DIGITAL BONUS EDITION

8+
Extra Circuits
& Projects

Circuit Special 2023
Design Ideas, Tips, and Tricks

I AM AI

CONTENTS

The **Elektor Circuit Special 2023** is available at newsstands and in the Elektor Store at elektor.com/circuit-special-2023.

The Elektor Circuit Special 2023 is out! You can find it on newsstands, it is available at Elektor.com, and it is in the hands of Elektor Green and Gold members around the world. With our "Circuit Specials," we continue the long tradition of bringing you project- and circuit-packed editions of your favorite magazine. This year, our editors and engineers have prepared extra Circuit Special content to share with you in this digital bonus edition. Curious what's inside this bonus issue?

Elektor community members have a wide variety of interests and backgrounds, so we've planned a little something for everyone: David Ashton presents a helpful "Ni-Cd Battery Rejuvenator;" Burkhard Kainka offers a simple solution for testing a capacitor's equivalent series resistance (ESR); Guy Weiler describes a DIY programmable fitness timer; Jan Buiting looks back at oscillographics with the Elektor "Spirator;" William Pretty covers a human interface for a DIY home alarm system; and more.

After you read through this bonus issue, I recommend that you dive into the complete Elektor Circuit Special 2023. As you progress through the articles and start projects of your own, I encourage you to share your innovations and ideas with your peers on the Elektor Labs online platform. Head over to elektormagazine.com/labs to start collaborating! If your projects catch the attention of our team, we just might contact you about featuring them in next year's Circuit Special!

Hob2Hood Interface Investigation

Talking to a Kitchen Cooker Hood

By Rob van Hest (the Netherlands)

Did you know that some cooktops and kitchen hoods can talk to each other? In this article, we will learn how to speak their language and then use it to control a deaf hood.
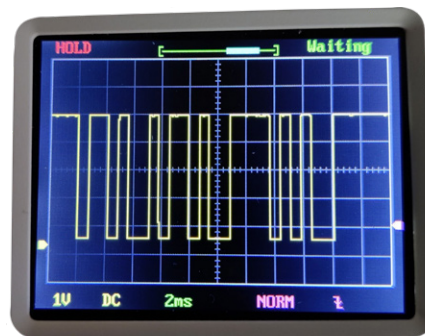
Figure 1: The bit stream for the Light On command.

After installing a new induction cooker top in our kitchen, I discovered that it was able to control a kitchen hood via the so-called Hob2Hood interface. I liked the idea of getting an automated hood without having to buy a new one. Searching on the internet, I learned that you could do this with an Arduino. However, I couldn't find anything about the communication protocol between the hob and the hood.

Reverse Engineering

What I did find was that the communication between the cooker top and the hood uses infrared light (IR) signals modulated at 38 kHz. As this is a standard IR remote control frequency, I connected an IR receiver module to an oscilloscope to see what the output looked like.

The cooker top can issue seven different hood commands: *Light On*, *Light Off*, and Speed 0 to 4. **Figure 1** shows the captured bit stream for the *Light On* command. Comparing the bit streams of all the commands showed that they all consisted of 24 data bits, preceded by a single start bit. The bit length was approximately 0.72 ms, which corresponds to about 1.4 kbit/s.

The control codes are listed in **Table 1**. Perhaps some or all of the bits should actually be inverted, but that makes little difference to the outcome.

Designing a Decoder Circuit

To build a receiver/decoder requires very few components. An IR receiver module, a small microcontroller and a few passive components is all you need. The schematic of the decoder can be found in **Figure 2**. At its heart is an 8-pin Microchip Technology PIC12F1822 microcontroller.

For the power supply, I used a universal mains power adapter that can deliver 400 mA at 5 VDC.

I ordered a 4-channel relay module online for controