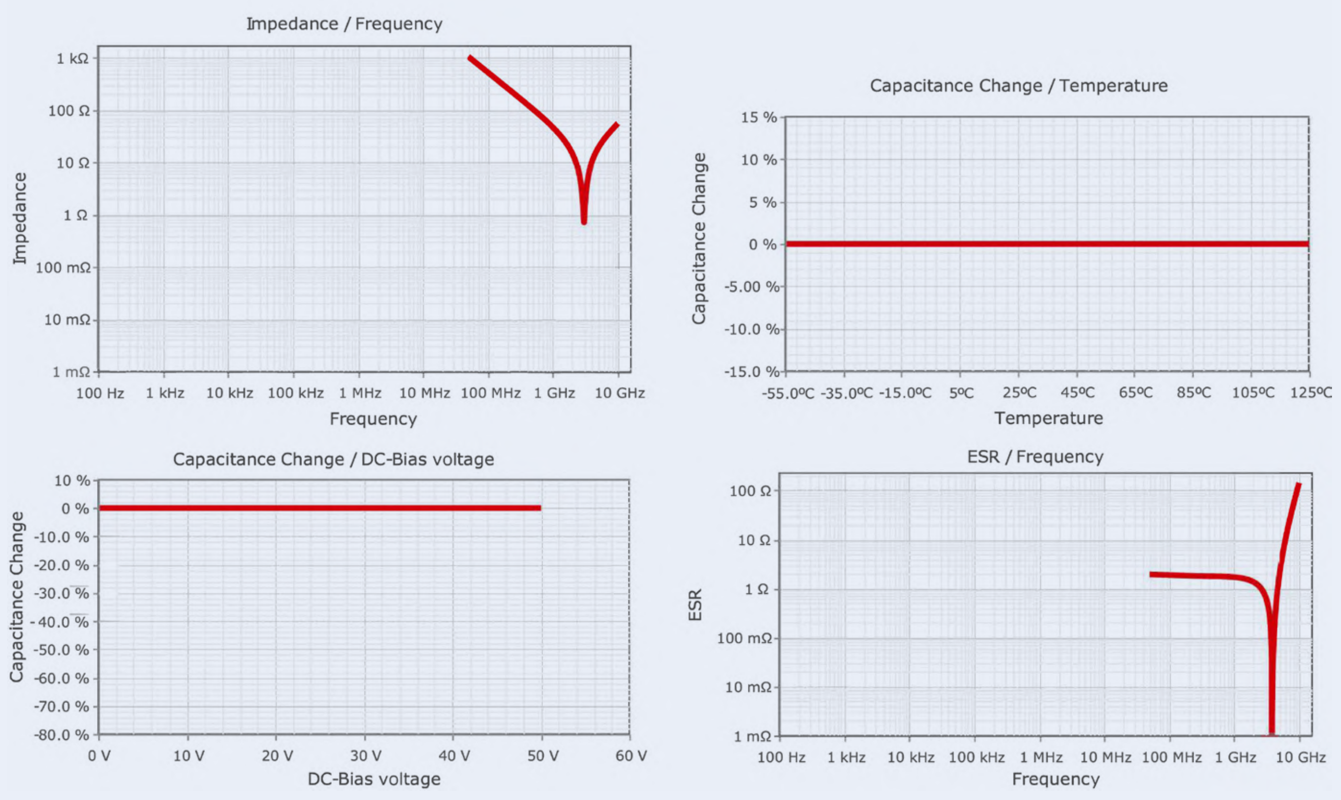


Figure 1: Electrical characteristics of the 3-pF RF capacitor WCAP-CSRF 885392005010.

frequency, ESR, DC bias drift, and temperature drift. These can also be simulated in the online simulation platform "REDEXPERT".

The capacitor's resonance frequency is around 3 GHz, the equivalent series resistance ESR is low up to the resonant frequency and both the DC bias drift and the temperature drift are negligible. In addition to a high Q-factor of over 460, a resonant frequency of 3 GHz results in a parasitic inductance of 0.3 nH, which is tolerable for most applications. Such values are critical for Wi-Fi, Bluetooth, and other applications operating in the GHz range.

RF Inductors from 500 MHz

Inductors without ferrite cores are common for antenna matching, as are inductors with ferrite cores for EMC filters. Where inductors have a ferrite core, the ferrite material must be carefully selected based on the impedance curves.

For the application of the filter as a low-loss antenna-matching network, or a similar RF application, inductors from the WE-KI series are most suitable. To achieve high quality and high resonance frequencies, the wire windings of these components are coiled on ceramic (**Table 1**).

Creating a Wi-Fi Interface

Figure 2 depicts an IoT interface board with a Wi-Fi interface as an example. The achievable radio range possible by a limited radiation source such as the transceiver depends heavily on antenna design, package, and PCB layout. It is not uncommon for RF performance data of circuits that use the same chipset and power, but a different layout and antenna design, to vary widely. In practice, most products feature a transceiver chip that combines transmit and receive functions. Of course, this implies that the antenna matching, feed lines and antenna itself are used for both transmit and receive operations. The receive channel typically has a wide dynamic range, with a sensitivity of > 95 dB and a 3 to 4 dB lower sensitivity at the antenna. Gain adjustments can compensate losses due to mismatch. The transmitting operation, however, is critical, because a 3 dB lower sensitivity of the antenna, or 3 dB higher losses on the path between the transmitting output stage and the antenna, requires twice as much power from the transmitter. If the Tx-chip is capable of generating the transmitting power, this inevitably leads to high current consumption and higher proportions of harmonic interference in the transmitting signal and possibly even EMC issues. However, the maximum permissible transmission level according to the relevant EMC directives must be observed. Since law regulates the maximum transmission level, it is even more critical to develop a well-tuned system on the receiver side that operates

Electrical Properties:

Properties		Test conditions	Value	Unit	Tol.
Inductance	L	250 MHz	10	nH	±5%
Q-Factor	Q	250 MHz	30		min.
Q-Factor	Q	900 MHz	66		typ.
DC Resistance	R_{DC}	@ 20 °C	0.13	Ω	max.
Rated Current	I_R	ΔT = 15 K	700	mA	max.
Self Resonant Frequency	f_{res}		4800	MHz	min.

Table 1: Characteristics of the SMD inductor 744761110A (data from the datasheet).

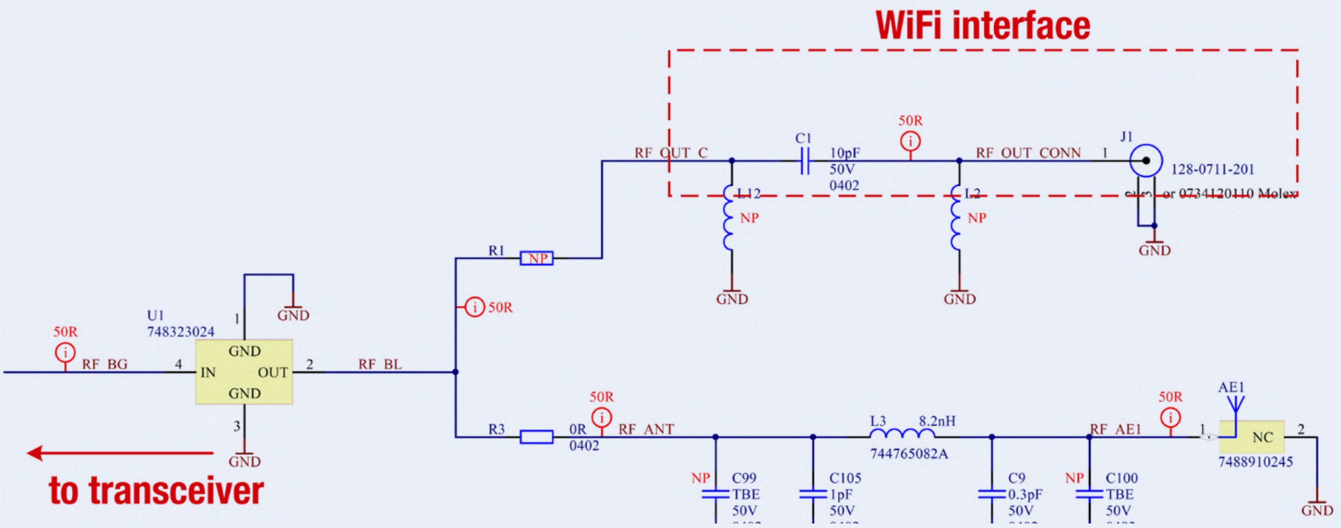


Figure 2: Wi-Fi interface circuit diagram of an IoT interface board.

with minimum received signal strength. It is worth noting that the manufacturer of the transceiver chip or communication module usually stipulates the most appropriate antennas in the specification. This ensures compliance with the applicable standards requirements of the RED directive 2014/53/EU, taking into account the application. If other antennas are used or if the recommended design is not followed, other RF properties will result!

Ceramic Chip Antennas

Ceramic chip antennas offer several advantages. They are small and therefore less susceptible to electromagnetic interference from nearby components. Changes to the board design or layout can easily be made without simulation. The antenna can simply be modified or even replaced with a different one. Mobile and high frequency applications such as GPS or 2.4 GHz radios frequently use chip antennas. **Figure 3** shows various WE-MCA multilayer chip antennas suited for Wi-Fi connections.

Matching Impedance of the Transmission Path

When the source's impedance equals the impedance of the load, maximum power is transmitted. This means that the source is the transmitter (output of the transmitter) with its impedance Z_T, which feeds a circuit path with an impedance $Z_L = Z_T$. The power transmits to the antenna with impedance Z_A, which should ideally be the same as the trace and the source. When all impedances have equal values, the maximum power results. If the match is not perfect, the transceiver sends a signal with amplitude V_{IN} into the circuit path, and only a portion of the signal reaches the antenna. The remaining signal reflects back at the transitions between source and line on the one hand and between line and antenna on the other.

Such mismatch can be considerably reduced by inserting a matching network, usually a π-, T-, LL- or LC-network. The network's capacitance and inductance values are in the pF and nH ranges. It is useful to have pattern sets with values ranging from 0.5 pF to 20 pF and 0.5 nH to 20 nH available for matching. As a result, antenna matching placement locations must be included in the layout of the Wi-Fi interface. **Figure 4** depicts the circuit diagram and the corresponding section of the board.

The layout and positioning of the L/C components and the chip antenna on the PCB are critical: the components for matching must be as close to the antenna or the antenna connection as possible. The antenna position, the size of the recessed area around the antenna, and the distance between the antenna and the reference ground plane affect the antenna's resonance frequency and impedance.

In this example, the antenna is positioned in the corner of the board. This way, the antenna is not surrounded by additional components, resulting in good antenna performance (**Figure 5**). The antenna's circuit path must be considered part of the antenna system. The edge area around the antenna cut-out, the ground planes, are plated through in all four layers. This ensures a stable reference ground for the antenna. The length of the circuit path feeding the antenna and the length and width of the ground plane influence whether the system works like a dipole or a monopole. If the ground plane is about 3 to 4 cm long and about 1 to 2 cm wide, the system will work as a dipole; if the ground plane is larger, the system will act as a monopole antenna.

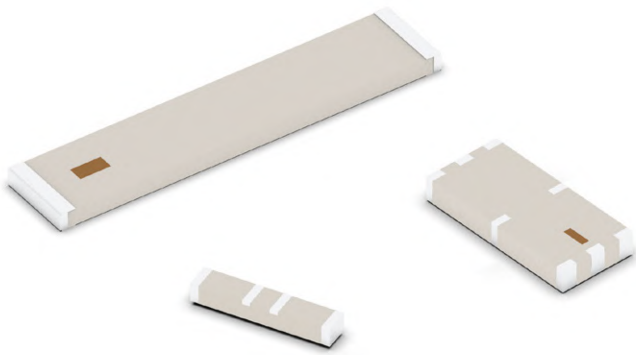


Figure 3: WE-MCA multilayer chip antennas in various designs.

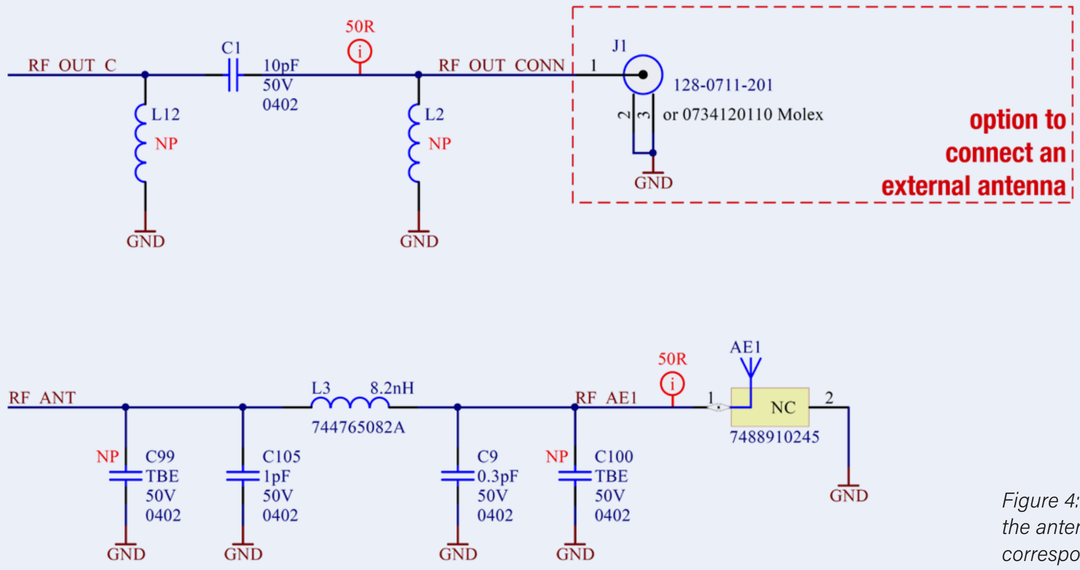


Figure 4: Matching networks for the antennas, circuit diagram, and corresponding layout section. C99 and C100 are not mounted.

Matching a Ceramic Chip Antenna

The example case study looks at matching the WE-MCA multilayer chip antenna (No. 7488910245) for maximum power transmission. A network analyzer measures the matching. The measured signal is the reflection attenuation S_{11}. Because of the differing impedances between the transceiver and the antenna, part of the signal coming from the transmitter reflects in the signal path. Since the impedance of the signal path is generally frequency-dependent, the reflection is also frequency-dependent. The smaller the impedance differences in the transmission frequency range, the lower the reflections. With the values shown in the circuit depicted in Figure 4, a reflection attenuation of 29 dB can be achieved, resulting in a VSWR (voltage standing wave ratio) of 1.06. **Figure 6** shows the reflection attenuation versus frequency.

For those who want help with antenna matching, Würth Elektronik offers an antenna matching service in addition to the components mentioned above. ◀

220369-01

About the Author

Dr.-Ing. Heinz Zenkner works on a freelance-basis at Würth Elektronik, specializing in Technical Marketing and Application Engineering. As an EMC subject-matter lecturer at the Technical Academy and a publicly appointed and sworn expert for EMC, he is a regular and authoritative contributor to various technical journals and books. In addition, he has worked as a lecturer at various universities and the Chamber of Industry and Commerce, and has participated in and led numerous seminars.

WEB LINK

Wireless Product Guide: we-online.de/wcs-product-guide

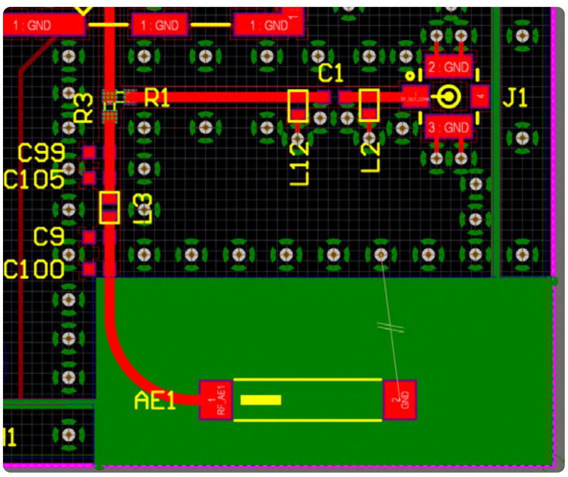


Figure 5: Layout section of the Wi-Fi interface board in the area of the Wi-Fi antenna.

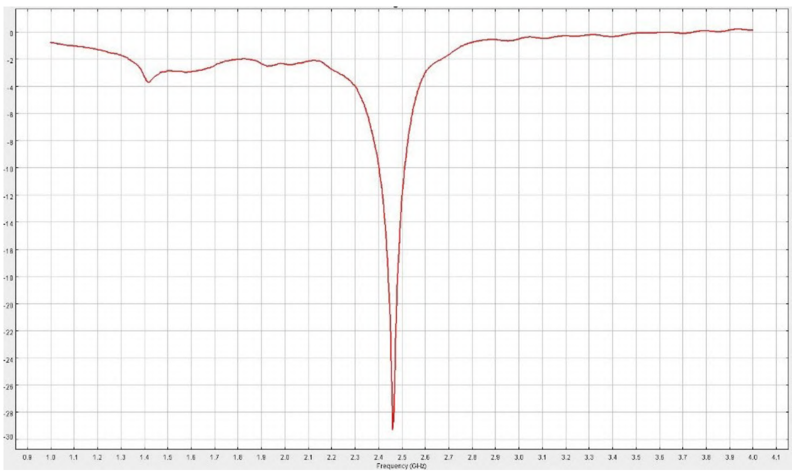


Figure 6: Reflection attenuation of the WiFi antenna port with matching network and hybrid antenna above the frequency.

Contributed by IoTize

Smartphones are the Heart of the IoT

The proliferation of screens, buttons and remote controls is over! In the age of the Internet of Things (IoT), the "physical" human-machine interface (HMI) is being digitized and replaced by a simple app on your smartphone. Apps make the monitoring and control of appliances much more accessible to today's users. Plus, once data is accessed from a smartphone app, sending it to a cloud platform (if necessary) is just one step more. In most cases, the app also reduces HMI costs while at the same time providing a more user-friendly and environmentally friendly interface.

Take the example of electric radiators (or any other heating system): a European directive now requires manufacturers to make heating appliances programmable. As a result, manufacturers first added screens to electric radiators to allow users to choose the temperature for each hour of the week. However, the tediousness of this task discourages most users so that less than 20% of Europeans were programming their electric radiators. This is in spite of the fact that programming could reduce household energy consumption by up to 15%!

Not only is the initial objective of that European directive (controlling energy consumption) not achieved, but the cost and carbon footprint of manufacturing appliances with a screen is unnecessarily increased. By contrast, a smartphone app that configures radiators offers only advantages. More efficient and user-friendly, the smartphone also reduces the cost of manufacturing these radiators and meets the objective of the European directive (**Figure 1**).

'No code' solutions for mobile HMI

Creating an HMI as a smartphone app is still rare because deporting a user interface to a mobile app involves multiple technologies: radio signals, communication protocols, cybersecurity, certifications, and mastery of Android and Apple environments, to name a few. The many facets of these technologies make development and projects complex.

By offering a ready-to-use solution that requires no coding to implement, wireless device suppliers can reduce development costs, shorten time-to-market, and eliminate technical risks. The French startup, IoTize, took a 'no code' approach for its TapNLink wireless modules, which offer a wide range of technologies – NFC, BLE, Wi-Fi, LoRa, and LTE-M.

The first characteristic of this no-code solution is its simplicity of implementation. The wireless device's firmware (Duetware) supports a 'no code' approach by pre-implementing all the features that are required in a connected device: communication protocols, encryption, access control, data handling, etc (**Figure 2**). This means adding the wireless device to your existing design is a simple process of connection and configuration. This is complemented by software tools that even generate any C code needed for integration with your existing application firmware, and an app generator for creating smartphone apps with graphical controls and data displays. In most cases, it takes less than an hour to connect and configure the wireless module and obtain a customized app.

The configuration of the module and of the graphical features of the HMI is enough to generate apps that communicate with the target equipment or appliance. The same app allows users to connect directly when their smartphone is close to the equipment, or remotely via the internet. The app can be provided in several versions according to the types of users and the associated rights (for example, apps that distinguish a user and a maintenance technician).

NFC: a versatile facilitator

The second characteristic of this no-code solution is the systematic use of NFC (Near

Figure 1: Smartphone avoids the need for extra LCD screens.

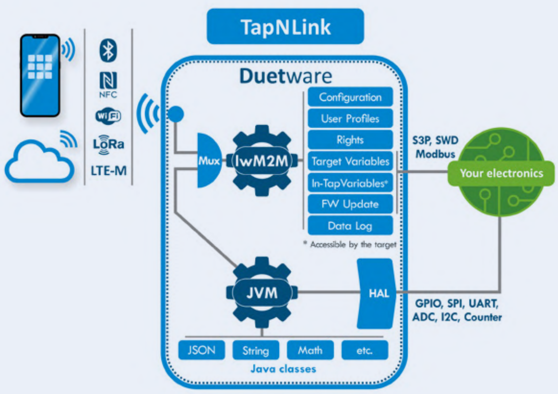


Figure 2: The wireless interface module's firmware supports a 'no code' approach by pre-implementing all features that are required.

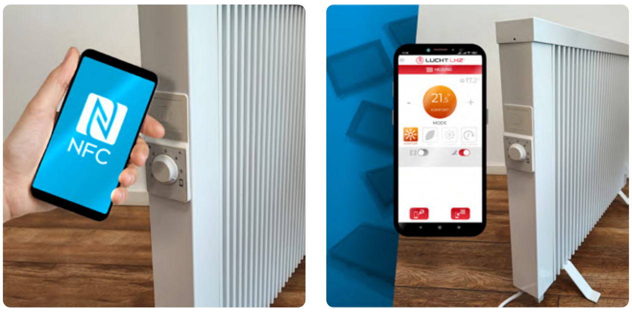


Figure 3: NFC is used to set up the communication.

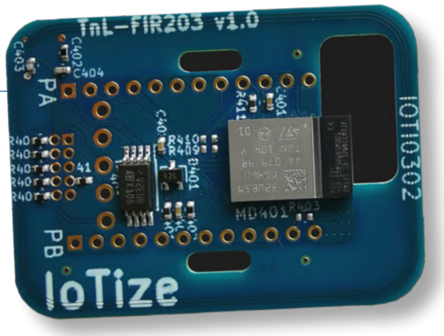


Figure 4: Modules for NFC, Wi-Fi and Bluetooth are pin-to-pin compatible with identical footprints.

Field Communication). It can be used either as a primary communication channel or as a companion to Wi-Fi and Bluetooth.

Historically, NFC is the technology used for contactless payments with Android or iOS smartphones. Based on this background, it offers multiple advantages when connecting smartphones to other appliances:

- **Security:** the communication takes place over a short distance. This protects against Man-In-the-Middle attacks. The user authentication is done immediately, and the construction of encryption keys promotes proximity-based confidentiality.
- **No energy consumption when not in use:** the energy for the NFC device comes from the electromagnetic field emitted by the smartphone during communication. Thus the equipment does not emit unnecessary, polluting radio waves. While used with Wi-Fi or Bluetooth, radio emissions can also be reduced because there is no need to constantly emit an advertising signal.
- **Automatic installation and launch of appropriate apps:** when a smartphone connects via NFC, the appropriate app is launched without the need to search for it on the smartphone. If the app is not installed already, the user is automatically redirected to the 'store' or to other sources from which it can be installed.
- **Optimized management of communication channels:** the module can be configured to wake up and pair the Bluetooth or Wi-Fi channels. There is no need to scan for available devices. No risk of connecting to the wrong Bluetooth or Wi-Fi device.

These NFC advantages have been so convincing, that appliance and equipment designers are using it for uses that include:

- **Three-stroke configuration** in which the NFC is a secure, low-cost, low-energy consumption smartphone interface for any appliance that requires configuration at commissioning or periodically through its life,
- **NFC wake-up and pairing** for appliances that use Wi-Fi or Bluetooth smartphone interfaces for on-site monitoring, testing, and firmware updates (**Figure 3**).

This was the case with electric heating appliance suppliers like Lucht LHZ who are deploying new NFC-equipped radiators to facilitate product configuration by installers and owners. And, thanks to the no-code approach to wireless integration, companies using the IoTize solution have been able to get a jump on the market and take advantage of this new opportunity early.

IoTize modules that pre-implement these NFC functionalities are already available, including the TapNLink NFC modules with ST25DV dynamic NFC tag, Wi-Fi modules with ESP32 and ST25DV, and a new ultra-low power consumption Bluetooth module that combines the STM32WB and ST25DV (**Figure 4**). All the wireless functionality is pre-implemented and qualified so that designers just select what they want to use.

Designers who want to try the complete hardware and software solution can get the TapNLink Wireless Primer from Elektor [1] and from dedicated distributors in most regions.

Open and generic interfaces

While IoTize's no-code solution was initially designed to connect smartphones to other appliances, it also solves other problems specific to remote monitoring in the IoT. Wi-Fi, LoRa, and LTE-M allow direct connection of devices to the internet, bypassing the smartphone and passing directly via a gateway. The addition of a Java Virtual Machine in the IoTize solution firmware makes it possible to support a direct network connection. The JVM can perform local calculations (edge computing), detect anomalies, and generate alarms. With very little code, it can also format data, and handle certificates and messaging with any IoT platform (AWS, Azure, etc.). These generic functionalities are destined not only to replace screens and keyboards but also to open a dialogue with equipment that was previously mute.

The new business of connectivity

Smartphones are fast becoming the universal tool for controlling appliances and equipment of all sorts. This evolution, combined with the complexity of the technologies involved, is leading companies to consider this connectivity as a business in its own right. In the same way, that car manufacturers don't make their own tires or injectors, manufacturers of other appliances are integrating generic, no-code connectivity solutions to address radio and cybersecurity issues, as well as those related to the generation of mobile apps. When faced with technical complexity, these solutions are more robust, simpler, faster to implement, and much more economical.

220375-01

About IoTize

IoTize is a Grenoble-based startup and expert in connectivity solutions. IoTize has been developing its solutions for six years and has already received multiple awards: twice awarded by NFC Forum, Innovation of the Year at Embedded World, IoT solution of the year at TRUSTECH, etc. Its development is focused on international markets in the areas of access control, HVAC (configuration), industrial control (monitoring), and intelligent sensors (alarms and IoT).

WEB LINK

[1] Clemens Valens, "IoTize Your Projects with TapNLink Wireless Primer," Elektormagazine.com: https://www.elektormagazine.com/news/review-iotize-tapnlink-wireless-primer

Audio Spectrum Analyzer with Dekatrons

A New Way to Use Vintage Tubes

By Charles van den Ouweland (The Netherlands)

We love to find new uses for vintage parts and especially old displays. They are often too nice to leave in a drawer. This audio analyzer project is an example. We give dekatrons a second life.

Dekatrons are gas-filled counting tubes dating from the 1950s [1]. For a short period in history, they were used as counters and memory cells in computers, like the Harwell Witch computer [2]. Each tube could hold one digit from 1 to 10, so you needed a lot of tubes to build a computer.

I had a couple of those tubes lying around, and I was thinking about a project in which they could be put into new use. Sometimes people who build nixie clocks add one dekatron as a second's counter, which is nice, but I wanted something new.

In this article, I am presenting a 7-band stereo audio spectrum analyzer with dekatron output. This is a great addition for any audiophile with a tube amplifier wanting to augment the visual spectacle!

The idea is simple: instead of representing the audio spectrum with a bar graph, using LED displays – or even more boring – an LCD, the volume of the audio signal in each frequency band is indicated by a number of dots lighting up in one of the dekatrons. Each dekatron forms a circle of dots. The left half of the dots represents the left part of the stereo signal and the right half the right signal.

The hardware of my audio spectrum analyzer (**Figure 1**) consists of a control unit for signal processing, and seven identical dekatron units for the display. And, of course, a power supply is also needed.

The Control Unit

Figure 2 shows the schematic diagram of the control unit and the dual power supply for the spectrum analyzer. Initially, I recycled and modified an existing board to build my own prototype for this circuit. By now, the schematic design file (Eagle file *msgeq7ctrler_Elektor.sch*) and pcb layout are available for downloading on this project's GitHub page [3].

The microcontroller (IC1) is an ATmega328P, well known from the Arduino. Here it's not used as an Arduino, though. The small (300-line) program is written in C using Atmel Studio.

To determine the audio spectrum, a dedicated MSGEQ7 chip from the American company Mixed Signals is used, that analyses the audio signal and divides it into seven frequency bands. The seven levels can then be obtained by a microcontroller by sending digital commands to the chip. The chip responds with a voltage level on its output, which is converted by the A/D converter in the microcontroller. There are two of these analyzer ICs in this design (IC2

High-Voltage Warning!

Dekatrons operate at supply voltages as high as 450 V. If you are not comfortable working with high voltages, don't even try to build this project or work with these tubes at all. Even after switching off the AC power, there can still be a high-voltage present on the high-voltage capacitor. Before working on the circuit, make sure the capacitor is discharged.

Neither the designer nor Elektor assumes liability for any damages whatsoever arising from the construction, testing, and use of (parts of) this project!

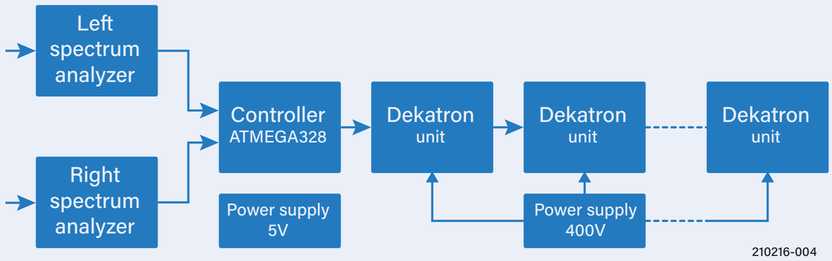


Figure 1: Block diagram of the audio analyzer.

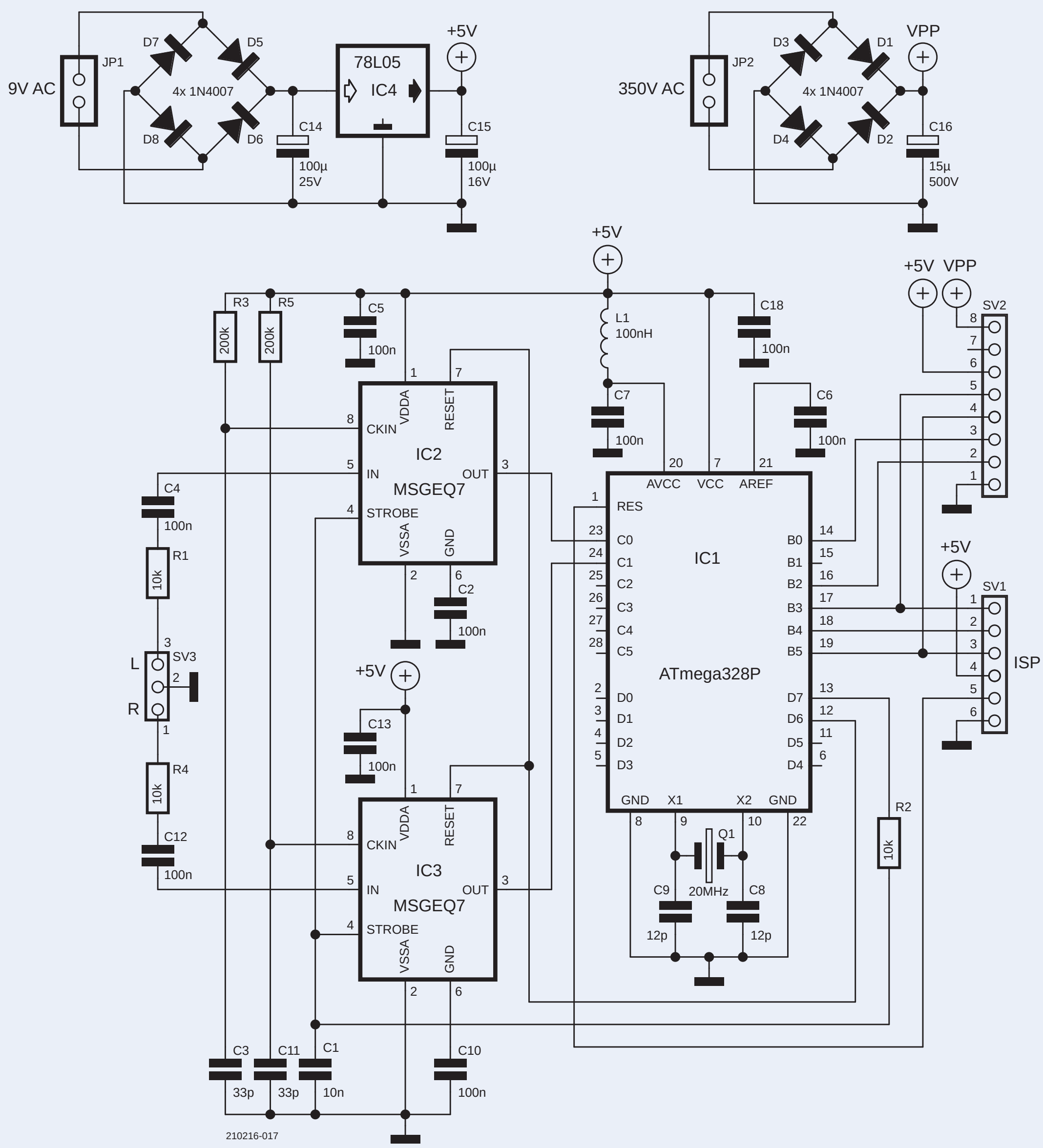


Figure 2: Schematic diagram of the main unit.

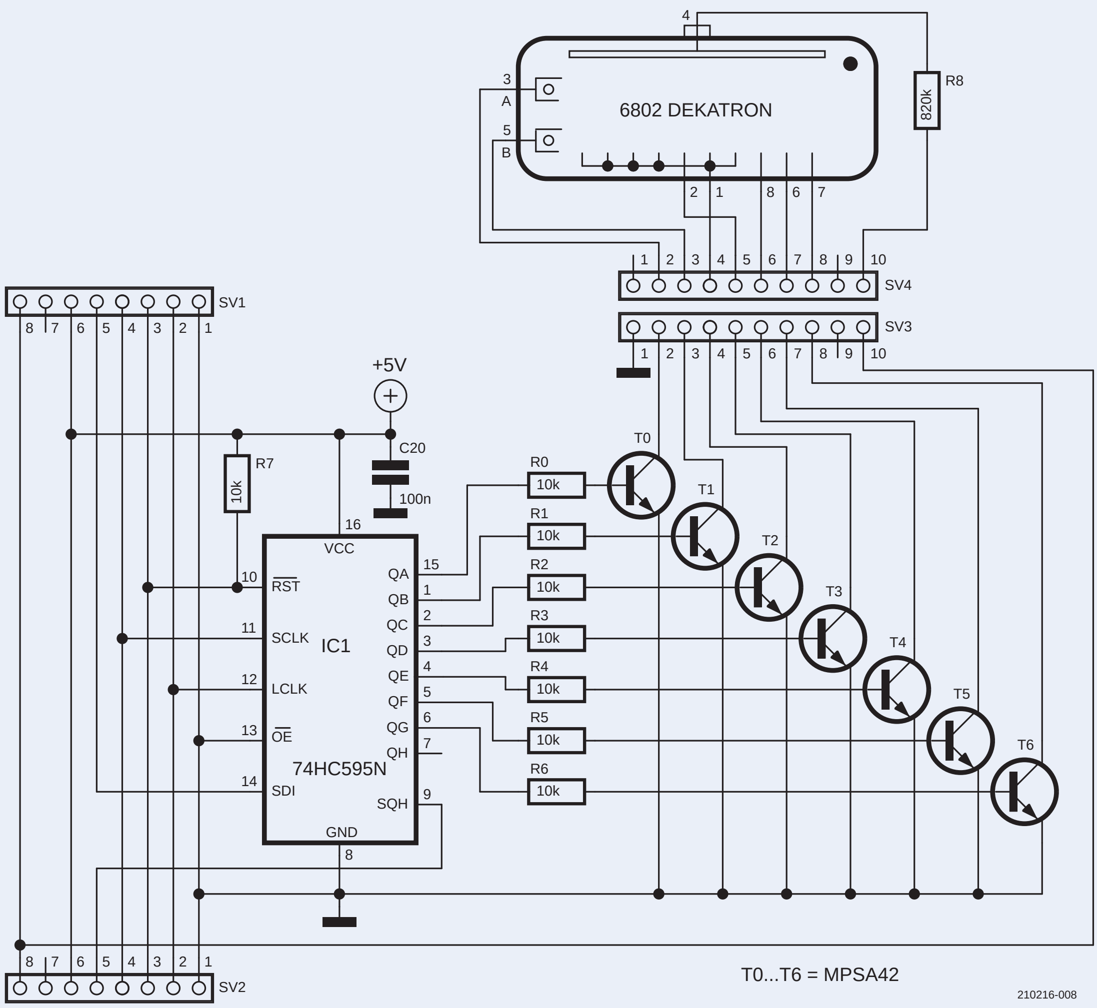


Figure 3: Schematic diagram of one dekatron unit.

and IC3), one for each audio channel.

Dekatron Unit

The microcontroller communicates with the dekatron units using three-wire serial communication, also known as SPI, at 5 Mbit/s. Each dekatron sits on a unit consisting of a 74HC595 (serial-in, parallel-out shift register) and a number of MPSA42 transistors, which are used as high-voltage drivers.

I designed a PCB (size 4 cm × 4.5 cm) for this unit, or actually two PCBs because one unit consists of two PCBs stacked on top of each other. **Figure 3** shows the schematic diagram. The top PCB contains the dekatron socket and the anode resistor, while the bottom PCB contains all other components. The holes for the dekatron's socket are suitable for different types of sockets (both ceramic and metal/plastic). The bottom PCB has a male connector SV1 on the left and a female connector SV2 on the right, so they can easily be daisy-chained to a string of arbitrary length. **Figure 4** shows the PCBs and assembled dekatron modules connected for testing.

PCB manufacturers often offer a prototyping service with ten PCBs for a fair price, so you have PCBs for 10 units at a bargain. The Gerber and drill files that you need to order these panels are contained in *Unit6802_Elektor.ZIP* [3].

Power Supply

The design needs two supply voltages: 5 V and 400 V. The electronic components needed for the supplies are also drawn in Figure 2. The 5-V supply is straightforward: a 9-V transformer with a diode bridge, a 78L05 voltage regulator, and some capacitors.

In the dekatrons, gas discharge takes place. In order for this discharge to ignite, a relatively high V_{PP} voltage of approximately 400 V is needed. My first solution was a switching supply built around an MC34063 or even a simple 555. But I found out that such a supply causes a lot of interference with the analog signals (especially on the output of the MSGEQ7), and it turned out near to impossible to get good results. Then I found a supplier who makes transformers according to customer specifications in China, Yollen Electrical Store. I ordered a couple of small 350 V, 1 VA, 50 Hz transformers from them at €3 each plus €6.50 shipping (**Figure 5**). Add four 1N4007 diodes (D1 to D4) and a 10 µF 500 V capacitor (C16) — that's all you need.

Even though this design uses tubes, the power consumption is very low. Dekatrons are so-called cold-cathode devices so there is no heater! The only power drawn by each tube is 0.3 mA wrth of anode current which adds up to approximately 1 watt of power on the high-voltage side. Both transformers are rated 1 VA and the total power consumption is below 2 W.

Software

The software for the ATmega328 is written in C in Atmel Studio, the source code and the HEX(adecima)l file for programming the microcontroller can be downloaded from the GitHub page [3]. The processing is either done in the main loop or in interrupt service routines. Interfacing with the analyzer chips (including control of the ADC) is done from the main loop, while interfacing with the dekatron units via SPI is done from interrupt routines.

The left and right audio signals are fed to two analyzer chips. Each of these contains seven bandpass filters (60 Hz, 150 Hz, etc., see **Figure 6**), and the output level for each band is sampled. The AVR gets the audio levels as analog signals, digitizes them, and uses the dekatrons as display devices.

Data Acquisition

In order to reduce noise at the software level, each value is measured 15 times and the best values out of these measurements are taken. As there are seven frequency bands and two channels (L and R), 210 analog to digital conversions are done each cycle, before the display is updated!

This works in detail as follows. Both analyser chips MSGEQ7 are reset. Then a strobe pulse is given to both chips. Now the level of the first frequency band is output on pin 3 of the chips. An analog to digital conversion (ADC) is performed on the left channel and then on the right channel. Both conversions are repeated twice, resulting in three values for Left and three values for Right. The lowest and the highest values are discarded, and only the middle values are kept. A second strobe pulse is given to both ICs, switching them to the second frequency band and the sampling process is repeated. The same goes

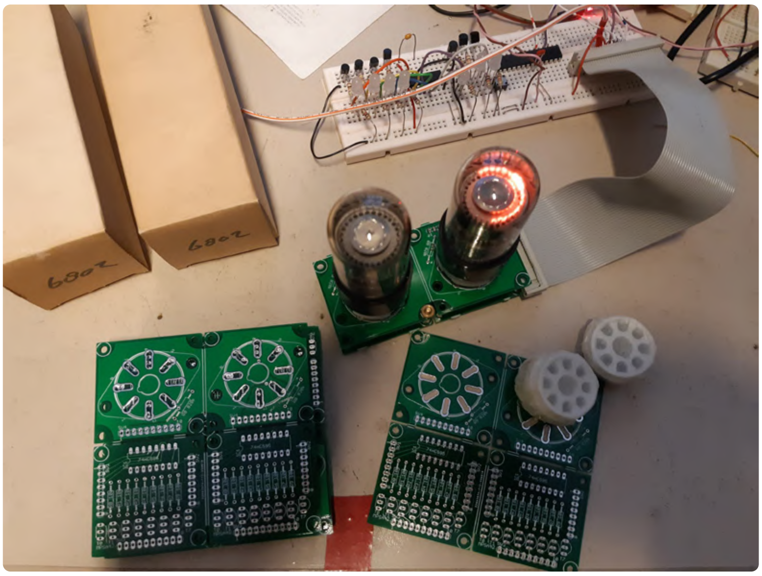


Figure 4: PCBs and assembled units.

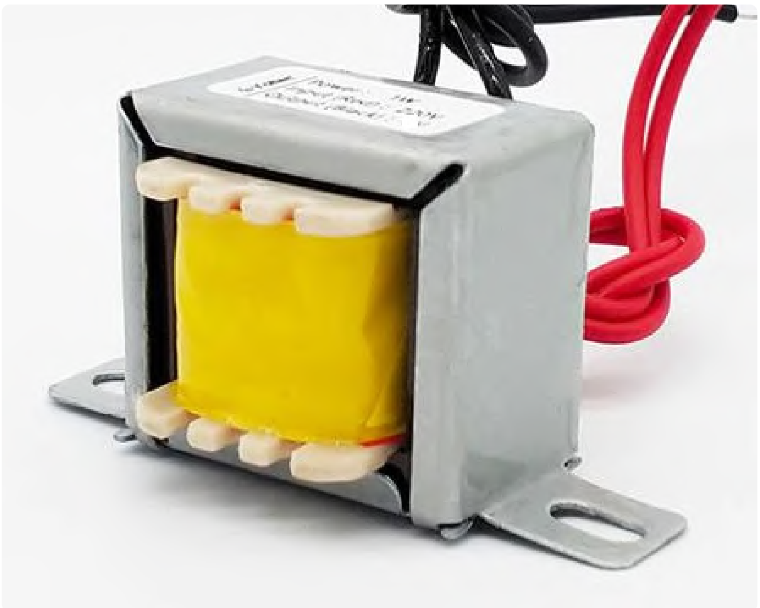


Figure 5: The AC power to 350 VAC transformer.

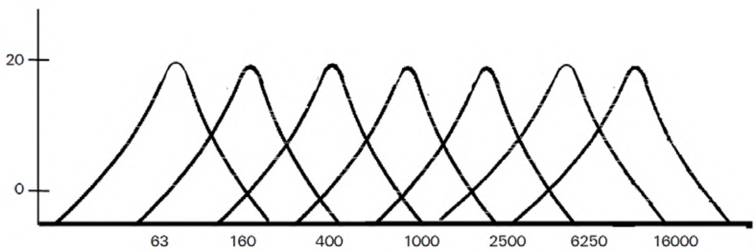
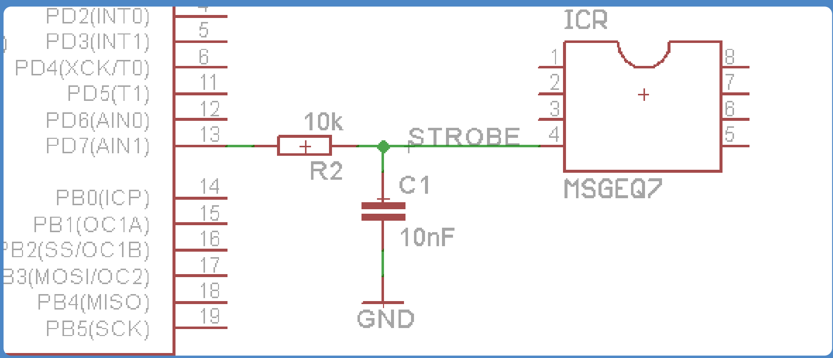


Figure 6: The MSGEQ7 frequency bands. (Source: Mixed Signal Integration)

MSGEQ7... Real or Fake?

Beware of fake chips! I bought some MSGEQ7s from a seller on Aliexpress and from a seller on Ebay and they both came up with fake chips. The ones I ordered from Sparkfun for $4.95 were the real deal. It's fairly easy to determine if you have a genuine or a fake chip. Just connect 5 V between pin 1 (V_{DD} =+5 V) and pin 2 (V_{SS} = 0 V) and measure the current. A genuine MSGEQ7 uses less than 1 mA. The fake ones I had guzzled more than 10 mA.

MSGEQ7 Output Noise



This chip was already covered in the May 2019 issue of Elektor [5], but what they didn't tell you is how to handle noise issues with that chip.

My first attempts in using the chip were not very successful because of noise issues. One in approximately every 10 values from the chip was just way off. On Internet forums, you see other people struggling with the same problem. I have taken a number of measures to overcome this.

First I tried to solve it in software, by sampling the same value multiple times and get multiple measurements from the chip. I discarded the highest and lowest values and averaged the rest of the samples. That helped a little bit. I also tried various hardware measures.

First of all, I tried physiscal distance between the microcontroller and the MSGEQ7. That didn't help at all. I tried separate power supplies for the micro and for the MSGEQ7, also in vain. Then I got rid of the switching power supply, I just resorted to a very simple power supply with mains transformers, diodes, capacitors and an 7805. That helped a little bit. In total desperation, I threw an RC filter in the digital path of the STROBE signal. To my great surprise, that solved the issue! You can find it in the schematic: C1 and R2 form a simple RC filter that admittedly make STROBE a terrible digital signal, but the MSGEQ7 still performs and the noise is (mostly) gone!

for the third up to the seventh frequency band. This data acquisition cycle is repeated five times and only the smallest values are used because I found out that the higher values are usually just noise.

Displaying Spectra

Controlling the dekatrons work as follows. As far as the program is concerned, each dekatron has 60 states. Why 60? Isn't a dekatron supposed to have 10 states, hence "deka"? Yes, originally the dekatron was supposed to have 10 stable states. But in between every two stable states are two transitional states. In total, the dekatron has 30 cathodes, 10 for each stable state, and between each stable state, there are two cathodes for the transition. Originally the dekatron was used as a counting tube. By applying a shifted pulse on the first and then the second transitional cathode, it could count "one up". This also means that, with a dekatron, it is easy to move the dot from one position to an adjacent position. But it is not so easy (and most of the time not even possible) to move the dot directly from one position to a random other position. This is only possible for a few of the stable positions. For some dekatrons, all ten stable positions have "random access", but other dekatrons have fewer. At the 6802 that I used only positions 0, 5, 8 and 9 have their pins exposed on the socket. In this design, the 0 position is initially used as a starting point and thereafter, the dot only moves to adjacent positions, either one step to the left or one step to the right.

In this application, we use all 30 cathodes. Furthermore, there is the possibility to energize two neighboring cathodes at the same time and this is used as a transition between one cathode and the next. In this way, we arrive at 60 states.

This application relies on rapid switching between states, so fast that it fools the eye. The current state of dekatron `i` (a number between 0 and 59) is stored in `values[i]`. Like a windscreen wiper, this is rapidly varied between two extreme states (**Figure 7**). The extreme states are stored in `minima[i]` and `maxima[i]`. The current direction of movement (towards minima or towards maxima) is stored in the Boolean `orient[]`. The state is changed every 150 µs. When 150 µs have elapsed, interrupt service routine `TIMER0_COMPA_vect()` is called. At this time, the new values to be sent out to the dekatrons are already prepared in `buffer[]`. The first value out of `buffer[]` is sent out over the SPI bus at a rate of 5 Mbit/s. When this SPI transfer is finished, interrupt service routine `SPI_STC_vect()` is called, which sends out the next byte over the SPI bus. When all seven bytes are sent out in this way, `SPI_STC_vect()` calculates the new `values[]`. The value of `buffer[]` is calculated, and it must contain the actual bit pattern to be sent over the SPI bus. The `map[]` array contains the translation from `values[i]` to `buffer[i]`. If necessary (when `values[i]` equals `minima[i]` or `maxima[i]`), `the orient[i]` is inverted.

The main loop changes `minima[i]` and `maxima[i]` based on the values from the ADC. The "windscreen wiper" is supposed to wipe between these two values. But sometimes the current position of the wiper (`values[i]`) is lower than the new `minima[i]` or higher than the new `maxima[i]`. When that happens, the function `checkRange()` detects that and changes the direction (`orient[i]`) of that wiper. By doing that, the wiper will automatically return to a position between `minima[i]` and `maxima[i]`.

Mechanical Construction

There were essentially two options: like a tube amplifier, put the tubes vertically, or mount them horizontally. I chose the latter because the tubes are top-view models. I wanted a housing for the project that could be placed in a stack of audio equipment. So I took the metal housing of an old piece of equipment (a cable TV decoder). I used it back to front and I added a new front panel, which is a piece of sheet metal taken from an old pc, with holes precisely drilled with a so-called step drill bit. Never drill a hole in sheet metal with a normal drill because it will not become round but more or less triangular! The holes were first carefully drilled with a 3018 CNC router, which I normally use for PCB milling.

Figure 8 shows the top view of the electronics of the audio spectrum analyzer inside the opened enclosure. The transformers and the PCBs were mounted on a piece of plastic household cutting board, which was then attached to the housing. The dekatron units were all screwed on a sturdy metal bar which was also attached to the enclosure. For safety, I connected the metal housing to protective earth.

Possible modifications

It is possible to simplify the design by making a mono model, which saves one MSGEQ7; or to reduce the number of frequency bands and thus the number of dekatron tubes needed. This can, for example, be done by merging the seven bands into five bands by combining the two highest and the two lowest ranges. It is also possible to use other dekatrons than the 6802, there are no particular features of the 6802 that make it more suitable for this application than any other dekatron.

Figure 7: Close-up of the wiper-shaped indication.

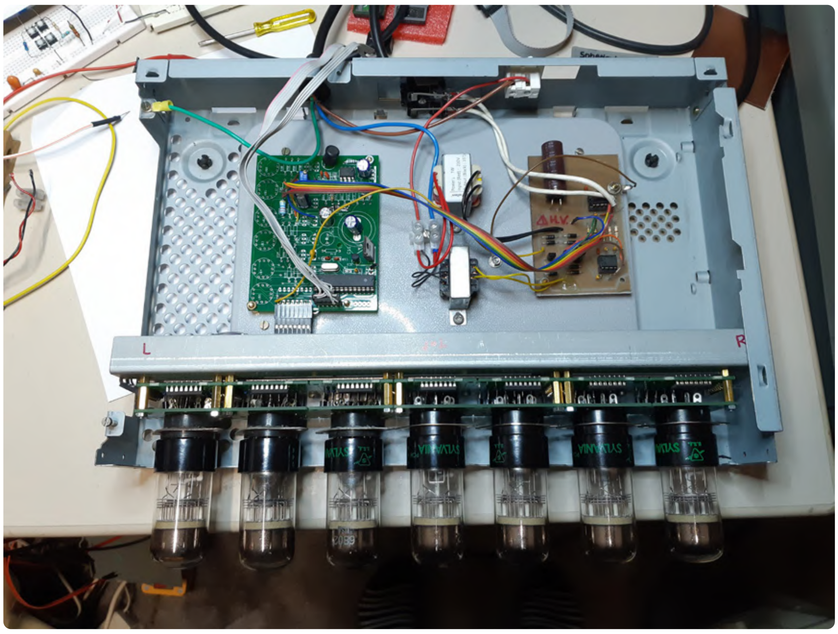


Figure 8: A view inside the enclosure of the audio spectrum analyzer.

COMPONENT LIST

Main unit

Resistors
R1,R2,R4 = 10 k
R3,R5 = 200 k

Inductors
L1 = 100 nH

Capacitors
C1 = 10 nF
C2,C4,C5,C6,C7,C10,C12,C13,C18 = 100 nF
C3,C11 = 33 pF
C8,C9 = 12 pF
C14 = 100 µF, 25 V radial
C15 = 100 µF, 16 V radial
C16 = 15 µF, 500 V radial

Semiconductors
D1...D8 = 1N4007
IC1 = ATMEGA328P
IC2,IC3 = MSGEQ7
IC4 = 78L05

Others
JP1,JP2 = 2-way terminal block
Q1 = crystal 20MHz
SV1 = 6-way SIL header
SV2 = 8-way SIL socket
SV3 = 3-way SIL header or stereo jack connector
Mains transformer 9 V 1 VA
Mains transformer 350 V 1 VA (see text)

Dekatron unit

EVERYTHING BELOW YOU NEED 7 TIMES!

Resistors
R0..R7 = 10 k
R8 = 820 k

Capacitors
C20 = 100 nF

Semiconductors
T0...T6 = MPSA42
IC1 = 74HC595N

Others
SV1 = 8-way SIL header
SV2 = 8-way SIL socket
6802 Dekatron with octal socket

This project shows that a dekatron can be used as a totally different kind of display than it was designed for: a bar (or rather an arc) of dots indicating signal strength. Of course, this principle can be used for other applications than audio spectrum analyzers. But the retro look of this spectrum display is great, especially when it is combined with a valve amplifier!

210216-01

Questions or Comments?

Do you have technical questions or comments about his article? Email the designer at labs@vanden.eu or contact Elektor at editor@elektor.com. Additional information can be found on this project's page on Elektor Labs [4].

RELATED PRODUCTS

- **Six Digit Nixie Clock with IN-14 Tubes (SKU 20044)**
  www.elektor.com/20044
- **Menno van der Veen,** *Modern High-End Valve Amplifiers* **(E-book) (SKU 18225)**
  www.elektor.com/18225
- **Menno van der Veen,** *High-End Valve Amplifiers 2* **(E-book) (SKU 18224)**
  www.elektor.com/18224

WEB LINKS

[1] Dekatron: https://en.wikipedia.org/wiki/Dekatron
[2] Harwell Witch Computer: https://en.wikipedia.org/wiki/Harwell_computer
[3] Github page with downloads: https://github.com/CharlesVanDen/AudioSpectrumAnalyserWithDekatrons
[4] This project on Elektor Labs: www.elektormagazine.com/labs/i-finally-found-a-useful-application-for-dekatrons
[5] Sunil Malekar, "Simple 7-Band Audio Spectrometers," Elektor 5/2019: www.elektormagazine.com/magazine/elektor-96/42640

Sending Data to Telegram

Get It Done with an ESP32 and a Few Parts

By Somnath Bera (India)

It is surprisingly easy to send data to a chat on Telegram. And not only that, it is quite fast too, allowing up to 12 messages per minute. Here is how to quickly build your own Telegram bot.

Publishing data on IoT platforms like ThingSpeak has gained immense popularity over the last years. Millions of channels have cropped up all over the world publishing useful and useless data. An advantage of ThingSpeak is that you can visualize your data as a pretty timeline trend curve. However, many a time you may not need to see data on a timeline. Rather, the raw data in quick succession will tell you everything that you need to know. For this a Telegram Bot channel really comes in handy.

Whereas a free account at ThingSpeak gives you a maximum of four data uploads per minute, the Telegram channel enables you to send up to 12 raw data packets per minute. This means that every five seconds a sample or measurement can be published, which is three times faster than on ThingSpeak. Of course, your Internet connection must be fast enough to achieve such a fast rate.

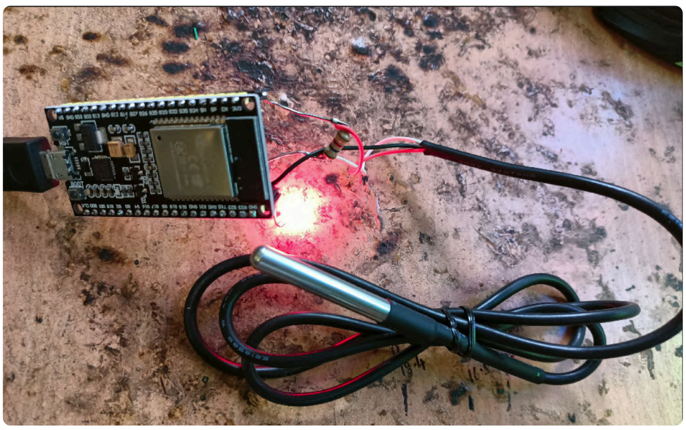


Figure 1: The prototype built for my Telegram bot experiments.

Project Idea

We can turn this idea into something practical with the help of an ESP32 to which we add an LED and a sensor of your interest. The sensor must, of course, be compatible with the ESP32 and it should be able to produce readings at a rate faster than every five seconds (>0.2 Hz). I used a one-wire DS18B20 temperature sensor. The LED indicates when data is being uploaded. **Figure 1** shows my prototype. Refer to **Figure 2** for the schematic of the circuit.

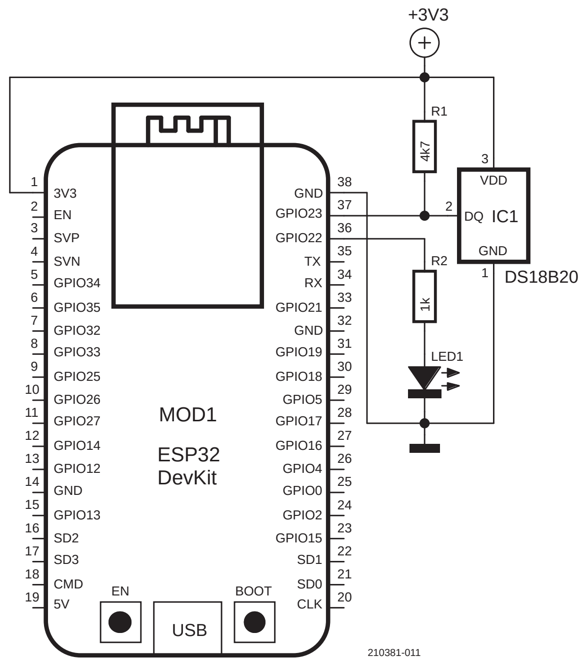


Figure 2: The Telegram bot reads the ambient temperature from a DS18B20 one-wire sensor.

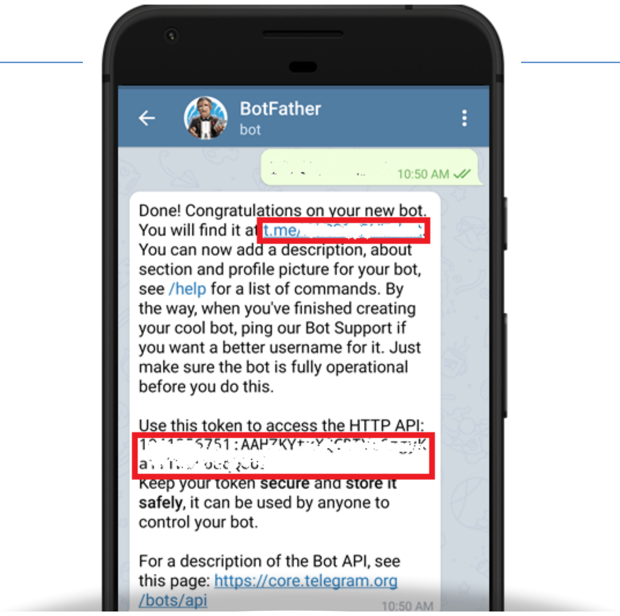


Figure 3: Use Telegram's BotFather to obtain an API token for your chat.

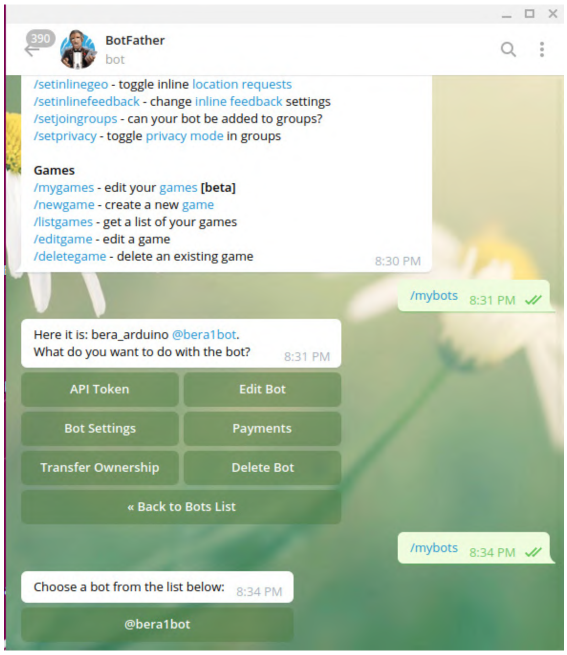


Figure 4: Use the `/mybots` command to access the list of all your bots.

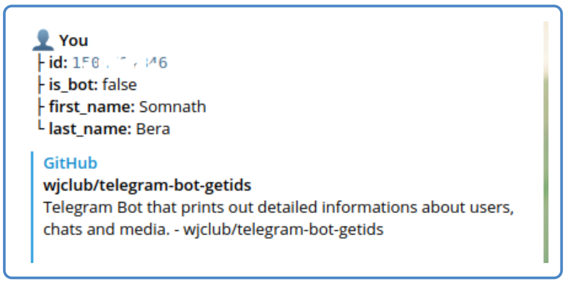


Figure 5: The chat ID is obtained by sending a message to a chat ID bot. There are several of them, this is @GetIDsBot.

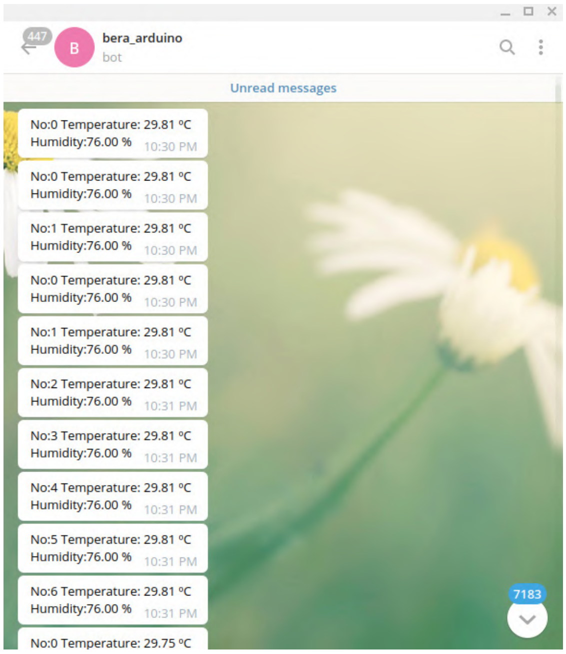


Figure 6: And away we go! Note that the humidity in these messages is just a hard-coded value in the software. My bot doesn't have a humidity sensor.

Build a Telegram Bot

To start, if you haven't done so already, you need to install the Telegram app on your mobile phone, tablet, or any of the other devices of your choice. You can get the free app easily from Google Play Store, App Store, etc.

After installing the Telegram app and setting up your account, you must create a channel which will then communicate with your ESP32(s). To do so, search the app for the 'BotFather' bot. When you open BotFather, you will see a 'Start' or a 'Restart' button. Tap or click it to open a list of commands and their applications. Then tap or click on the `/newbot` command and enter a name for your bot. I used *bera_arduino*. After naming the bot, you need to set the username. While setting the username you should keep in mind that it must be unique and it should end with 'bot', e.g. *bera1bot*. As soon as you have set the username, your bot will be created and you will see an API token (**Figure 3**). Save this somewhere as it is needed by the ESP32 program.

By the way, you can also obtain the API token by entering the `/mybots` command. Select the bot for which you need the token (**Figure 4**) and tap the *API Token* button.

The last thing required to get started is your chat ID. To get it, search for a bot named *@GetIDsBot* or *@myidbot* (IDBot) and send it a message. Anything will do. In response it will print some data about your channel that includes the ID (**Figure 5**). Write it down somewhere as it is needed by the ESP32 program.

Now you are all set with all the credentials to run your chatbot in Telegram.

Software

Download the example program from [1] and enter your API token (`BOTtoken`) and chat ID (`CHAT_ID`). Also, fill in your network credentials. Note that the program supports multiple Wi-Fi network connections. If one fails others will be tried automatically by the ESP32. Also, if the connection stalls or fails to transfer data to Telegram, the ESP32 will restart with another connection.

Most of the heavy lifting is done by the library *UniversalTelegramBot*, written by Brian Lough [2]. JSON support is required for this, therefore the *ArduinoJson* library must be installed too. Both libraries are available from the Arduino IDE's Library Manager. I have provided the facility for getting the software to work on ESP8266 as well.

Aftermath

With my fiber-to-home Internet connection, I get 12 readings per minute (i.e., one reading every five seconds), which is extremely good. If the network slows down, the number of uploads will be reduced. Now I wish Telegram also to have a trend recording like that provided by ThingSpeak.

210381-01

Questions or Comments?

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

RELATED PRODUCTS

- Joy-IT NodeMCU ESP32 Development Board (SKU 19973) www.elektor.com/19973
- Elektor Ultimate Sensor Kit (SKU 19104) www.elektor.com/19104

WEB LINKS

[1] This project at Elektor Labs: www.elektormagazine.com/labs/publish-by-telegram
[2] Universal Telegram Bot Library: https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot

A Fliege Notch Filter for Audio Measurements

Make Better Measurements with a Notch Filter

By Alfred Rosenkränzer (Germany)

When measuring the key characteristics of an audio circuit, for some parameters it is very desirable to remove the test signal from the circuit output before feeding the output signal to the input of the test equipment. That's exactly what this article is about.

As previously mentioned in the article "Low-cost Audio Tester" [1] in Elektor 07-08/2022, the USB audio interfaces that can be used with suitable software to set up an audio test station have a major shortcoming: they lack a switchable notch filter ahead of the input to suppress the test frequency component in the output signal of the audio device under test. But why is a notch filter so important?

Optimizing with a Notch Filter

The easiest way to illustrate the importance of a notch filter is to consider a test setup (as described in the above-mentioned article) intended to measure the quality of the audio interface itself. If you connect the input and output of the interface (loopback) and generate a typical test signal – usually a sinusoidal signal with a frequency of 1 kHz – you can view the harmonics in the resulting frequency spectrum. Unfortunately, with this arrangement, it's not possible to distinguish between the harmonics already present in the signal from the generator and the harmonics added by the test equipment (for example, in its A/D conversion stage).

With relatively high harmonic levels, such as from a fully driven audio output stage, this is not much of a problem because it's clear that the lion's share of the harmonics comes from the circuit under test, not the signal generator or the test equipment. When you want to measure the distortion characteristics of a good audio preamp, however, things are more difficult because the distortion of the device under test has roughly the same magnitude as the distortion of the test equipment.

To deal with this issue, professional audio test equipment has a notch filter ahead of the input stage. This filter strongly attenuates the amplitude of the test signal (i.e. its fundamental frequency). Removing the test signal frequency from the output of the device under test eliminates the problem of harmonic generation by the signal-processing circuitry in the test equipment since the test equipment only sees the signal artefacts added by the device under test. The level of these artefacts is much lower than the level of the test signal fundamental frequency.

A notch filter also provides another important benefit: strong attenuation of the test signal fundamental frequency allows the sensitivity or the gain of the test equipment to be increased significantly, so that very low-amplitude harmonics and distortion components, which otherwise would not be visible (due to the limited resolution of the A/D converter, for example), can be measured. If you increase the sensitivity of the test equipment by a factor of 10 with the notch-filtered signal and the signal to noise ratio of the test equipment is slightly less than 90 dB (typical for a 16-bit A/D converter), you can easily measure harmonics and noise as low as -110 dB. Not bad, actually.

The integrated notch filters in professional audio test equipment are switchable, which means the notch frequency can be set. These

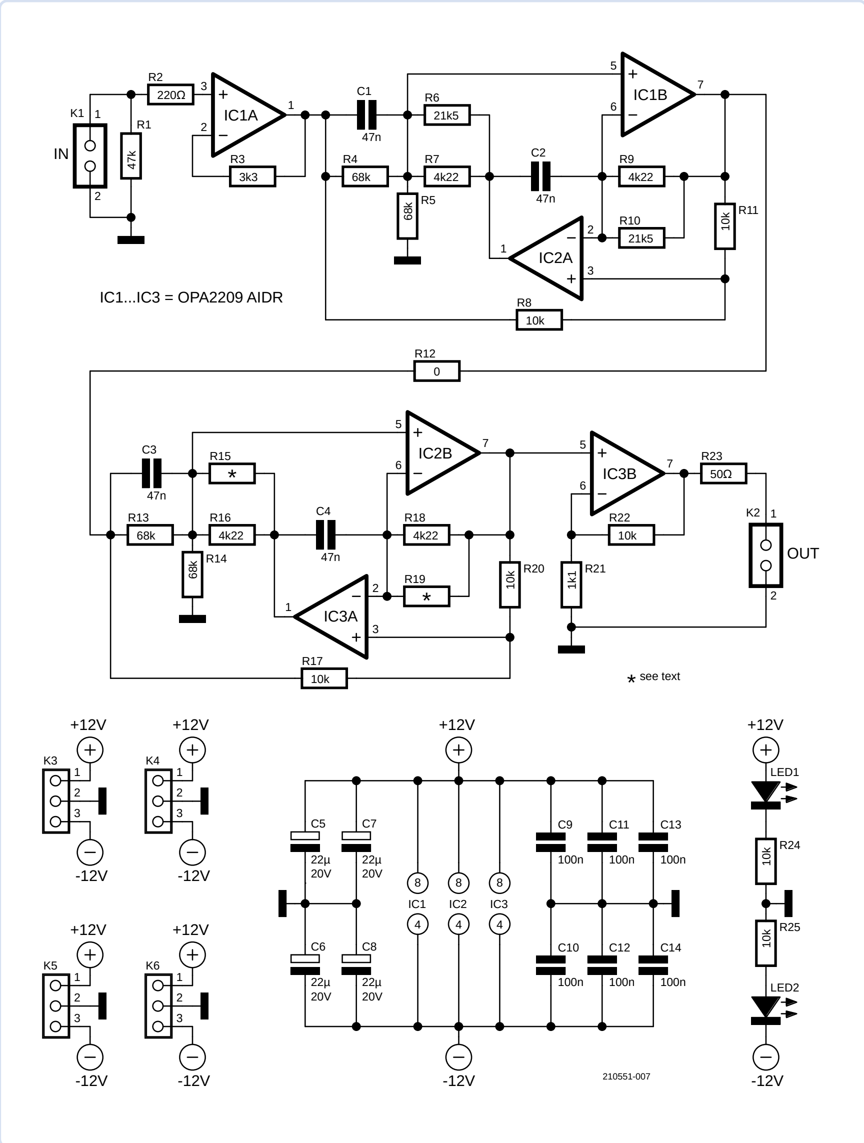


Figure 1: The circuit of the double notch filter needs only three (high-quality) opamps.

filters are usually designed so their filter frequency automatically tracks the generator frequency, which allows sensitive measurements to be made easily at different frequencies. As you might expect, such filter circuits are complex and costly.

Requirements and Limitations

A notch filter suitable for audio measurements should ideally attenuate the test signal as much as possible (in practice, by more than 50 dB) while passing the harmonics unchanged. As even the second harmonic should not be significantly attenuated and it is only twice the frequency of the test signal, the notch filter needs to have a high Q factor.

And, of course, the filter circuit should be designed so that it does not generate its own harmonics or any significant noise. This means you need very good opamps to build this sort of filter, but right now these are hard to find.

To keep the cost and complexity of a DIY notch filter within reasonable bounds, the notch filter described below does not have an adjustable frequency. On the other hand, it has an additional feature that is essential for use with USB audio interfaces, most of which do not have calibrated gain settings: the output signal is internally amplified by a factor of 10 (20 dB). This makes it easy to measure low-level distortion components.

Filter Circuit

To my knowledge, there are two suitable forms of implementation for this purpose: the Fliege notch filter and the state variable filter. I opted for the Fliege version on account of its characteristics, and I connected two Fliege filters in series to obtain enough attenuation.

Figure 1 shows the full circuit. Opamp IC1A buffers the input signal, so the first Fliege filter stage built around IC1B and IC2A is driven by a low-impedance source with defined signal conditions. The output of the first filter stage is fed through the 0 Ω resistor R12 to the second filter stage, which has the same component values. IC3B amplifies the filtered signal by 20 dB. If this is not desired, the gain can be reduced to 1 by omitting R21 and reducing the value of R22 to 3.3 kΩ.

Calculation and Tuning

To assist in the calculations, relevant instructions and software [2] are available online. The filter Q should be set to 10 to avoid excessive attenuation of the second harmonic. A higher Q would make it more difficult to determine the right component values since the limited gain bandwidth of the opamps would also have to be taken into account. In any case, the Q is boosted by the series connection of the two filter stages.

Resistors R4 and R5 together with R6 and R7 in parallel (or in the second filter stage R13 and R14 together with R15 and R16 in parallel) determine the Q of the filter. The formula for the Q of the first filter stage is:

$$Q = R4 / (2 \times R6||R7)$$

under the condition that R4 = R5 so that the gain is 1. The Q is nearly 10 because the value of R6||R7 is approximately 3.5 kΩ.

Capacitors C1 and C2, as well as R6||R7 and R9||R10, have the same values and determine the notch frequency according to the formula:

$$f = 1 / (2 \pi RC)$$

Resistors R8 and R11 also have the same value. The second filter stage has the same layout as the first.

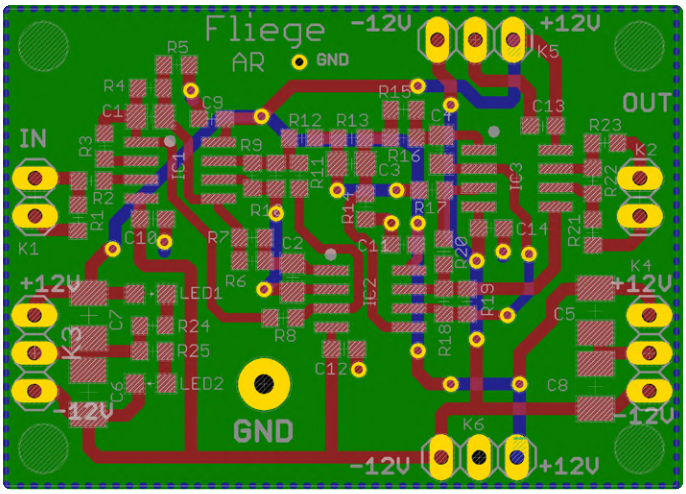


Figure 2: The PCB layout is designed for SMD components, but they are large enough to make manual soldering possible.

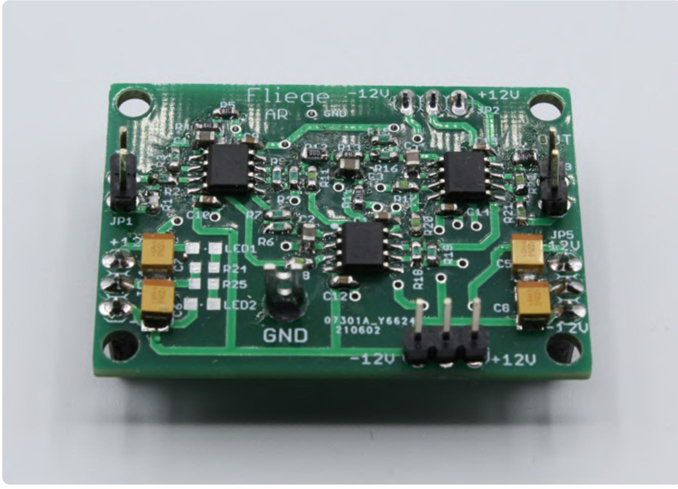


Figure 3: The author's fully assembled prototype filter board.

Capacitors C1 to C4 should have as nearly as possible the same value, for which reason they should be measured using a suitable LCR meter. If the actual value differs from the calculated value, the notch frequency can be fine-tuned using the parallel resistors R6 and R10 or R15 and R19.

In addition, both filter stages should not only be tuned to the test signal frequency but also have as nearly as possible the same notch frequency. This is where R12 comes in handy: it can be removed temporarily (or not fitted at first) to allow the two filter stages to be fine-tuned independently.

Then the output signal of the first filter stage can be taken and measured at the output of IC1B, and the first filter stage can be tuned to the desired frequency using R6 and R10. Once the right resistor values have been determined by experimentation, they can be fitted.

Next, the second filter stage has to be tuned. For this, its input (the junction of C3, R12, and R13) can be connected temporarily to the output of IC1A and the filter frequency likewise tuned using suitable values for R15 and R19. Once the notch frequencies of the two filter stages are sufficiently matched, this connection can be removed and R12 fitted. The notch frequencies do not have to match perfectly.

Final fine-tuning with resistors R15 and R19 can be done with the aid of an audio voltmeter by slightly varying the generator frequency. High attenuation of the target frequency is a sign of a high overall Q and indicates good overall matching of the notch frequencies of the two filter stages. When everything is satisfactory, fit resistors R15 and R19.

Circuit Board(s) and Filter Quality

Figure 2 shows the PCB layout for the circuit. The layout files in Eagle format can be downloaded free of charge from the Elektor webpage for this article [4]. **Figure 3** shows my fully assembled board. Connector pads for the 12 V supply voltages are provided on all four sides. This makes it easy to connect filter boards in series or plug a filter into a suitable board with voltage regulators to form a board sandwich as shown in **Figure 4**. By the way, the combined filter board and power supply board fit exactly into a type 1550Q Hammond aluminium enclosure, as shown in **Figure 5**. The component overlay on the PCB was omitted for the sake of simplicity, but the circuit and board layout files in Eagle format

COMPONENT LIST

Resistors (all SMD 0603, 1%)

R1 = 47 kΩ
R2 = 220 Ω
R3 = 3.3 kΩ
R4,R5,R13,R14 = 68 kΩ
R6,R10 = 21.5 kΩ
R7,R9,R16,R18 = 4.22 kΩ
R8,R11,R17,R20,R22,R24,R25 = 10 kΩ
R12 = 0 Ω
R15,R19 = see text
R21 = 1.1 kΩ
R23 = 50 Ω

Capacitors

C1,C2,C3,C4 = 47 nF, SMD 0805
C5,C6,C7,C8 = 22 µF, 20 V, electrolytic, SMC-B
C9,C10,C11,C12,C13,C14 = 100 nF, SMD 0603

Semiconductors

LED1 = LED, red, SMD 0805
LED2 = LED, green, SMD 0805
IC1...IC3 = OPA2209 AIDR, SO08

Others

K1,K2 = 2-pin pinheader, pitch 0.1"
K3...K6 = 3-pin pinheader, pitch 0.1"
PCB*
Power supply PCB*
Aluminium enclosure Hammond 1550Q

* See text

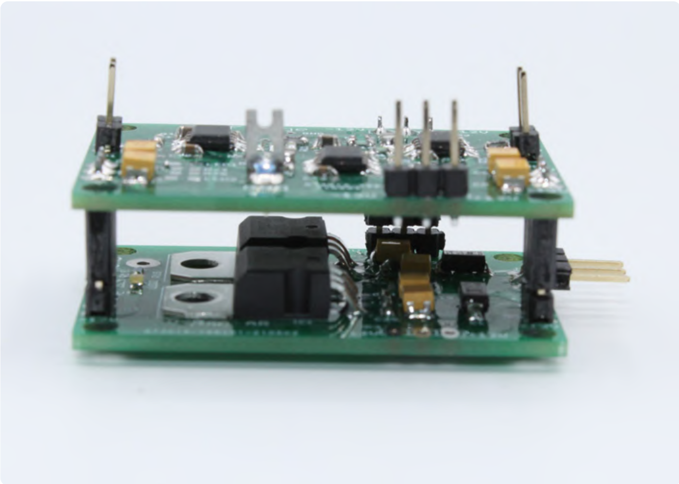


Figure 4: The filter board plugged into a simple board of the same size fitted with voltage regulator ICs.

Figure 5: The PCB sandwich installed in an aluminium enclosure.

can also be downloaded from [4]. I still have a few bare boards, so if you're interested you can contact me by email (see box). The linked archive contains the files for all the circuits, including a B version of the power supply board that I haven't tested yet.

Figure 6 shows the frequency response of the notch filter with the usual center frequency of 1 kHz. The attenuation of nearly 70 dB at 1 kHz is pretty good for the intended purpose. A spectrum plot (**Figure 7**) was generated using an Audio Precision APx555 in order to check the distortion and harmonic generation of the notch filter. Considering the cost and effort, the result — with over 90 dB SNR — is very respectable.

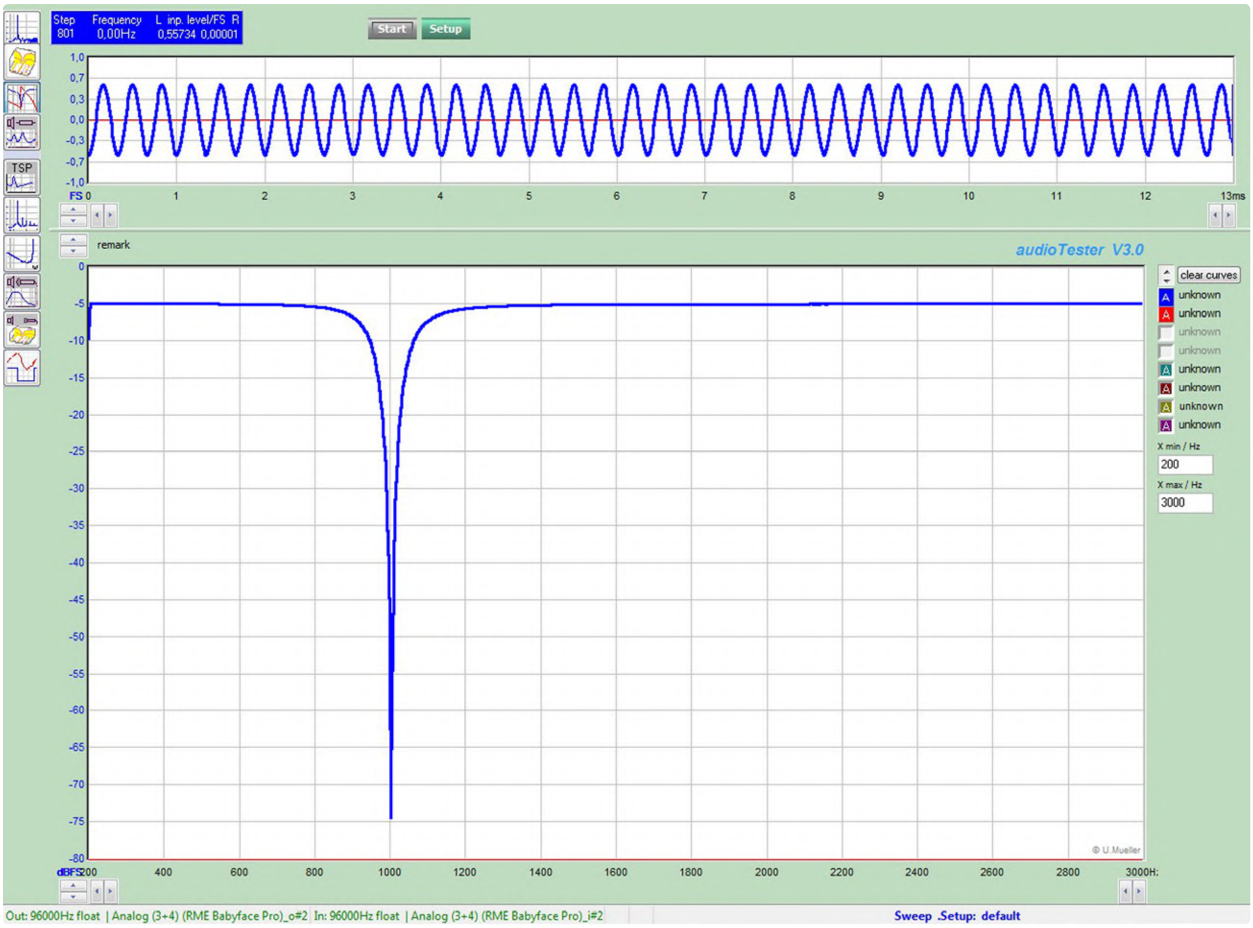


Figure 6: Frequency response of the double Fliege notch filter. The attenuation at the test signal frequency is nearly 70 dB, which is truly good.

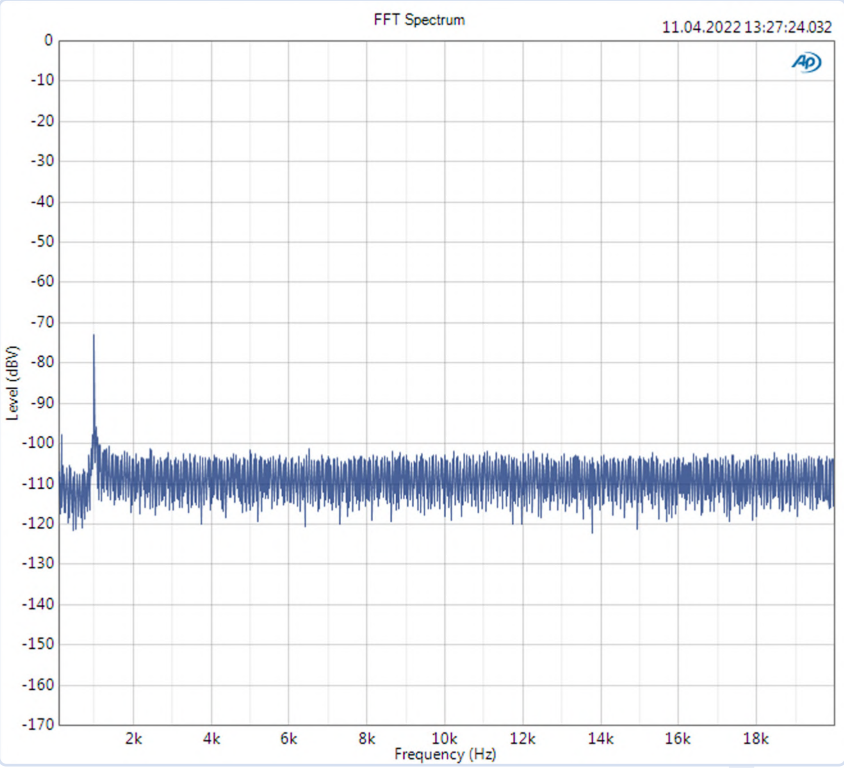


Figure 7: Spectrum plot of the notch filter. Its harmonics are buried in the noise floor at -100 dB.

Use

To establish a starting point, you should first use a switchable attenuator to set the level of the signal to be measured (without the notch filter) to a usable working range of approximately -5 to -10 dBFS (decibel relative to full scale) of the audio test equipment or USB audio interface concerned and note this level. Then connect the notch filter ahead of the instrument input. Sightly adjust the generator frequency to minimize the fundamental frequency level of this signal as registered by the instrument or audio interface. Then measure the levels of the harmonics. The gain of IC3B must be subtracted from the measured harmonic levels in this spectrum. For example, with 10x gain, a measured value of -80 dB becomes -100 dB. If the instrument or the audio interface has a calibrated or calibratable gain setting, you can further increase the overall gain for specific purposes in order to measure extremely low signal levels.

To ensure you make meaningful measurements with the notch filter, you should bear in mind that good measurements are only possible if the signal from the generator has the lowest possible distortion and noise. In some cases, it may be necessary to connect a narrow-band low-pass filter after the generator in order to obtain a truly good test signal.

210551-01

About the Author

Alfred Rosenkränzer worked for many years as a development engineer, initially in the field of professional television equipment. Since the late 1990s he has been developing high-speed digital and analog circuits for IC testers. Audio is his personal hobby.

Questions or Comments?

Do you have technical questions or comments about this article? Contact the author by email at alfred_rosenkraenzer@gmx.de or contact Elektor at editor@elektor.com.

WEB LINKS

[1] A. Rosenkränzer, "Low-Cost Audio Tester," Elektor 7-8/2022: https://www.elektormagazine.com/200604-01
[2] Notch filter calculator 1: https://earmark.net/gesr/opamp/notch.htm
[3] Notch filter calculator 2: https://tinyurl.com/yc2wj73v
[4] Circuit board layout download: https://www.elektormagazine.com/210551-01

CO_2 Meter Teardown

Is It Hackable for Your Projects?

By **Luc Lemmens** (Elektor)

Air quality measurement is a very popular subject in electronics nowadays, and CO_2 meters are one of the more hot topics in the DIY and maker's scene, judged by the number of CO_2 projects that are offered to Elektor for publication. But there are also affordable, commercially available CO_2 meters in a nice enclosure, with a color graphic LCD. We were wondering what's inside and if these devices or their components can be used for own developments. One way to find out: buy one, open it and do some reverse-engineering!

A device sold on Amazon as the "ETE ETEMATE CO_2 meter" caught our eye, which in addition to CO_2 concentration also measures and displays temperature and relative humidity. The same thing is available on Aliexpress (without brand name), also other models in a slightly different enclosure, or with more or different quantities on the display. This suggests that there is a standard platform for a range of measuring devices, depending on the software on the processor board and — of course — on the sensors that are connected.

The meter arrived in a completely white carton box, containing the device itself, a short micro-USB charging cable and a multi-lingual Instruction Leaflet. Nor the label on the box, nor the instructions reveal any information on the original brand or manufacturer. The device contains a rechargeable 1200 mAh lithium battery, which apparently was completely drained: the meter didn't switch on when the power button was pressed. Connecting a micro-USB charger helped: a charging symbol appeared on the LCD, and eventually the CO_2 concentration, temperature and relative humidity were displayed. I was a bit surprised to see that it showed 25°C for room temperature (and I was sure it was colder!), but for that moment I didn't pay much attention to that. After all, the battery was completely flat and most of these instruments need some burn-in time to show accurate measure-

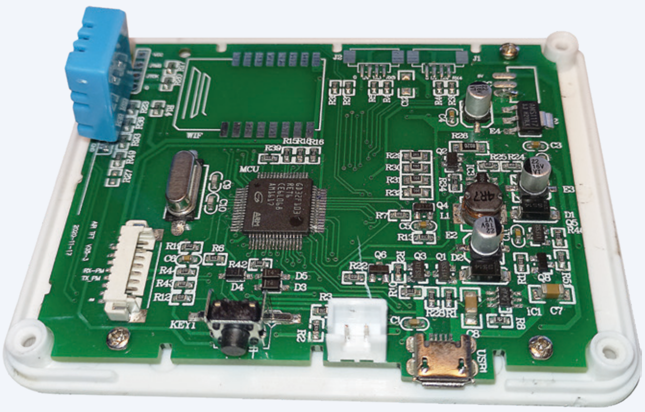


Figure 1: Overview of the main PCB.

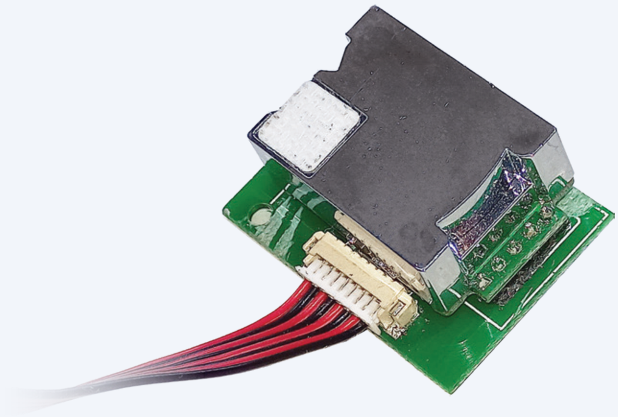


Figure 2: The CO_2 sensor — a clone of the MH-Z19?

ments. I decided to keep it on the charger during the night. The next day, I still saw the nice graphic LCD showing a slightly fluctuating CO_2 concentration, but rock-steady 60% relative humidity and (still!) 25 centigrades. That didn't look good... I unplugged the charging cable, trusting that the battery was fully charged, but the display went completely dark. Reconnecting the charger didn't help this time, no way that this thing was going to switch on again.

Normally, this would be the point to return the item to the supplier and ask for a replacement or refund. But in this case, time to write this tear-down story is short and for this goal a working CO_2 meter is not really required, after all we just want to know what's inside the enclosure. And to be honest: wouldn't it be nice to (maybe) repair a device that otherwise would be condemned to land on one of the gigantic piles of electronic junk we are producing and literally are consuming? Even if I'm not able to repair it, at least it would have served its purpose of being instructive for this article.

Opening Up

The backside of the plastic enclosure is produced in one piece, the display is attached to the lid, that is in some way fixed to the case. (Attached with screws or tabs? Glued?) Prying the lid out could be tricky: displays can be very fragile and easily break if too much force is applied. I first used my trusted Swiss Army knife to pry between the side of the display cover and the enclosure, and apparently it was easy to remove a thin plastic foil that covers the display and the lid and lo and behold: four small phillips screws were revealed. So the good thing is: the device can be opened and closed without any damage!

A PCB with most of the electronics and the LCD are attached to the lid, only the battery and the CO_2 sensor are attached to the back of the enclosure.

Temperature/Humidity Sensor

Most of the components on the main PCB (**Figure 2**) can easily be identified, with the most notable part the blue plastic DHT-11 temperature and relative humidity sensor [1]. This is certainly not the most accurate sensor, but it does the job for a relatively simple consumer device like this CO_2 meter; it is affordable and widely available.

The Microcontroller

This is probably the most important component to investigate. If it's difficult or impossible to develop and program new firmware, we can simply forget that we can use this device for new designs. Fortunately, that won't be an issue: in this meter a GD32F103-RET6 [2], an ARM Cortex M3 32-bit MCU from GigaDevice Semiconductor Inc. is used to process the sensors' readings and display them on the TFT display. The PCB has a 4-pin TH-footprint for the SWD debug/programming interface marked SWD1, next to the temperature/humidity sensor. The GD32F103 is pin-compatible with 64-pin PLCC STM32 microcontrollers, so if you prefer the latter, even if you don't have SMD reworking tools, you could replace this MCU using a small, normal soldering iron.

Power and Charger

About one third of the PCB is occupied by the power and lithium battery charging circuitry. Nice to have, of course, but not of real importance for new or own developments. The bad news about this section is that there are some SMD ICs that I couldn't identify. From initial measurements on the PCB, it looks like it's powered on when the power button is pressed and that an output of the MCU keeps it alive once the device has successfully booted. That would also explain the manual stating that the button must be pressed for about three seconds to switch the CO_2 meter on, which suggests that there is some interaction or feedback between the microcontroller and the power circuit. It could be, for example, that the lithium battery voltage is measured (deep-discharge protection) or that the presence and functioning of the sensors and/or display is being checked before the meter will stay powered on. Hard to tell without knowing what the firmware exactly does, or without exact knowledge of the power supply circuit.

The CO_2 Sensor

A CO_2 meter needs a CO_2 sensor, of course. Just like the meter itself, there is no type number or brand to be found on this module (Figure 2). Looking at pictures of other sensors on the Internet, it resembles the MH-Z19 [3] from Winsen Electronics Technology the most, but it certainly is not the same thing. The bottom side of the unknown sensor is covered by a break-out board that routes the two SIL-pin headers of the sensor to a connector for the wiring to the main PCB. Removing this small PCB revealed that only the power supply and

Rx/Tx are routed, the pinning is identical to the MH-Z19. There is also a PWM output, so it's most probably correct to say that is a clone of the well-known CO_2 sensor from Winsen.

LCD

That leaves one very important part: the display. The main PCB is mounted to the lid of the enclosure with four screws and the display is secured to this PCB with double-sided tape. Be careful when you disassemble this; the display and processor board are connected by a flexfoil cable that can easily be damaged and is hard (if not impossible) to replace. Fortunately, there will hardly be any reason to take it apart. We already did this for you to find out what type it is: the marking "CL028-04" on the flexfoil points to a 2.4", 240 x 320 pixel TFT display, which is a ILI9341-based [4], 37-pin parallel port LCD screen.

Not Mounted...

And then there are some empty footprints for components on the main PCB, probably for other applications or future developments. First of all, there's an footprint marked WIF1, which is most likely reserved for an ESP12 (ESP8266-based) module (see **Figure 3**). On this footprint, only VCC, GND, RxD and TxD are routed, which is good enough to add a basic WiFi-interface to the board.

Finally, there are two empty footprints marked J1 and J2, most probably for mini-USB connectors, and a 3-pin power connector. These can always be useful when you want to add more hardware.

No Guarantee

As mentioned earlier, this device is being sold all over the web, under different brand names, in different enclosures and for prices ranging from about €20 to €80. At the lower end of this price range, you will certainly get value for your money, with an ARM Cortex M3 32-bit MCU, a graphic colour LCD, two sensors, and a lithium battery with micro-USB powered charging circuit. The enclosure can be reused, and there is enough space inside left for extra hardware. There are some (I/O) pads on empty footprints, so even if the sensors remain attached, there are some spare I/Os accessible. But of course, more reverse engineering needs to be done. A complete schematic may be needed if you want to make your own applications with this hardware.

And will you get exactly the same hardware if you buy a CO_2 meter that looks like the one I discussed here? You most probably will, but unfortunately, I can't guarantee that!

210180-01

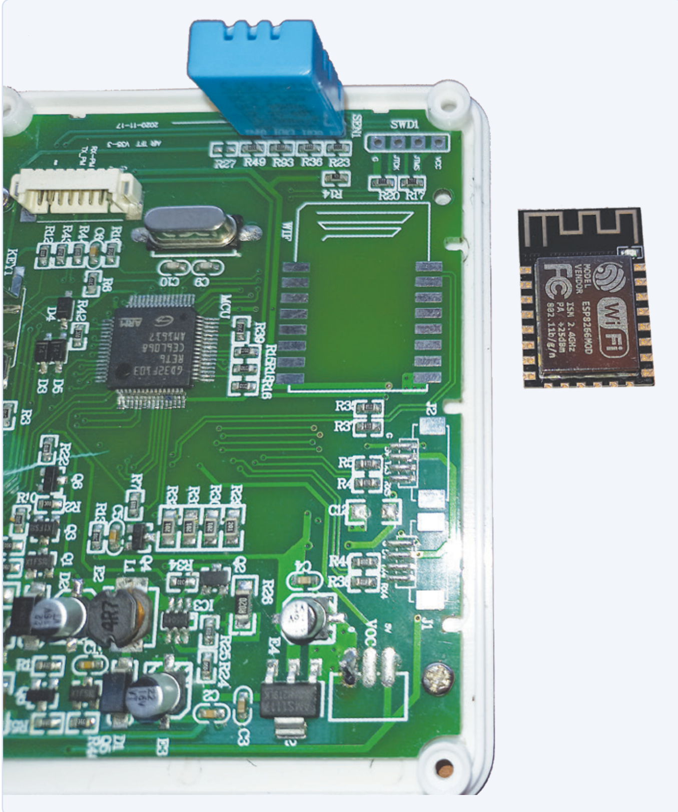


Figure 3: An ESP12F module is a very likely candidate for WIF1.

Contributors

Text and Illustrations: **Luc Lemmens**
Editor: **Jens Nickel and C.J. Abate**
Layout: **Giel Dols**

Questions or Comments?

Do you have questions or comments about his article? Email the editor at luc.lemmens@elektor.com!

WEBLINKS

[1] **DHT-11 temperature and relative humidity sensor**: https://bit.ly/3jLfUPp
[2] **GD32F103-RET6 Datasheet**: https://bit.ly/3hiMF4M
[3] **MH-Z19 CO2 sensor**: https://bit.ly/3wf1B8g
[4] **ILI9341 LCD driver**: https://bit.ly/3dNbOm0

PUT-ting It All Together

The Programmable Unijunction Transistor Explained

By Roel Arits (The Netherlands)

Do you know (or maybe remember) the Programmable Unijunction Transistor (PUT)? It was a special part in the '70s. In this article, we not only explain how it works, we also bring it back to life with a small circuit collection.

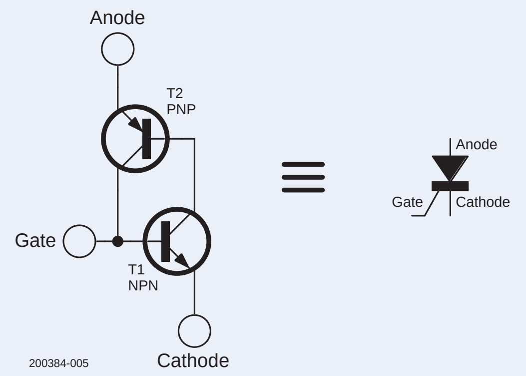


Figure 1: The Transistorized Universal Thyristor (TUT).

Please Welcome the PUT

Let's step into the world of old-school analog electronics and see what kind of interesting things can be accomplished with the rather simple transistor combination of **Figure 1**. These two complementary transistors feature self-latching behavior similar to the thyristor (a.k.a. SCR) and the Unijunction Transistor (UJT). If, in the circuit of Figure 1, we add a second gate, we obtain the Programmable Unijunction Transistor, or PUT, of **Figure 2**. Note how the gate on the thyristor symbol has moved to the anode.

The UJT and PUT were popular devices in the seventies because it was easy to build sawtooth– and pulse generators with. They are still being produced, but you rarely see them in modern designs. In this article, we will present some circuits that can be built around a PUT.

But First, Our Old Friend the Thyristor

The circuit in Figure 1 behaves like a thyristor, but it is not a direct replacement for it. Therefore, the circuit is also called a Transistorized Universal Thyristor (TUT). Let's first recall how a thyristor works.

When a small current (a few milliamperes) is injected into the gate, the thyristor will start to conduct provided that the anode-to-cathode current is above a certain threshold, the latching current. Once the thyristor is conducting, the gate can be disconnected. The only way to make the thyristor stop conducting is to lower the anode-to-cathode current below the minimum holding current. A way to achieve this is by briefly interrupting the power supply.

A PUT Behaves Like a Thyristor

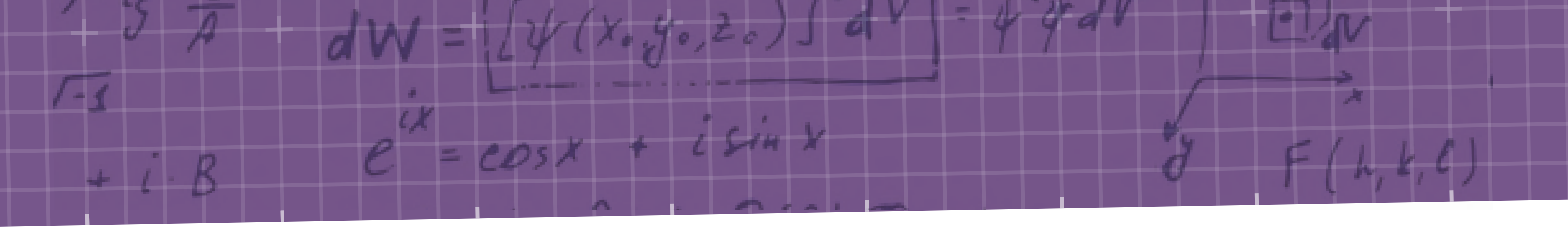
To understand how the PUT works, refer to **Figure 3**. There are two gates, G– and G+, but if we limit ourselves to G+, then the description is also valid for the UJT.

At power on, S1 and S2 are open. T2 blocks because its base is pulled high by R1 and T1 blocks because its base is floating. Since T2 is not conducting, the output is pulled high by R2.

Now, when S1 is pressed, the base of T1 is pulled high and T1 starts conducting. This will pull the base of T2 low, making T2 to conduct too and the output goes low. Now the base of T1 receives current via R2 and T2, allowing S1 to be released while both transistors keep conducting. So, when S1 is pressed once, the output goes low and stays low. The only way to reset the circuit is to briefly disconnect it from the power supply. This is the same self-latching behavior as shown by a thyristor.

And Now Using the Other Gate

Pressing S2 instead of S1 produces the same result but in a slightly different way. In this case the base of T2 is pulled to ground, making it conduct. The output will go low and the base of T1 starts to receive current via R2. T1 begins to conduct. This in turn will pull the base of T2 low, allowing S2 to be released. As before, the circuit has latched with the output low.



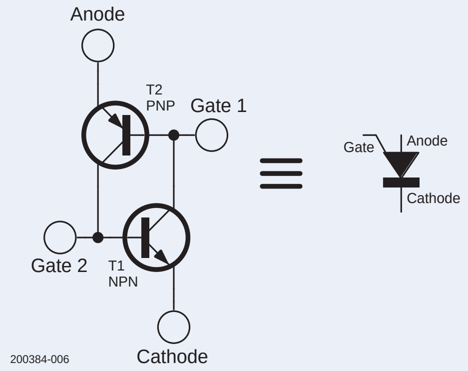


Figure 2: Two gates make the TUT programmable and we obtain the PUT.

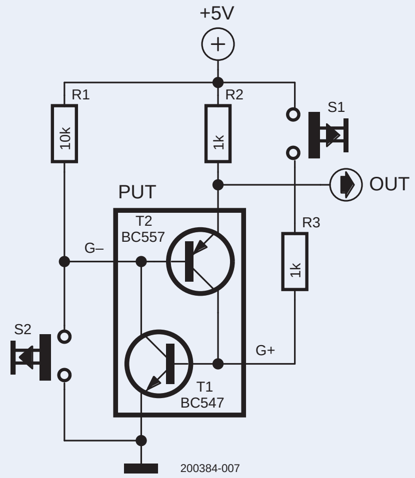


Figure 3: Press either S1 or S2 to make the output Low.

An UJT has one gate, a PUT has two (see Figure 2). The difference between the two gates of a PUT is that when gate 1 (G–) is used, the PUT will conduct when the anode voltage exceeds the voltage on gate 1 with 0.7 V (base-emitter voltage drop). When using gate 2 (G+), the PUT will conduct when the cathode becomes 0.7 V lower than the voltage at gate 2. This is where the 'P' in PUT comes from. The trip voltage at which the transistor starts conducting can be 'programmed' by applying a reference voltage to the gate of the PUT.

A Simple PUT Circuit

The circuit in **Figure 4** illustrates the behavior of the discrete PUT. Because R1 and R2 have identical values, the voltage at node 1 is half the supply voltage (i.e., 2.5 V). The voltage at node 2 can be set with a potentiometer.

If the voltage at node 2 is set to a voltage of at least 0.7 V below the voltage at node 1, T1 will start conducting. Now current can flow through the base of T2 via T1, and T2 becomes conducting too. T2 pulls the base of T1 to the positive rail, ensuring that T1 keeps conducting. The transistor combination is latched and will remain like that even when the voltage at node 2 goes up and becomes higher than the voltage on node 1. The circuit will unlatch when the voltage at node 2 gets close to the supply rail as it prevents current to flow through T1.

The circuit in **Figure 5** is the 'upside-down' version of the circuit from Figure 4. It works similarly, except that the voltage on node 2 now has to be 0.7 V above the voltage at node 1 for the transistor combination to latch.

Of course, in both cases the PUT will also unlatch if the power supply is interrupted briefly.

A Latching Switch

The circuit from **Figure 6** is a practical application of Figure 3. When the power is connected, the LED will light up, getting its current via R1 from

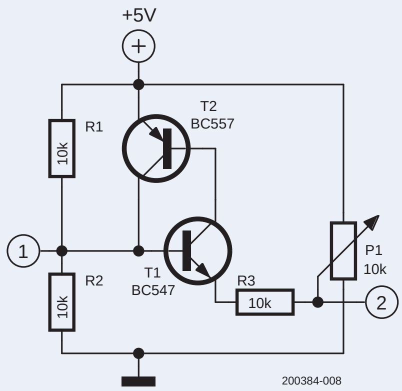


Figure 4: Turn potentiometer P1 down to make the circuit latch.

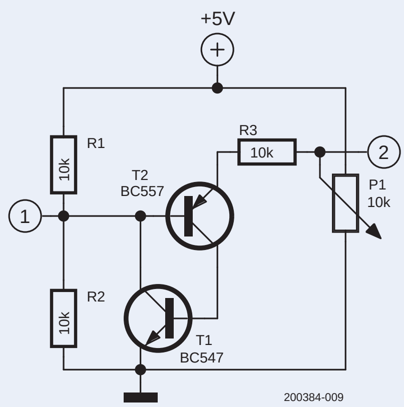
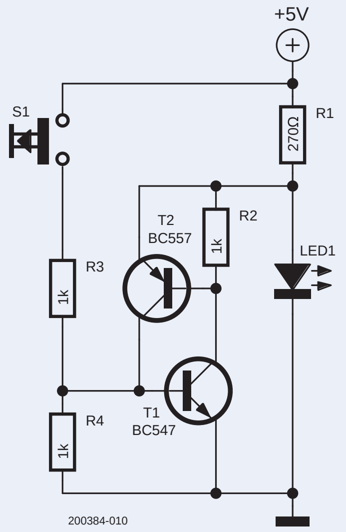


Figure 5: Figure 4 upside down. Now, P1 needs turning up for the circuit to latch.



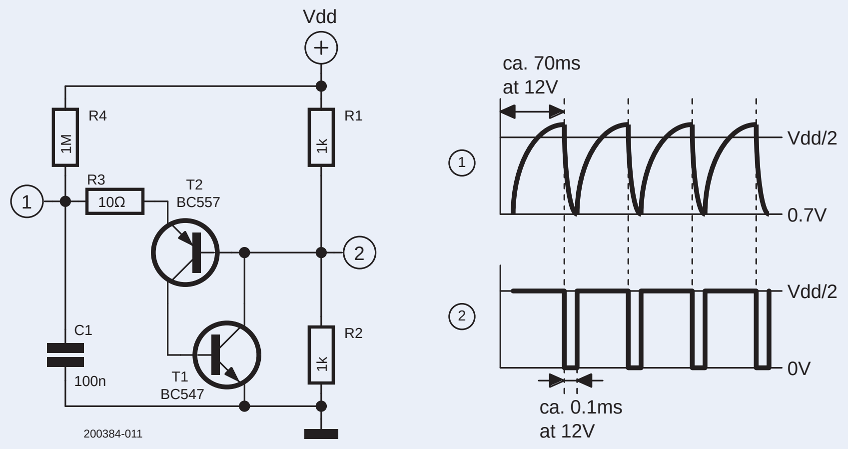


Figure 6: A simple off-only switch.

Figure 7: The simplicity of this relaxation oscillator design is one of the reasons why the PUT was used a lot.

the power supply. The PUT formed by T1 and T2 is not active because the base of T1 is pulled down to ground by R4 and the base of T2 is pulled up to the voltage over the LED. So, neither transistor is conducting.

When S1 is pressed, the circuit latches into conduction (see above) and shunts the LED, which will go off. After releasing the switch, the PUT combination remains latched and the LED will stay off. Interrupt the power supply briefly to switch the LED on again.

Relaxation Oscillator

A relaxation oscillator (as opposed to a harmonic oscillator) using the PUT circuit is shown in **Figure 7**. This circuit works with a wide range of supply voltages and generates a non-linear sawtooth waveform with a rising ramp at node 1. At node 2 we find a pulse signal.

The slope of the ramp, and therefore the frequency of the output signal, is determined by R4, C1 and the supply voltage. Increasing the value of R4 or C1 will decrease the frequency, as will increasing the supply voltage. The pulse width of the pulses that are generated when the ramp restarts is controlled by R3.

How Does It Work?

Initially C1 is discharged, so node 1 is at 0 V. T2's base-emitter is reversed biased (the base of T2 is at Vdd/2, while its emitter is at 0 V), so T2 will not conduct. Because T2 blocks, T1 doesn't get any base current and will not conduct either. The voltage over C1 (node 1) increases while the capacitor is being charged through R4. When the voltage over C1 reaches the point where the emitter-base junction of T2 becomes forward biased (at about Vdd/2 + 0.7 V), T2 will start conducting, and the circuit latches as described before. The voltage at node 2 will drop to 0 V.

However, since T2 is now conducting, C1 will quickly discharge through R3. When the voltage over C1 becomes lower than 0.7 V, T2 starts blocking and the circuit unlatches. The voltage at node 2 jumps back to Vdd/2 and C1 can charge again through R4. This process repeats itself over and over. The width of the low-going pulse at node 2 can be adjusted within certain limits by changing the value of R3.

If you rearrange the circuit in a way similar to Figure 4 and swap C1 and R4, the output signals will be inverted.

Some PUTfalls

The value of R4 should be high enough to allow the PUT to unlatch. It is therefore preferable to play with the value of C1 to adjust the oscillating frequency.

When the values of R1 and R2 are too high, the PUT does not receive enough gate current to firmly latch and C1 will not be discharged deep enough to unlatch the PUT again. So, it is better to keep these values low. When decreasing the value of R4, the values of R1 and R2 should be decreased too to allow the PUT to discharge C1 deep enough to unlatch again.

The value of R3 cannot be increased indefinitely. When it is too high, the PUT will not get enough current to latch firmly and cannot discharge C1 far enough to unlatch again. As a result, the oscillator will not start when R3 is too high.

Symmetrical Square Wave Generator

By adding a resistor and a diode to the circuit from Figure 7, it can be made to generate a symmetrical square wave (**Figure 8**). Because of D1 the lower side of C1 is no longer fixed to ground, but can be pushed below zero, going negative.

When C1 is charging, D1 is forward biased and the junction between C1 and D1 will be at +0.7 V. As before, when T2 starts conducting it will pull node 1 down to 0 V. When this happens, the voltage at node 3 will go negative, because C1 is still charged. D1 is now reverse biased

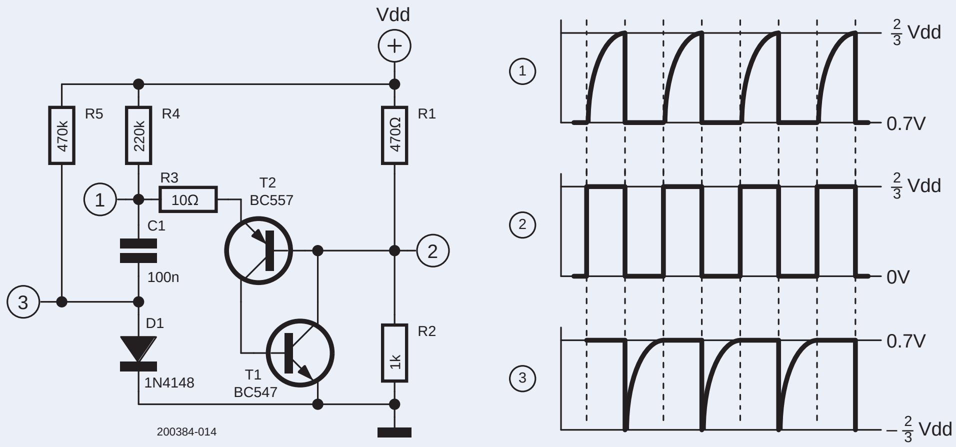


Figure 8: A simple squarewave oscillator.

and no longer conducts, and C1 will discharge through R5. Due to the negative voltage at node 3, the voltage over R5 is about twice as high as the voltage over R4. Therefore, its value is about twice the value of R4 to ensure equal charge and discharge rates. The result is a symmetrical waveform at node 2. R3 has no function anymore and can be omitted.

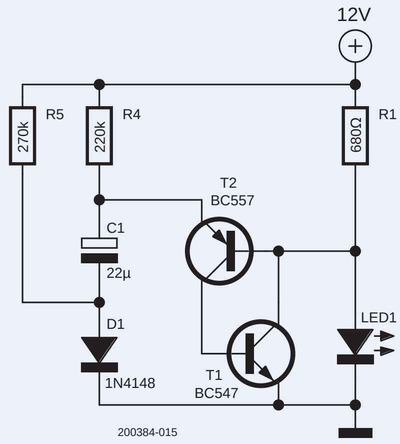


Figure 9: A 1-Hz LED blinker.

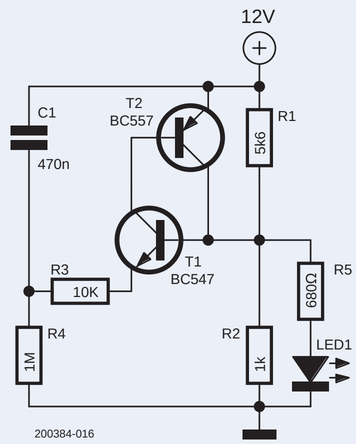


Figure 10: 'Upside-down' 1-Hz LED flasher.

LED Blinkers and Flashers

If, in the circuit of Figure 8, R2 is replaced by an LED, then we obtain an LED blinker (**Figure 9**). With the given component values, the LED will blink at approximately 1 Hz. Since R3 had become useless, it was removed from the circuit.

Figure 10 shows an 'upside-down' circuit to flash an LED at a rate of around 1 Hz. R3 determines the pulse width and therefore the duration of the flash. With the given values, the pulse width is about 11 ms.

Similar to Figure 10, but with the LED connected in a different place, current limiting resistor R5 becomes superfluous and can be removed (**Figure 11**). Its role has been taken over by R2 which is in series with the LED when the PUT is conducting.

That's All, Folks!

We will stop here even though there is much more to say about PUTs and their applications. The circuits presented in this article showed some basic applications and explained how things worked. The intention was not to be exhaustive, but to inspire you and at the same time teach some analog circuit analysis. More circuits based on the PUT can for instance be found at [1] and [2].

200384-01

Questions or Comments?

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

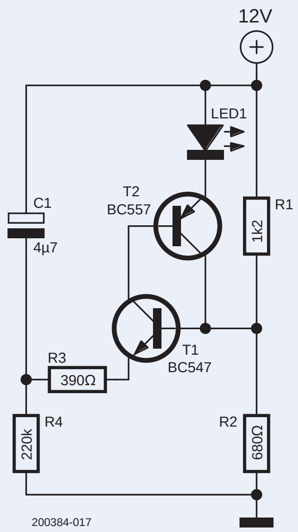


Figure 11: This 'optimized' LED flasher uses a larger capacitor but saves a resistor.

WEB LINKS

[1] More PUT Circuits at Elektor Labs: https://www.elektormagazine.com/labs/put-ting-it-all-together
[2] Roel Arits, "LED Dimmers (1)," Elektor 9/2018: https://www.elektormagazine.com/170404-02

Round Touchscreen for Raspberry Pi

HyperPixel 2.1 Round from Pimoroni

By Clemens Valens (Elektor)

Optimized for the Raspberry Pi Zero and Zero 2W, this round capacitive touchscreen with its 2.1" (53 mm) diameter is great for creating cool-looking devices for e.g. metering and home automation applications.

The HyperPixel 2.1 Round from Pimoroni [1] is a round 2.1" In-Plane Switching (IPS) capacitive touchscreen with high-speed Display Pixel Interface (DPI). Like its square and rectangular HyperPixel 4 brothers, the 2.1 Round is intended for Raspberry Pi. Actually, the size is optimized for the Raspberry Pi Zero and Zero 2W, but, as it has the standard 40-pin HAT connector, it can be plugged on any Raspberry Pi equipped with such a connector as long as you are careful about the mechanical side of things.

Specifications

The display's resolution is 480 by 480 pixels, but as it is round, you must, of course, subtract the corners. It has 18-bit color depth, meaning 262,144 colors, and supports up to 60 fps (frames per second). The viewing area has a 2.1" or 53.3 mm diameter and a viewing angle of 175°. Its full diameter is 72 mm (2.83") with a height of 11 mm (0.43").

Figure 1: The HyperPixel 2.1 Round showing the Raspberry Pi Buster desktop.

With a Raspberry Pi Zero attached, the total height (or depth, whatever you prefer) is 17 mm (0.67").

`Q: how many pixels are lost due to the rounded corners?*`

As the display uses almost every pin of the Hardware Attached on Top (HAT) connector, you cannot add other extension boards. However, the display does provide an alternate I^2C port for connecting things to.

The current consumption of a HyperPixel + Raspberry Pi Zero 2W system is about 300 mA in normal desktop operating mode. When booting I observed peaks of over 500 mA, so to be on the safe side you would need a 5 W, 5 V power supply.

Driver & Libraries

To use the HyperPixel 2.1 Round on a Raspberry Pi, you must install a driver first. Detailed instructions on how to do this are given on the Pimoroni website [1] and in their GitHub corner [2]. The drivers are for Raspberry Pi OS Buster only, but support for Bullseye is being worked on. Even though Bullseye is not yet supported, I did give it a try. My demo program (see below) worked the same, except that the menu of the desktop got overlaid at the top of the screen (**Figure 1**). I plugged the HyperPixel 2.1 Round on a Raspberry Pi Zero 2W running Buster and enabled SSH to make life a bit easier later on. After installing the driver and rebooting, the display showed a tiny version of the desktop, and I could open a terminal window. Now the HDMI port no longer works and the HyperPixel has become the only display. As it is rather tiny, I did most of the work over SSH on my Windows laptop computer.

To use the display in your own applications, you need Pimoroni's Python3 library [3]. It includes a few examples, but they didn't work well for me. After some searching, I found that upgrading the *pygame* library to the latest version (2.1.0 when I did this) solved the problems I was having. After that all the demos worked fine. Note that they feature touch, so touch the display to change the colors.

Touch needs a driver to make it work as a mouse on the desktop. Unfortunately, such a driver does not seem to exist yet, but you can use the library's 'deamon' example *uinput-touch.py* instead. That works quite well.

My First Application

Once I had the display up and running, it was time to see if I could do

`*A: 480 × 480 × (1 - 0.25 π) = 49,444 (i.e. almost 21.5%)`

Figure 2: The YouTube subscriber counter clock allows dragging of the counter value.

Figure 3: The HyperPixel 2.1 Round has the same diameter as the inside of a Pringles can.

something with it. My idea was to use it for my YouTube subscriber counter [4] to replace the 7-segment display by something sexier. The clock demo seemed like a good starting point. All I had to do was add the YouTube API query part to get the subscriber count and then print the number on the screen.

First, I installed *httplib2* which is required for the YouTube subscriber counter snippet. Running the program now revealed that I also needed *libsdl2-ttf* to print text on the screen. Keep in mind that your application must call *pygame.init* if you want to work with text and fonts and that you must load a font. Loading a font is rather long on a Raspberry Pi Zero, so I put that at the beginning of the program. Using the font, you render the text into an image that you copy or 'blit' to the screen buffer to make it visible (**Figure 2**). Because the display has touch capability, I added a feature allowing you to position the subscriber count anywhere on the screen simply by dragging it. Like the clock, its color depends on where you touch the screen. To make it stand out, it will not use the same color as the clock unless you touch the center to make everything white.

My code can be found at ClemensAtElektor at GitHub [5].

Additional Remarks

A word about rotating the screen. I wanted the screen rotated in such a way that the Raspberry Pi Zero's USB and HDMI connectors point upwards. This corresponds to 180° or 'inverted.' The Pimoroni driver provides utilities to do this, but they didn't work for me. What did work, however, was simply adding the line `display_lcd_rotate=2` to the */boot/config.txt* file and reboot the system.

Note that for some reason the center of the screen buffer may not exactly be the center of the screen, it can be off in the vertical direction by several pixels. You can correct this by adding an offset where the sign of the offset depends on the rotation of the screen. You can see in my code how I handled that.

Alternate I²C Port

I gave the display's alternate I²C port only a quick try. According to the Pimoroni website, it would be I²C port 3, but I found it as port 11. You can find out what you have by issuing the command `i2cdetect -l` or by looking in the */dev* folder for files that start with *i2c*.

I connected a Mabee MPU6050 accelerometer module [6] to it to see if it would be found. The *i2cdetect* tool showed a device at address 68 hexadecimal, which is indeed the address printed on the module, so it seemed to work. I could also read the sensor after installing the *mpu6050-raspberrypi* library.

It Fits Exactly Inside a Pringles Can!

I happened to notice that the diameter of the HyperPixel 2.1 Round display is almost the same as a Pringles can, it fits exactly inside. Therefore, I decided to build my YouTube subscriber counter clock into an empty Pringles can.

I cut the top and bottom of a can and attached the display-with-Raspberry-Pi assembly to the bottom part with stand-offs so that the display came flush with the inside of the ring on the top part. Now the display is protected when the lid is on while touch keeps working (**Figure 3**).

A Nice Add-On

Summarizing, I think the HyperPixel 2.1 Round is a really nice add-on for a Raspberry Pi Zero 2W. It also works on a Zero. The image quality is very good, and touch works fine in your own applications. It would be a pretty cool addition to a home automation system.

Officially there is only support for Buster, but I did try it on Bullseye. My program worked the same, except that the menu of the desktop got overlaid at the top of the screen.

220358-01

RELATED PRODUCTS

> **HyperPixel 2.1 Round (SKU 19870)**
> www.elektor.com/19870

WEB LINKS

[1] Pimoroni website: https://shop.pimoroni.com/products/hyperpixel-round
[2] Pimoroni GitHub corner: https://github.com/pimoroni/hyperpixel2r
[3] Python3 library: https://github.com/pimoroni/hyperpixel2r-python
[4] YouTube subscriber counter: https://youtu.be/PuCCMZFNgQE
[5] ClemensAtElektor at GitHub: https://github.com/ClemensAtElektor/HyperPixel2r
[6] Mabee MPU6050 accelerometer module: www.makerfabs.com/mabee-6aixs-imu-mpu6050.html

Remote Sensing with Connection Loss Detection

Using nRF24L01+ Modules

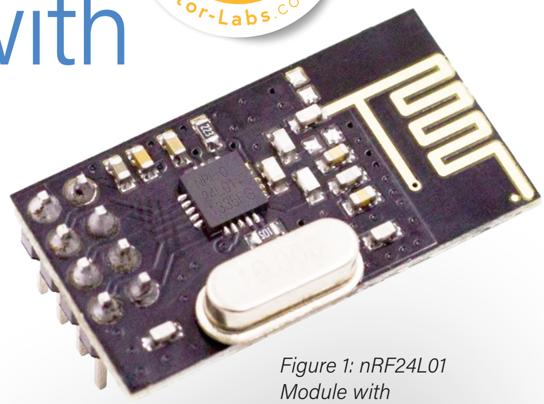


Figure 1: nRF24L01 Module with PCB antenna.

By Roel Arits (The Netherlands)

Wireless data transmission enables many applications (e.g., in the field of home automation). However, the communication must be robust and reliable. This PIR sensor project is a basic example of how to detect loss of connection with two-way communication modules.

The nRF24L01+ RF modules (**Figure 1**) are interesting, low-cost solutions for wireless two-way communication [1]. The modules have a serial peripheral interface (SPI) bus that enables configuration and control via a microcontroller. Many examples of projects using Arduino boards in combination with these RF modules can be found on the Internet.

With the project described here, I want to add an extra feature that uses the two-way communication to detect the loss of communication between the *transmitter* and the *receiver*. It's a bit odd to talk about a transmitter and a receiver when using two-way communication, since modules act as a transmitter *and* as a receiver at the same time. But for clarity we label one module as the transmitter because its main task is to transmit the state of a (PIR) sensor to another module, which receives the data for further processing.

The ability to detect the loss of communication is very useful for remote sensors, because when there is no communication, data can get lost without noticing. It is also useful when installing the sensor to check if both nRF24L01 RF modules actually "see" each other and are not out-of-range.

In **Figure 2** you can see an overview of the project and how both nRF24L01 modules communicate with each other — and how the detection of loss of communication is implemented.

Circuit Diagrams

Let us begin with the schematic of the transmitter (**Figure 3**). The nRF24L01 needs to be powered with 3.3 V, but the I/O pins of the nRF24L01 are 5-V tolerant. So, connecting the nRF24L01 SPI bus directly to a 5-V Arduino Pro Mini is no problem.

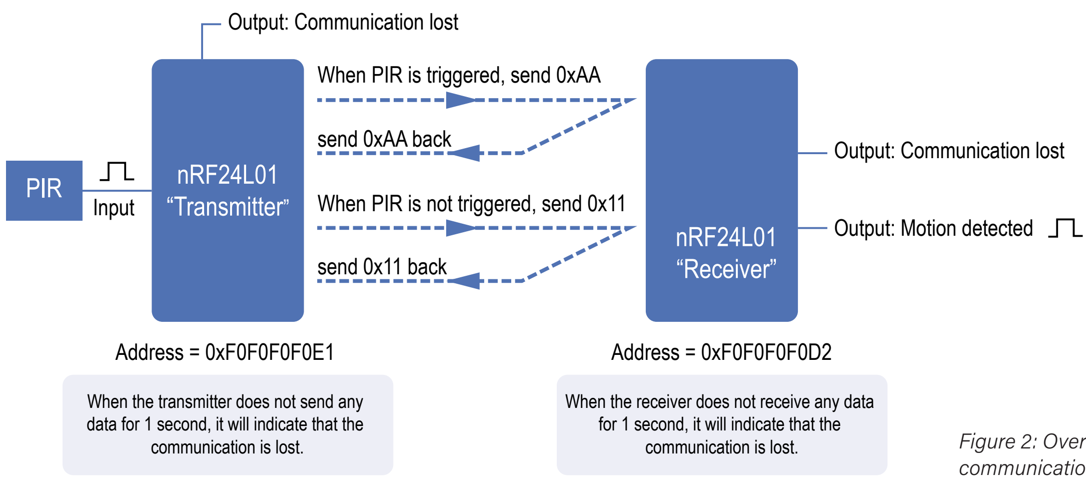


Figure 2: Overview of communication steps.

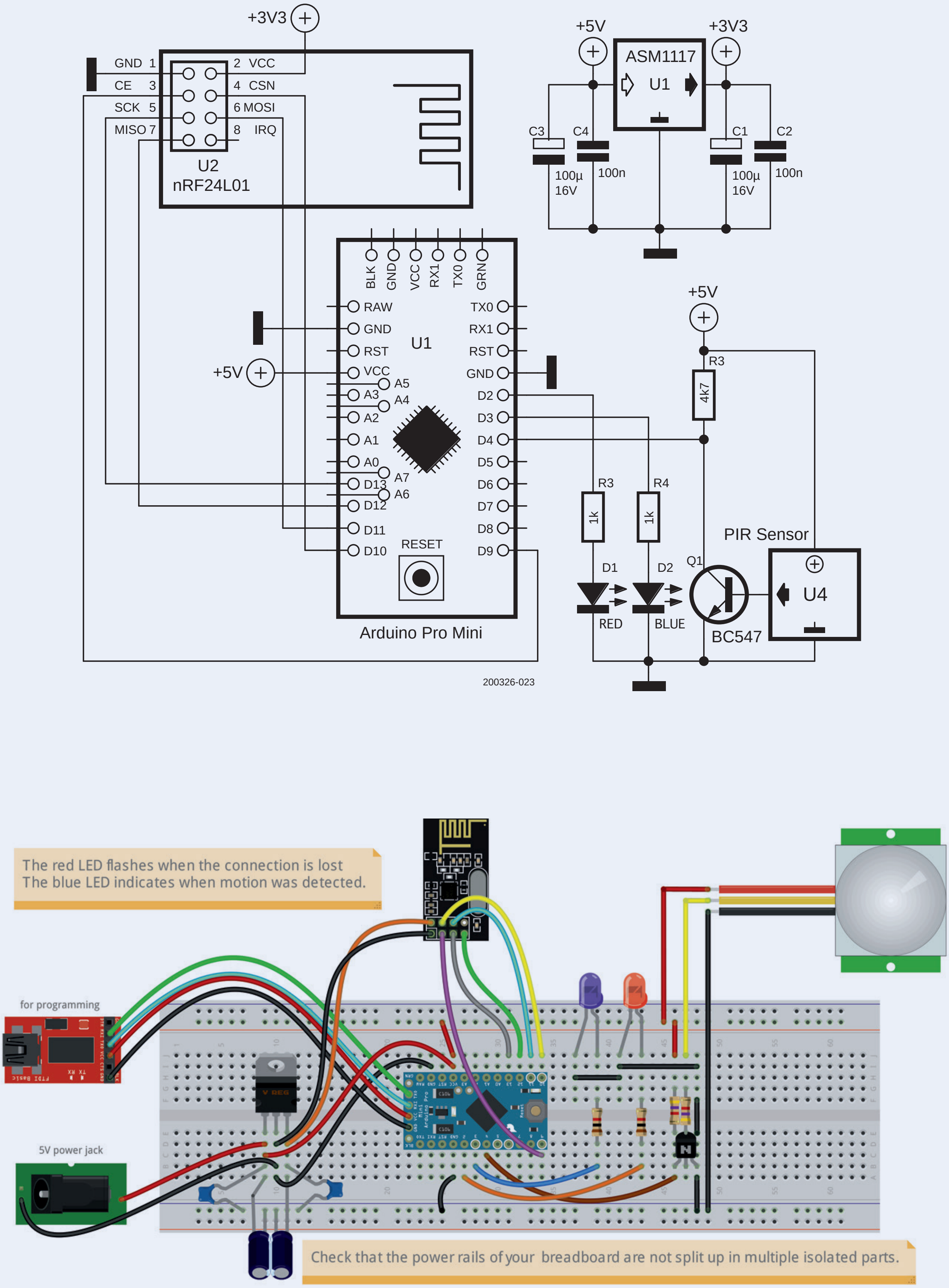


Figure 3: Schematic for the transmitter part.

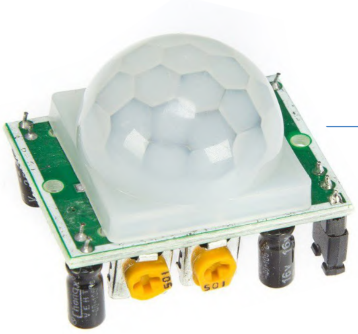


Figure 4: PIR sensor module.

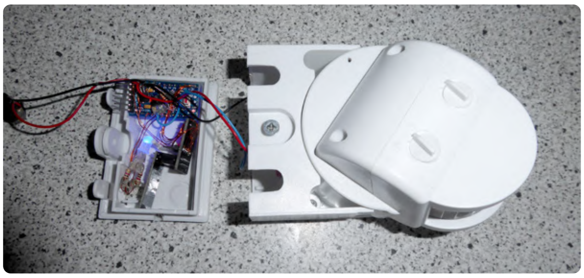


Figure 5: Transmitter with PIR sensor.

Figure 6: Receiver unit.

The power supply bypass capacitors should be placed as close as possible to the Arduino and the nRF24L01 module, respectively, to suppress all the switching noise coming from these chips. This is often overlooked in Arduino projects and can cause all kind of unexpected problems. It is also good practise to use multiple bypass capacitors in parallel with different values (e.g., C1 = 100 µF and C2 = 100 nF). Electrolytic capacitors are not as effective at higher frequencies as ceramic capacitors or polypropylene film capacitors. By putting different capacitor types in parallel, a more effective filter is created over a wider frequency range.

The PIR sensor (**Figure 4**) is connected to D4 of the Arduino. When the PIR sensor requires a different voltage than 5 V, this can be accommodated. Q1 acts as an (inverting) voltage level translator in case sensors are used that have, for example, a 3.3 V or lower output voltage. This way you can connect other types of sensors without changing the hardware. When the PIR sensor has a 5 V level output, Q1, R4 and R3 can be omitted and the PIR sensor output directly connected to pin 4 of the Arduino Pro Mini. When doing so, the Arduino sketch needs to be adapted, so the trigger input is high active instead of low active: `#define TRIGGER_ACTIVE_LEVEL 0 /* 0 = low active, 1 = high active */`.

The red LED (D1) will flash when the connection between the transmitter and the receiver is lost. When the connection is restored, the red LED will stop flashing and everything will work as normal again.

The blue LED (D2) indicates that the PIR sensor detects motion. This trigger event will be send over to the receiver as a trigger code byte. When the PIR sensor does not detect motion, then a live beat code will be send to the receiver. This way the receiver knows if there is a motion trigger or not.

The following code will show the livebeat code and trigger code defines:

```
#define LIVE_BEAT_CODE 0x11 /* code that is transmitted
as a live signal from transmitter to receiver */
#define TRIGGER_CODE 0xAA /* code that is transmitted
when transmitter detects activity at the trigger input */
```

Figure 5 shows the Transmitter unit with the PIR sensor.

Receiver

The receiver (**Figure 6**) will send the same code that it received from the transmitter back to the transmitter as an acknowledge. Because of this continuous communication between the transmitter and receiver, both can easily determine when the connection is lost.

The receiver circuit (**Figure 7**) is almost identical to the transmitter circuit. In the receiver, the red LED (D1) will start flashing when the connection is lost and the Arduino will send a tone to the speaker using the `tone (x, y)` command, with `x` = frequency and `y` = duration. The speaker is an 8 Ω version and is connected to the Arduino via an electrolytic capacitor. The higher the value of the capacitor, the higher the speaker volume will be. For a 50 Hz tone for 150 ms the following code is used:

```
tone(SPEAKER_PIN, 50, 150); /* output a tone of 50Hz for
a duration of 150ms on the speaker output */
```

I've chosen for a low-frequency tone because that is less annoying/disrupting than higher tones. The blue LED (D2) indicates when motion is detected. You can connect a relay to this output via a transistor to power on/control any other device when motion is detected.

200326-01

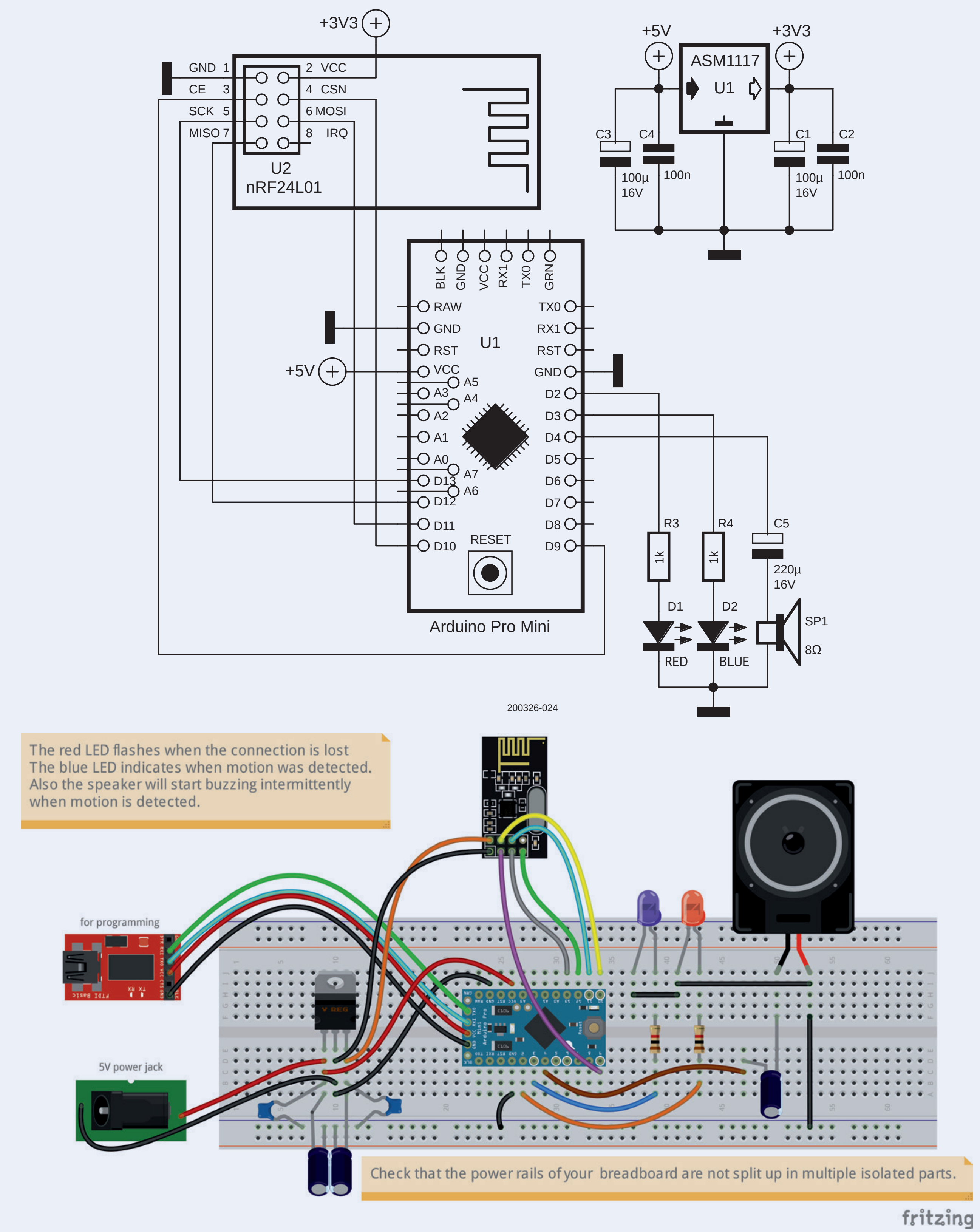


Figure 7: Schematic for the receiver.

WEB LINKS

[1] RF Modules Datasheet: www.nordicsemi.com/Products/nRF24-series
[2] This project on Elektor Labs: www.elektormagazine.com/remote-nrf24l01

Digital FM Receiver with Arduino and TEA5767

Stayed Tuned with an Arduino Nano

By Hesam Moshiri (Iran)

To build an FM receiver, developers can use highly integrated chips, such as the TEA5767 from NXP. Here we describe an FM receiver with a frequency range of 76 to 108 MHz and 2 x 3 W stereo sound output. An Arduino Nano reads three push buttons for frequency tuning and diplays the FM station data on an LCD.

FM receivers are among the top popular circuits of any electronic enthusiast. In this article, I will introduce a complete digital FM receiver design that is equipped with an LCD screen and three push-buttons. It can search for FM signals in the range of 76 to 108 MHz and can be tuned manually and automatically (Scan mode). The signal strength is also displayed as a bar graph on the LCD. The output sound is amplified by a high-quality, powerful 3 W + 3 W Class-D stereo amplifier. To control the receiver, I used the inexpensive and popular Arduino Nano board.

Hardware

Figure 1 shows the schematic diagram of the device. The circuit consists of three main parts: the FM receiver module, the audio amplifier, and the digital section with the Arduino Nano that controls the device. Add a telescope antenna, a power supply, and two 8-Ω speakers to build this simple digital FM receiver.

Receiving the Radio Signal

The FM receiver module is based on the TEA5767 [1]; it is a well-known integrated circuit that can be controlled via the I²C bus. It covers the FM frequency range from 76 to 108 MHz and outputs L and R stereo audio channels that must be amplified. Even with an earphone, the audio signal level from the TEA5767 is too low. Frequency tuning and signal strength measurement are performed by the Arduino Nano code.

A low-pass RC filter (R4, C7, C8, and C9) reduces power supply noise. R5 and R6 are mandatory pull-up resistors for the I²C bus, and CON1 is a UFL connector that provides an antenna connection. **Figure 2** shows the TEA5767 module.

The Audio Amplifier

The PAM8403 [2] is a 3 W + 3 W HiFi Class-D amplifier that can operate with only a single 5-V supply. The maximum output power can be achieved using 4-Ω speakers; however, for this project, 8-Ω speakers are recommended to limit power dissipation in the voltage regulator (IC2). According to the datasheet, "The PAM8403 is a 3 W, class-D audio amplifier. It offers low THD+N, allowing it to achieve high-quality sound reproduction. The new filter-less architecture allows the device to drive the speaker directly, requiring no low-pass output filters, thus saving the system costs and PCB area."

C1, C2, and C3 are used for noise decoupling on the power supply pins, R2, R3, C4, and C5 are used to transfer the output audio to the amplifier. Also, they form high-pass RC filters to remove low-frequency noise. **Figure 3** shows the reference circuit of the PAM8403 integrated circuit. P1 and P2 are right-angle two-pin XH connectors that are used to connect the speakers to the board. POT1 controls the output sound level.

Arduino in Control

AR1, an Arduino Nano board (see **Figure 4**), is used to control this digital FM receiver. It drives a standard 8 x 2 character alphanumeric LCD (LCD1). It reads and reacts to the push buttons (SW1, SW2, and SW3), and sends/receives the TEA5767 data via the I²C bus. R1 sets the contrast level of the LCD and C10, C11, and C13 are used for switch debouncing.

Power Supply

The 7805 voltage regulator in D2PAK case [3] is the main component of the power supply that provides a stable +5 V supply for the circuit. C12, C14, and C15 are used for noise decoupling, and the integrated switch of POT1 (the stereo volume potentiometer) powers the device on and off.

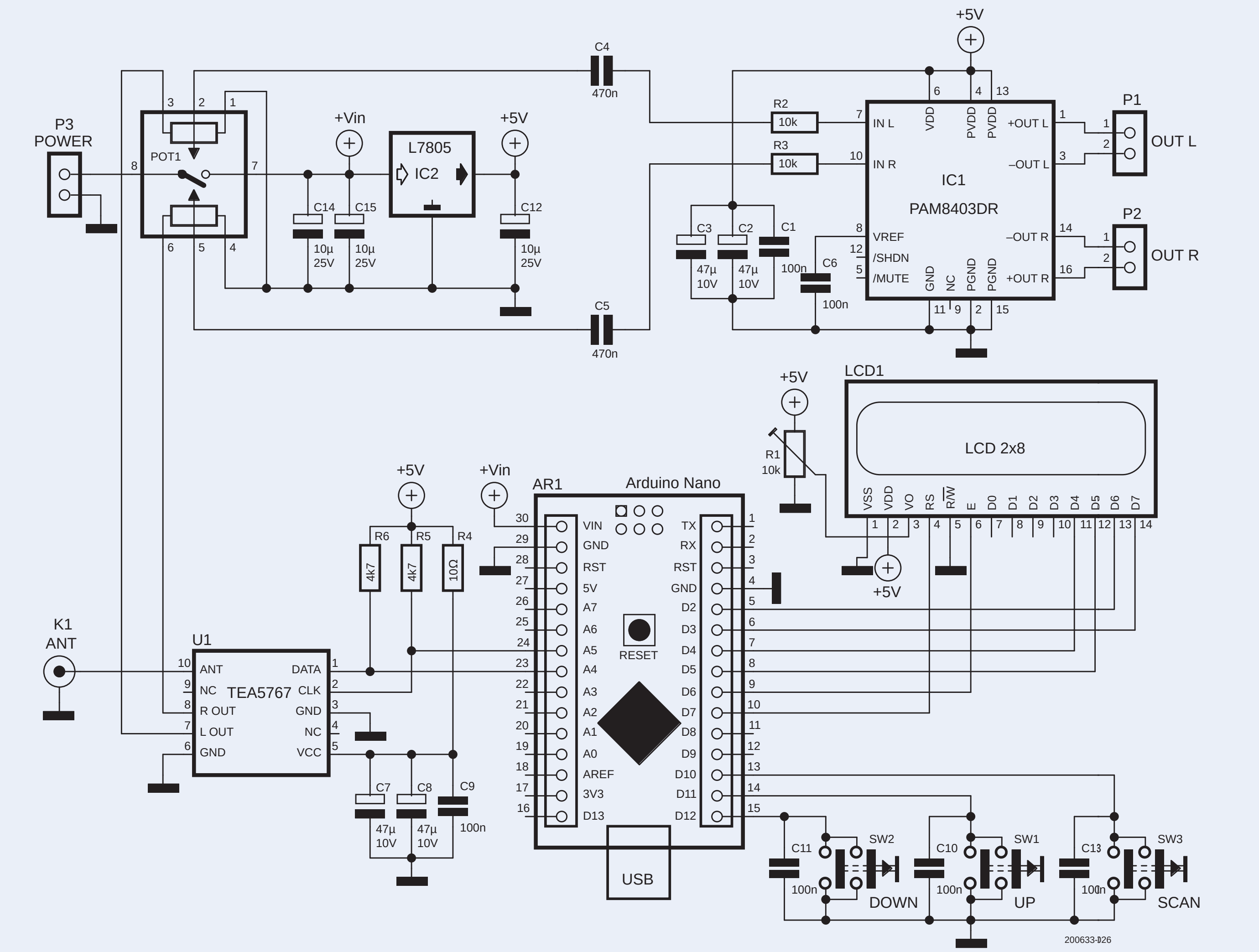


Figure 1: The schematic diagram of the FM receiver.

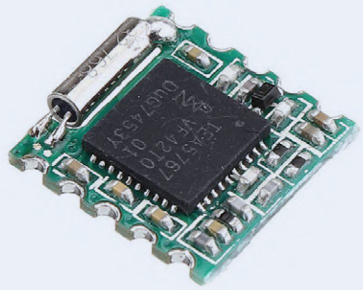


Figure 2: The TEA5767 FM receiver module.

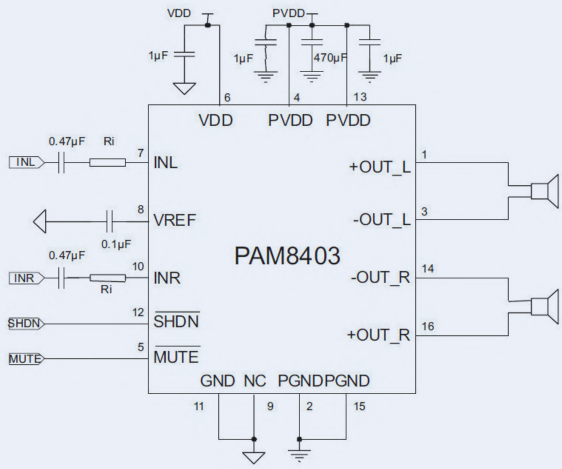


Figure 3: Reference design for the PAM8403 stereo amplifier. (Source: Diodes Incorporated)

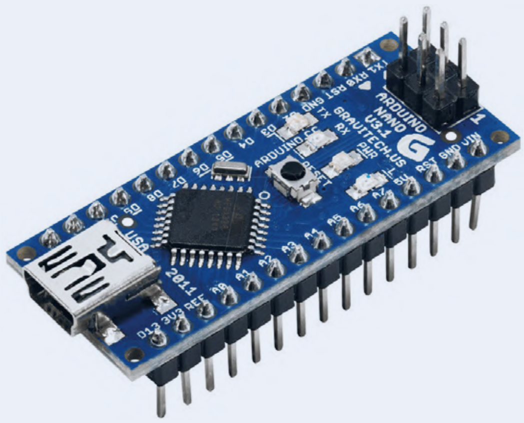


Figure 4: The Arduino Nano module.

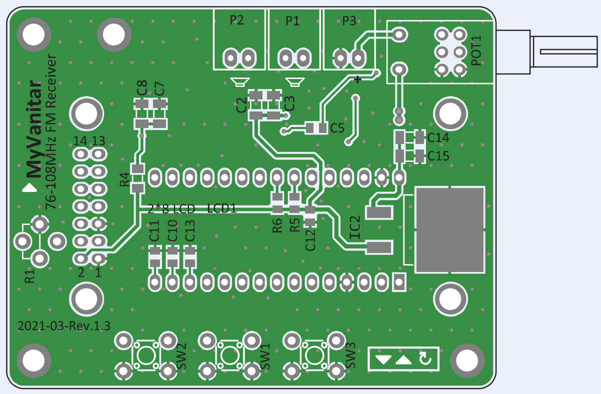


Figure 5: PCB layout of the receiver.

PCB Layout and Assembly

Figure 5 shows the PCB layout of the digital FM receiver. You can download the PCB's Gerber and drill files of the author from the project's page on the Elektor Labs website [4]. You can order the board from your preferred supplier.

Even though most of the components on the PCB are SMD parts, soldering will not be too difficult with a small soldering iron and thin soldering wire. Start with the SMD components, take your time and check the solder joints of each component before continuing with the next one. Save the through-hole parts until last and pay close attention to which side of the PCB they should be soldered. The Arduino Nano board is mounted on the bottom side, and the LCD is on the top side of the board, preferably on socket strips. The correct orientation of the TEA5767 receiver is marked with a small rectangle on the PCB, depicting the crystal on the module.

Figure 6 shows the 3D views of the board, showing how this project is constructed. **Figure 7** shows the high-quality fabricated PCBs of the digital FM receiver circuit.

For reference, **Figure 8** shows the assembled PCB from the top, and **Figure 9** shows the bottom view of the assembled board. You will also need four 5-mm spacers to fix the LCD on the PCB board. You should use a UFL-to-SMA-F connector to connect your antenna to the board. **Figure 10** shows this type of connector. **Figure 11** shows the telescopic antenna that can be connected.

Please note that an early version of the PCB is shown in Figures 6 to 9, but there are only minor differences with the final layout!

Arduino Code

The Arduino sketch for this FM receiver (*FM_receiver.ino*) is available for download at this project's Elektor Labs page [4]. The library for the LCD (*LiquidCrystal*) and the library for the I²C-bus (*Wire*) are supplied with the Arduino IDE. However, the TEA5767 library must be downloaded from GitHub [5] and installed manually. Simply copy *TEA5767.CPP* and *TEA5767.H* into the folder where the sketch is saved. Just connect your Arduino Nano to the computer, and then compile and upload the code.

Operating the Receiver

The lower limit of the frequency is 76.0 MHz and the upper limit is 108.0 MHz. You can increase or decrease the frequency by 0.1 MHz by pressing the Up (SW1) and Down (SW2) buttons, respectively. Similarly, if you long-press these buttons, the frequency will be increased or decreased continuously. So it is pretty easy to tune the receiver to your desired frequency (FM station). Moreover, with the Scan button (SW3) the receiver can automatically find FM stations with sufficient signal strength and lock the receiver on the

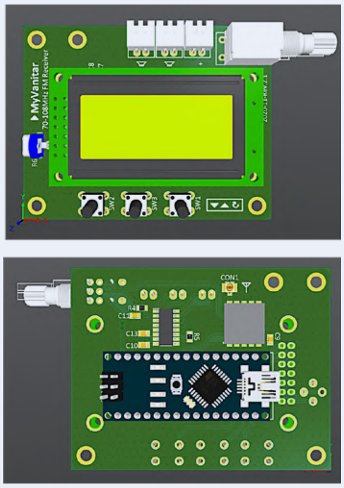


Figure 6: PCB, 3D-views.

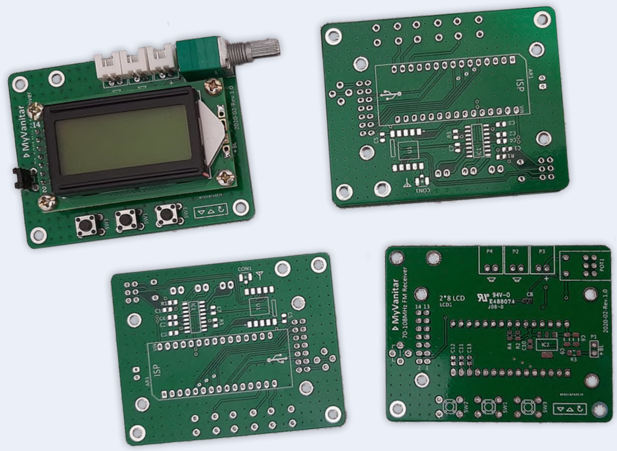


Figure 7: PCB, assembled top side and 3 x unpopulated.

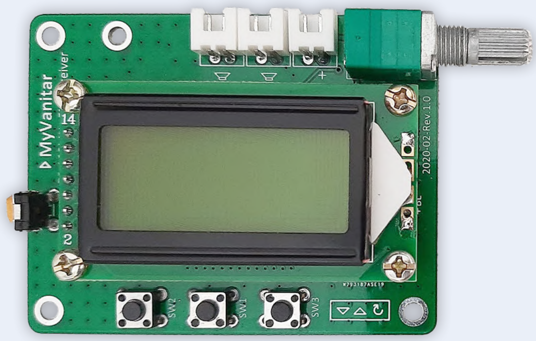


Figure 8: Assembled receiver, top side view.

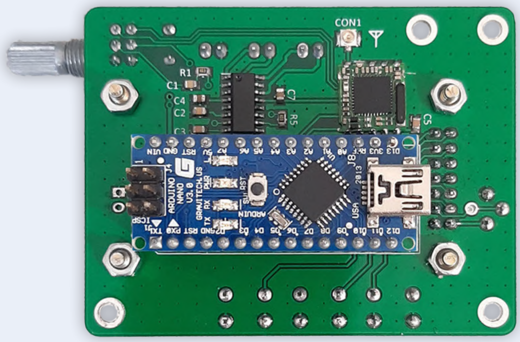


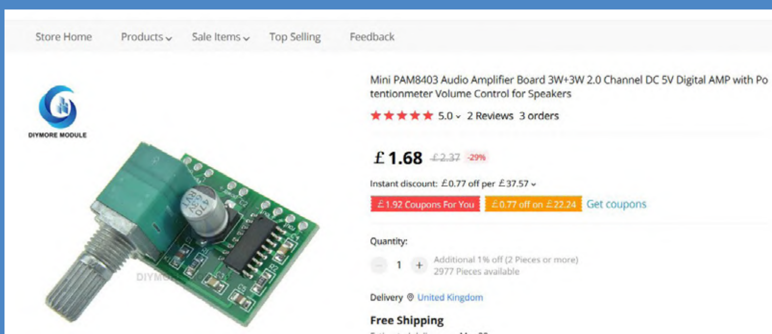
Figure 9: Assembled receiver, bottom side view.

Figure 10: Antenna connection cables with the small UFL and larger SMA connectors.

Figure 11: Telescopic antenna.

Finding POT1

In general, this combination of stereo volume control and switch is really hard to find, especially if it must match with the footprint on a PCB, like POT1 in this project. Apparently, this one is being sold as 'RV0971GS,' we're not sure about the brand. However, the author of this project found a dirt-cheap Far-East amplifier module, like shown in the picture below, that provides the PAM8403 we need for IC1 and the potentiometer. An online search on "PAM8403 volume control" will lead you to several suppliers selling a module as shown in the picture below. It contains, among others, the amplifier IC and a volume control that resembles (or is?) the RV0971GS, the latter should fit on the board designed by the author. We emphasise the word "should", because with such modules you can never be 100% sure what you will get when you order it. It goes without saying that you must desolder both components from the module first, a hot air soldering station and/or desoldering tools will come in handy for this task.



frequencies. To search the next station, press the Scan button again.

The FM signal strength is displayed on the LCD screen as a bar graph. To the right of this bar is an indication if the sound of the received station is either mono ("MN") or stereo ("ST"). In **Figure 12**, the receiver is tuned to a powerful mono FM station at 100.0 MHz frequency.

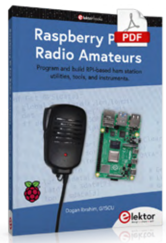
This project can also be found on the Elektor Labs website [4], where the software, the PCB design, and Gerber files associated with this FM receiver are available for download. There is also a video on YouTube that shows how this FM receiver works [6].

200633-01

RELATED PRODUCTS

- **Joy-IT Nano V3 (SKU 18615)**
  www.elektor.com/18615

- **D. Ibrahim, *Raspberry Pi for Radio Amateurs* (E-book) (SKU 19487)**
  www.elektor.com/19487



Contributors

Idea, Design, Text: **Hesam Moshiri**
Illustrations: **Hesam Moshiri, Patrick Wielders**
Editor: **Luc Lemmens**
Layout: **Giel Dols**

Questions or Comments?

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

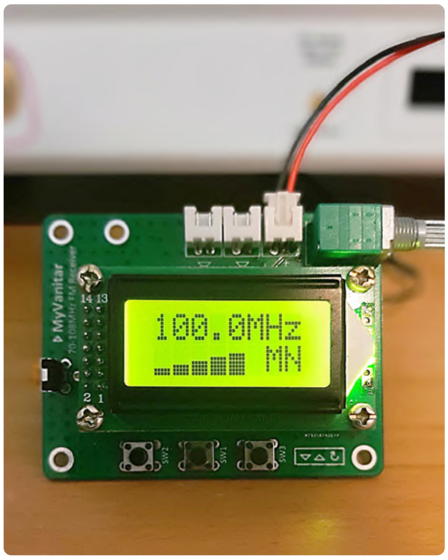


Figure 12: The receiver tuned to 100 MHz, a strong mono signal.

COMPONENT LIST

Resistors
R1 = 10 k trimmer, vertical style
R2, R3 = 10k, size 0805
R4 = 10 Ω, size 1206
R5, R6 = 4.7 kΩ, size 0805
POT1 = 50 kΩ stereo potentiometer + Switch (RV0971GS, see text)

Capacitors
C1, C6, C9, C10, C11, C13 = 100 nF, size 0805
C2, C3, C7, C8 = 47uF, size 1206
C4, C5 = 470 nF, size 0805
C12 = 10 uF, size 0805
C14, C15 = 10 uF, size 1206

Semiconductors
IC1 = 2 x 3 W audio amplifier PAM8403DR (SOIC-16)
IC2 = 5 V voltage regulator MC7805CD2TG (D2PAK-style)
U1 = FM Receiver module TEA5767

Miscellaneous
AR1 = Arduino Nano board
LCD1 = 2 x 8-character alphanumeric LCD
P1, P2, P3 = 2-way XH-connector, 2.54 mm pitch
SW1, SW2, SW3 = tactile push button 6 mm x 6 mm
CON1 = UFL connector SMD + telescopic antenna (see text)

WEB LINKS

[1] TEA5767 Datasheet: https://www.sparkfun.com/datasheets/Wireless/General/TEA5767.pdf
[2] PAM8403 Datasheet: https://www.mouser.com/datasheet/2/115/PAM8403-247318.pdf
[3] 7805 D2-PAK Datasheet: https://www.st.com/resource/en/datasheet/l78.pdf
[4] This project on Elektor Labs: https://www.elektormagazine.com/labs/digital-fm-receiver-with-arduino-and-tea5767
[5] Arduino TEA5767 library download: https://github.com/andykarpov/TEA5767
[6] The digital FM receiver on YouTube: https://youtu.be/qgci6huZ_-I

Changing an OLED Interface from SPI to I²C

By Luc Lemmens (Elektor)

Have a display configured for SPI? You could buy an I²C display, but why not do it yourself? Read on to learn how to change the display from SPI to I²C.

Some time ago, I needed a small graphic screen with I²C interface for a project I was working on in the Elektor Lab. The first possibly suitable display that I found on the (too full) lab table was a 0.96" 128 × 64 pixel OLED module that is available with an SPI or I2C interface. And mine was, of course, configured for SPI.

I figured that I could easily change the display from SPI to I²C by simply following the legend on the white silkscreen of the PCB: just move some resistors. The first attempt failed, and I wasn't sure if the display was still functional at all; this is sometimes uncertain with a part on the workbench that has already been used for another prototype. I decided not to waste more time and ordered a brand new one with I²C interface — problem solved. But recently, I made a mistake when I wanted to buy the same type of display and accidentally ordered an SPI version instead of an I²C. I tried again to change the configuration of the interface and this time succeeded, but not before I found some help on the Internet.

Affordable and Available

This type of OLED display is one of those generic modules that is very popular in the maker scene: affordable, readily available, and supported with numerous libraries and software examples. They are based on the SSD1306 controller and made by many manufacturers, but it's unlikely that you'll find any markings, or anything else that helps to identify the brand. I was lucky to find a datasheet/application note on the internet for a six-pin 0.96" OLED module that contained the information I needed to reconfigure the SPI interface to I²C. Funny enough, the manufacturer simply writes in the application note that this information came from a customer! Maybe a coincidence, but even the component references in the document matched with the display module I had on my workbench.

Figure 1 shows the back side of the display module. The first steps to change the display from SPI to I²C are indicated on the PCB: move R3 to position R1. R8 — an empty footprint on the SPI version — is mentioned too, and the document notes that it must be a 0-Ω resistor for the I²C interface. (Some solder or a jumper wire will work too.) The pin marked *DC* on the connector of the module determines the

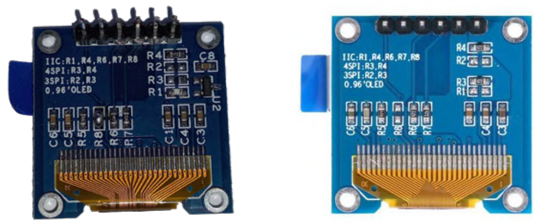


Figure 1: The original SPI-version, and the modified I²C next to it.

I²C address: connect it to GND for default address 0x3C and to VCC for 0x3D. The RST (reset) pin needs a passive power on reset circuit (i.e., a 10-kΩ resistor to VCC and a 100-nF capacitor to GND). The document also mentions a CS pin that is not present on the display I had; it is only found on seven-pin modules. If you have one of those, this pin must be connected to GND, according to the information I found, but I couldn't verify that.

Of course, it makes more sense to simply buy an I²C display instead. No additional components and connections are needed to connect it.

220005-01

Questions or Comments?

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

HomeLab Tours

A Hobby Does Not Retire

By Wim Stok (Netherlands)
and Eric Bogers (Elektor)

The EE series of kits from Philips were, for many, their first introduction to the world of electronics, and the beginning of a life-long hobby (and, in some cases, even an occupation). This was also the case for Wim Stok from Groningen.

Well, those electronics kits from Philips. Practically everyone from the older generation of (hobby-)electronics engineers will be familiar with them. And you can say what you like, but they were good and well thought out — including the clever spring contacts that made soldering unnecessary. True, after plenty of use, the reliability of the contacts did go down, but by that time you had either given up on the hobby or had bought your first soldering iron.

“On my eleventh birthday, I received the Electronic Engineer kit (EE20) from Philips as a present, and I used it to build the first radio of my own. For me this was the beginning of a fascinating hobby for which I now have, after my retirement, more time to spend on.”

“Later, I built amplifiers, with varying success, but now, in any case, soldered together. My projects were usually a mixture from descriptions in electronics magazines and Philips kit sets. During my time as a student, I discovered *Elektuur* (as it was still called then). When building the circuits described therein, in those adventurous times, their correct operation was not always guaranteed. A keeper however was the Equin amplifier from 1975/1976, with an output stage that contained so-called Quad-triplets. Built on a circuit board of my own design and combined with a preamplifier and a regulated power supply it still adorns my living room to this day. The outputs are connected to two ‘geluid uit de pijp’ loudspeakers — a design from 1984 by the magazine *Audio & Techniek*.” (The original article title translates from Dutch as ‘sound from the pipe’.)

For those who are interested: Fortunately, the English-language articles have been preserved; these can be downloaded in PDF format from [1][2]. We therefore can print the original schematic for the Equin amplifier here as ***Figure 1****.*

“I also built a cassette recorder on the frame of an old reel-to-reel recorder (**Figure 2**), with a reject Nakamichi combi-head and a flywheel that I was able to make on a professional lathe. This cassette recorder could be adjusted for three different type of tape. My audio system is completed with a FM tuner; the enclosure of which was unfortunately never quite finished (**Figure 3**).”

“During my professional life as a physiologist, electronics as a hobby was pushed into the background somewhat. I did however collect equipment and components that became available during the multiple reorganisations within the electronics department of the hospital I used to work at. For later, after I retired. For now, therefore.”

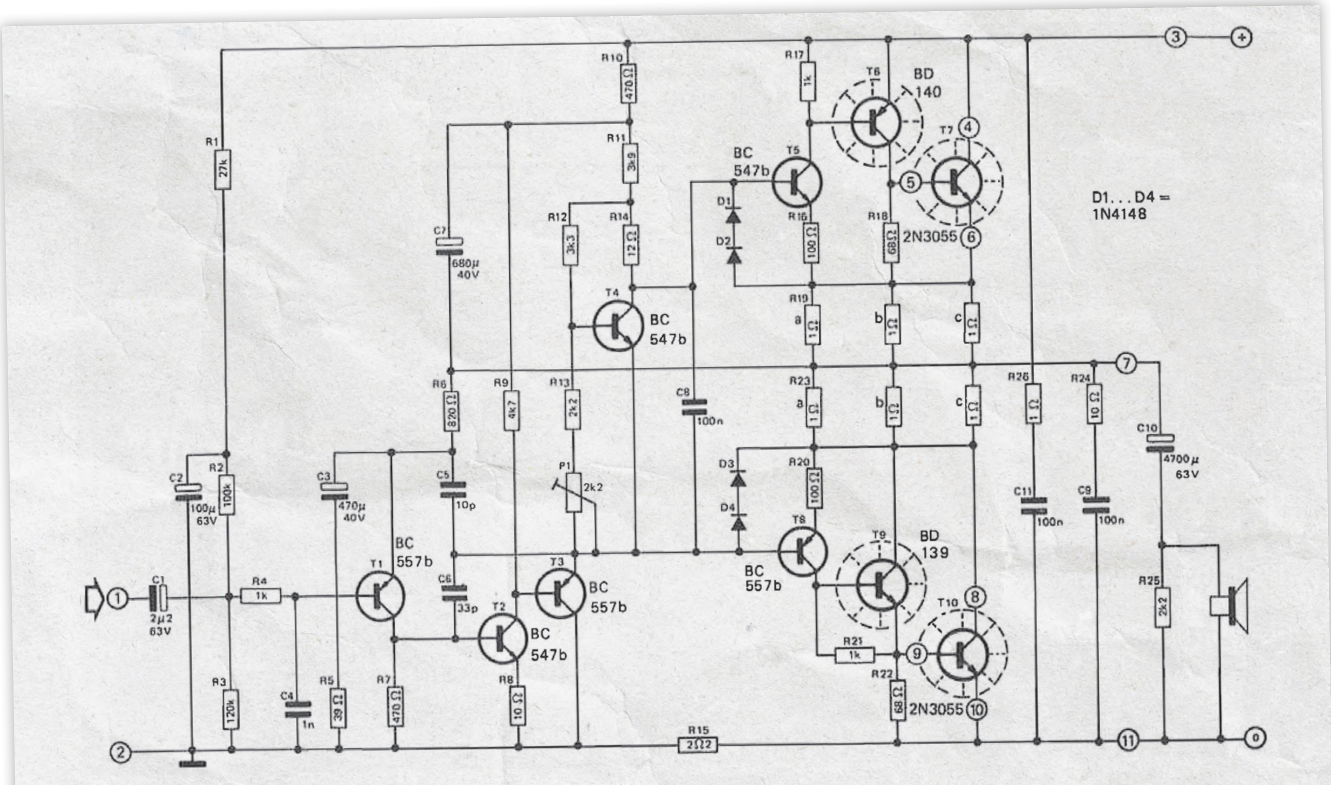


Figure 1: The original schematic for the Equin amplifier, with apologies for the average scan quality.

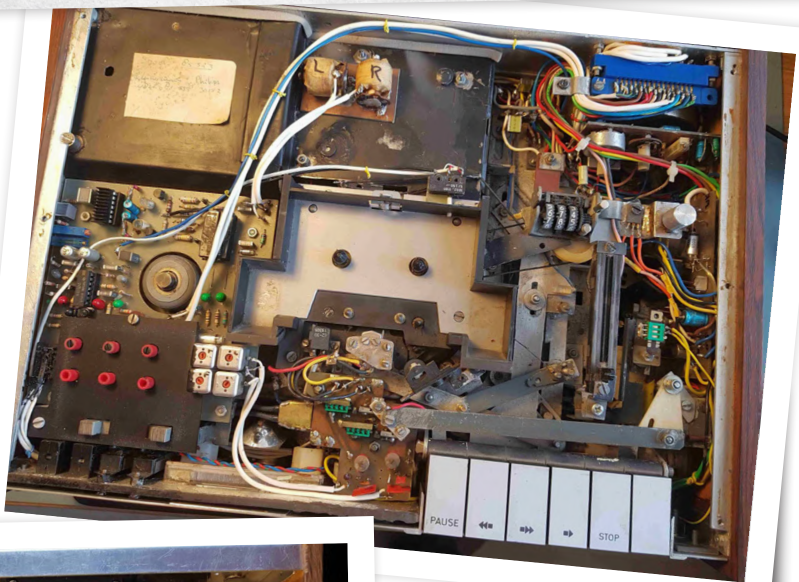


Figure 2: DIY cassette recorder.

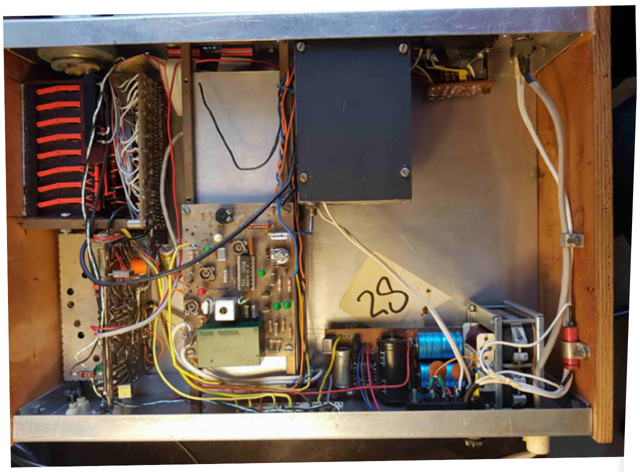


Figure 3a: DIY tuner on the inside...

Figure 3b: ...and from the outside — the enclosure hasn't been completed yet.

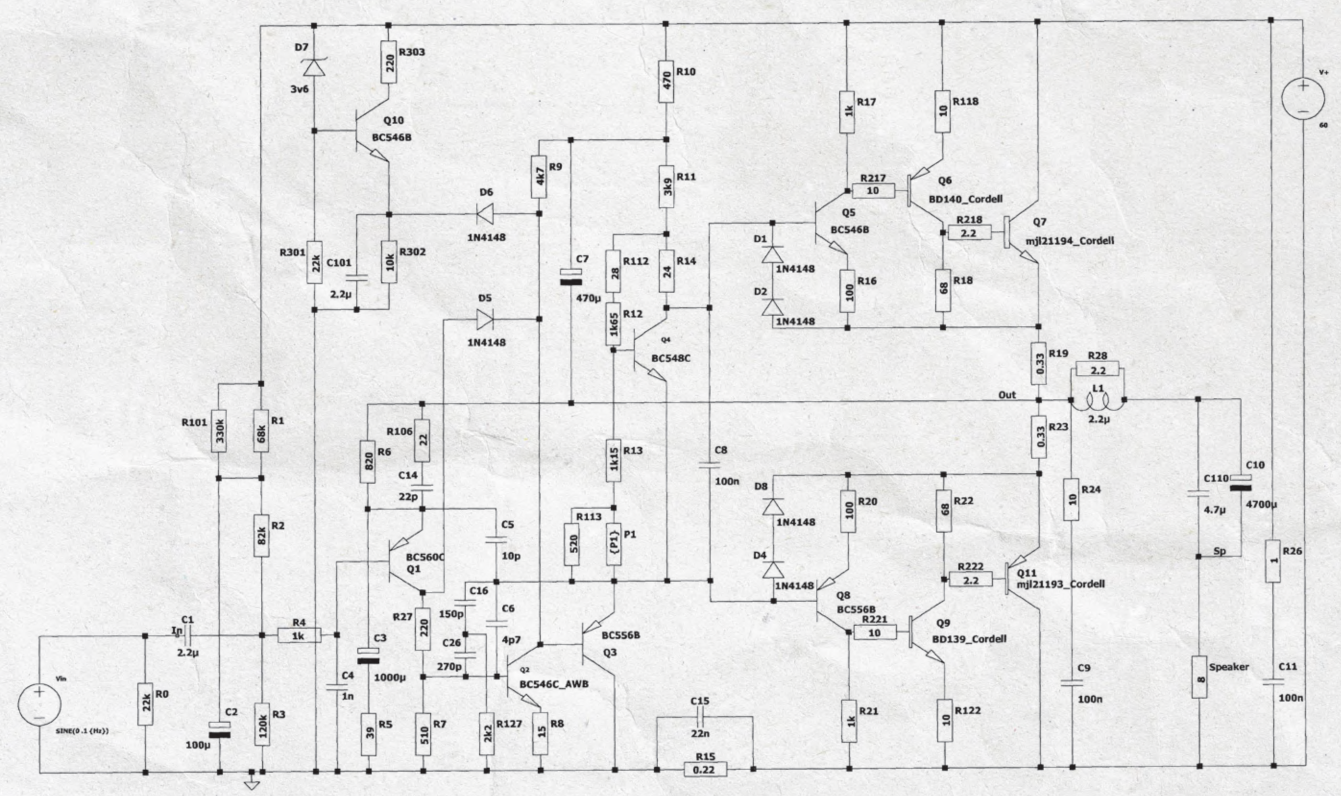


Figure 4: The schematic for the new Equin amplifier.

"I had available an enclosure with a toroidal transformer (2 x 55 V) and two fat electrolytic capacitors of 10 mF/100 V that turned out to be in an excellent state still. I therefore decided to build a 60-V version of the Equin amplifier, where the power supply for the two amplifier channels would be kept separate as much as possible. In contrast with the original Equin design, the power supply will now also be regulated."

The schematic of this new version of the Equin amplifier is printed in ***Figure 4****. As a current-feedback design the Equin has, by its very nature, already a large open-loop bandwidth. This is even further optimised here through a two-pole compensation network, instead of the more common single Miller capacitance. The final stage uses faster transistors (NJW21193/94). At an optimally adjusted bias, an LTSpice simulation of this Equin version at 1 kHz shows a distortion of less than 0.001%.* ***Figure 5*** *shows this new Equin in full glory.*

"My lab is relatively modest in size. **Figure 6** gives an impression, shortly after my move from Almere to Groningen. I don't own any advanced measuring equipment, so that I cannot measure the above-mentioned distortion of less than 0.001% in practice. But I do listen to it with much satisfaction, and that is what finally counts after all."

"Finally a tip to wrap up: I make my circuit boards myself, using the LaserJet/iron method. I print the layout on Hema photo paper and then melt the toner to the circuit board using a hot clothes iron. I subsequently remove the paper by soaking it in water. Now etch it and a perfect printed circuit board is the result!"

220056-01

Contributors

Text and photos: **Wim Stok**
Editor: **Eric Bogers**
Translation: **Arthur de Beun**
Layout: **Giel Dols**

Questions or Comments?

Do you have technical questions or comments prompted by this article? Send an e-mail to the editor of Elektor via editor@elektor.com.

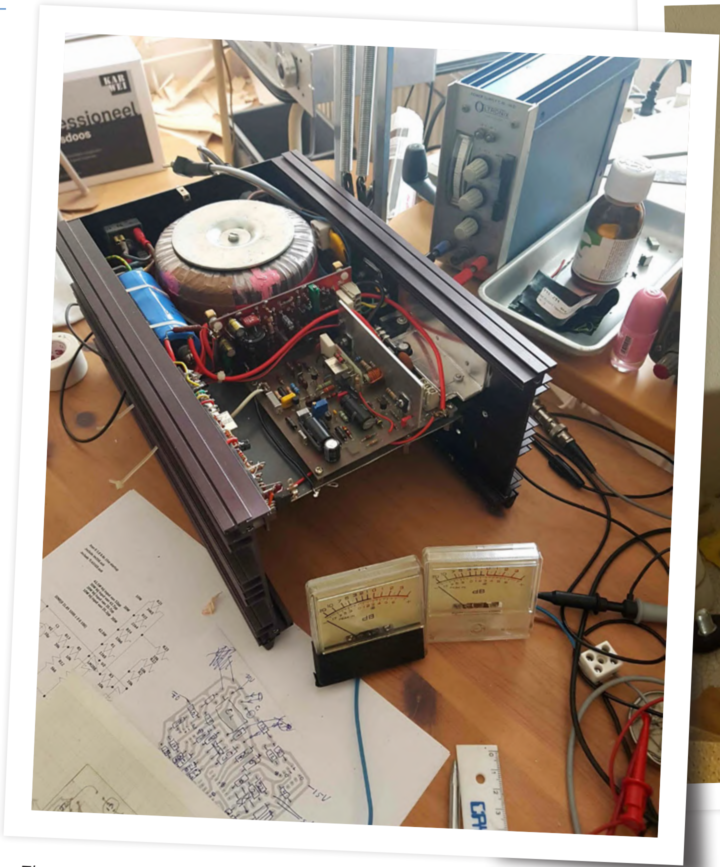


Figure 5a: Equin mk2 under construction...

Figure 5b: ...and in its enclosure.

Figure 6: My lab shortly after my move to Groningen.

WEB LINKS

[1] Equin (1), Elektor 4/1976: https://www.elektormagazine.com/magazine/elektor-197604/57609
[2] Equin (2), Elektor 5/1976: https://www.elektormagazine.com/magazine/elektor-197605/57619

elektor store

www.elektor.com

The Elektor Store

Never expensive, always surprising

The Elektor Store has developed from the community store for Elektor's own products like books, magazines, kits and modules, into a mature webshop that offers great value for surprising electronics. We offer the products that we ourselves are enthusiastic about or that we simply want to try out. If you have a nice suggestion, we are here (sale@elektor.com). Our main conditions: **never expensive, always surprising**!

Sensepeek 6003 PCBite Kit incl. 4x SQ10 Probe and Test Wires

Price: €164.95

Member Price: €148.46

www.elektor.com/20255

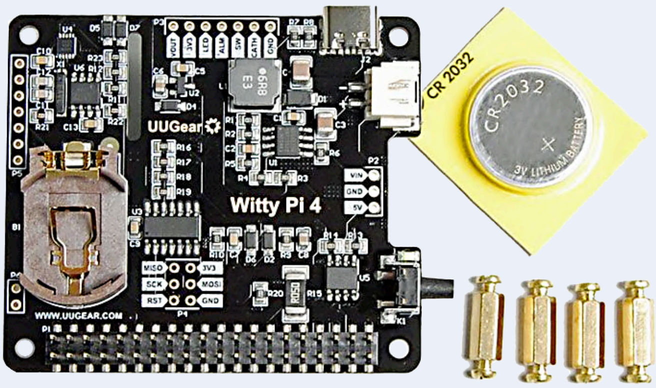
Pimoroni Weather HAT for Raspberry Pi

Price: €49.95

Member Price: €44.96

www.elektor.com/20235

Witty Pi 4 – RTC and Power Management for Raspberry Pi



Price: €39.95

Member Price: €35.96

www.elektor.com/20232

LDROBOT D300 LiDAR Kit – 360° Laser Range Scanner (12 m)

Price: €79.95

Member Price: €71.96

www.elektor.com/20222

Elektor US-Style Siren Kit

Price: €34.95

Special Price: €29.95

www.elektor.com/20218

Weller WLACCHHM-02 PCB Holder with 4 Magnetic Arms

Price: €79.95

Member Price: €71.96

www.elektor.com/20217

A Decade of Ethics in Electronics

Tessel Renzenbrink Reflects on the Digital Society and More

By Priscilla Haring-Kuipers
(The Netherlands)

Tessel Renzenbrink has written for *Elektor* about new technology, the Internet and its issues. After a decade, she took a position at the Netwerk Democratie foundation. Together we look back on these 10 years and talk about what she is doing now.

Haring-Kuipers: How did it start?

Renzenbrink: It started with a tiny advertisement in a newspaper that a friend pointed out to me: they were looking for a web editor to take over the Tech the Future website and write about the "impact of technology on society." At first, I wrote about everything — from new materials to biotech — but along the way, I realized that I found the Internet itself was the most interesting. Like a lot of people, I saw its potential to foster enormous change, especially in the distribution of power.

Writing for Tech the Future was one of my first assignments as a freelance writer, and

I wasn't good at writing quickly. I wanted to dive into a topic, and Elektor has always given me a lot of freedom to follow my interests. The Internet has an enormous amount of influence on everything and everyone, including electronics. At first Tech the Future was a separate entity, then it was part of the weekly Elektor newsletter and finally Elektor CEO Don Akkermans turned it into a part of the monthly magazine as Elektor Ethics. I think Don was among the first to think more about the consequences of technology, a topic that is getting much more attention now. And Elektor continues to raise awareness about the wider impact of electronics. In 2021, Elektor launched the World Ethical Electronics Forum (WEEF) in collaboration with ELEKTRONIK PRAXIS [1].

Haring-Kuipers: What were some big moments in the past decade?

Renzenbrink: One event I remember vividly were the SOPA PIPA protests in 2012. The United States Congress proposed two laws, the Stop Online Privacy Act and the Protect IP Act, which would cripple the free flow of information on the Internet. What was fascinating is that there was enormous resistance online, but in the offline world and media, it wasn't mentioned at all. Until January 18 when thousands of websites went black in protest — Tech the Future went dark too. The Internet Blackout was in the news everywhere, and suddenly lawmakers paid attention and changed their voted to No. The legislation was off the table.

Now I get nostalgic for a time when this seemed like our greatest problem. In 2013 whistle-blower Edward Snowden revealed that the Internet had become one large mass surveillance machine. Around the same time, we saw the emergence of the Internet of Things [2]. We were putting all these devices in people's houses that were full of privacy and security holes, importing the risks of the digital realm into the physical world. My writing changed after Snowden. From being optimistic about the possibilities of the Internet and its potential for redistributing power, I started focussing more on the risks.

The awareness in society about the negative impact of digital technologies grew after 2013, but I think the true awaking came after widespread reporting about the Cambridge Analytica scandal in 2018. That company used Facebook data to micro-target voters in an attempt to dissuade them from voting in the Brexit referendum and the 2016 US presidential elections. There is much more attention now to reduce the negative sides of digital technologies. The European Union is a frontrunner in trying to reign in big tech. The recently passed Digital Services Act and the Digital Markets Act are meant to give us at least some control over these gigantic companies.

(Source: Ehimetalor Akhere Unuabona, Upsplash)

Locally, I have also seen the change from tech-optimism into a more critical approach. In the past Amsterdam really wanted to be this "smart city" supported by many large tech companies that promised everything could be more sustainable, more efficient and more socially equal if you just implemented their technological solutions. Now there are several initiatives that promote smart citizens over smart cities, that emphasize citizen participation and the act of asking Amsterdammers what they want instead of starting with technical solutions and then looking for a problem to solve. Now Amsterdam wants to be a "digital city" which is based on very different ideas, and they are forming an alliance with cities like New York and Barcelona in the Cities Coalition for Digital Rights in an effort to join forces to oppose the influence of big tech companies on the local sphere [3].

Haring-Kuipers: What are you doing now?

Renzenbrink: A year and half ago, I became the co-director of the foundation Netwerk Democratie [4] that promotes a resilient democracy in a digital society. It has been really interesting to change from researching and writing into experimenting with practical solutions and doing things. Since its inception in 2011, the foundation has focused on using digital tools to support democracy. For instance: using digital means to give citizens more decision power or to make governments more transparent. In 2021 we added a focus on democratising technology. Digitization has had a great impact on society. But often the development of technology has been led by big tech and governments, without any democratic consultation. We advocate that it is both necessary and possible to apply democratic processes to collectively decide what our digital society should look like.

Recently, we hosted an Artificial Intelligence Working Lab. Here we looked at an algorithm that may be used on the streets of Amsterdam one day. We invited citizens to discuss the algorithm. What positive and negative sides do they see? Do they want to see it applied in the city? And if so, what safeguards would they want? For developers it is interesting to hear the citizens' perspective and for citizens it is important to have a say in the development of technology that may influence their life one day.

Now that I haven't done any writing for a while, I really miss it. I realize that writing was a really good way to keep yourself informed and to take the time to delve deeper into specific topics. There might be a writing project in the foundation soon.

220303-01

World Ethical Electronics Forum 2022

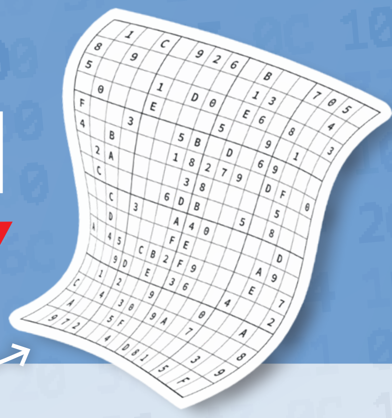
In November of 2021, Elektor launched the World Ethical Electronics Forum (WEEF) in Munich, Germany. The event inspired global innovators in electronics with an open discussion about ethics and sustainable development goals (SDGs). In addition to Elektor engineers and editors, the list of speakers and panelists included Dr. Stefan Heinemann (Professor of Business Ethics at the FOM University of Applied Sciences), Dr. Paula Palade (PhD, Jaguar Land Rover), Margot Cooijmans (Director, Philips Foundation), and several other thought leaders, including Priscilla Haring-Kuipers. Visit the WEEF webpage (https://www.elektormagazine.com/weef) to stay informed about Elektor's plans for WEEF 2022, which will take place in November of this year.

WEB LINKS

[1] Elektor, "World Ethical Electronics Forum (WEEF): A Focus on SDG, Not Just Profits," July 2021: www.elektormagazine.com/news/ethical-electronics-weef
[2] T. Renzenbrink, "The Internet of Things: Robots, RFID & Co-operation," Elektor, December 2012: www.elektormagazine.com/iot-coop
[3] Cities Coalition for Digital Rights: https://citiesfordigitalrights.org/
[4] Netwerk Democratie: https://netdem.nl/en/

Hexadoku

Puzzles with an Electronic Touch



Traditionally, the last page of *Elektor magazine* is reserved for our puzzle with an electronics slant: welcome to Hexadoku! Find the solution in the gray boxes, submit it to us by email, and you automatically enter the prize draw for one of five Elektor store vouchers.

The Hexadoku puzzle employs numbers in the hexadecimal range 0 through F. In the diagram composed of 16 × 16 boxes, enter numbers such that **all** hexadecimal numbers 0 through F (that's 0-9 and A-F) occur once only in each row, once in each column and in each of the 4×4 boxes (marked by the thicker black lines). A number of clues are given in the puzzle and these determine the start situation.

Correct entries received enter a prize draw. All you need to do is send us **the numbers in the gray boxes**.

SOLVE HEXADOKU AND WIN!

Correct solutions received from the entire Elektor readership automatically enter a prize draw for five Elektor store vouchers worth **€50.00 each**, which should encourage all Elektor readers to participate.

PARTICIPATE!

Ultimately October 17th, 2022, supply your name, street address and the solution (the numbers in the gray boxes) by email to: **hexadoku@elektor.com**

PRIZE WINNERS

The solution of Hexadoku in edition 7-8/2022 (July & August) is: **01D5C**.

Solutions submitted to us before August 15th were entered in a prize draw for 5 Elektor Store Vouchers.

The winners are posted at www.elektormagazine.com/hexadoku.

Congratulations everyone!

	3		0			A	1	E	9			6		5	
	D	1			C		0	6		F			7	8	
E	8			9							0			D	2
2		9		7		5			C		8		E		1
C	E		6	8		2	9	A	0		5	D		3	B
	1				B		5	D		7				F	
F		5			7					2			0		E
			B	E	D					C	6	7			
			1	A	E					6	D	C			
3		6			1					8			4		D
	A				0		C	3		1				2	
8	C		F	6		3	B	4	5		E	0		1	7
9		A		B		F			8		3		D		6
D	7			C							1			4	3
	F	B			A		3	C		9			2	7	
	5		3			4	2	7	6			B		C	

7	6	A	2	C	1	B	E	9	D	3	0	5	8	F	4
8	B	C	D	3	7	F	2	4	A	1	5	6	9	0	E
E	5	0	9	4	6	D	8	B	C	2	F	7	1	3	A
F	1	3	4	5	9	A	0	6	7	8	E	B	C	2	D
9	C	E	7	D	A	1	3	5	4	F	2	8	B	6	0
A	2	F	B	6	4	9	7	E	8	0	1	D	5	C	3
1	D	5	0	B	C	8	F	A	6	7	3	9	E	4	2
6	3	4	8	0	E	2	5	C	B	D	9	F	7	A	1
0	4	2	3	7	B	5	9	D	F	C	6	E	A	1	8
5	7	6	C	E	8	4	1	0	9	A	B	3	2	D	F
B	8	9	F	2	D	0	A	1	3	E	4	C	6	5	7
D	A	1	E	F	3	C	6	7	2	5	8	0	4	9	B
C	9	7	5	1	2	3	4	F	E	B	D	A	0	8	6
2	E	B	A	8	0	6	D	3	5	4	C	1	F	7	9
3	F	8	1	9	5	7	B	2	0	6	A	4	D	E	C
4	0	D	6	A	F	E	C	8	1	9	7	2	3	B	5

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.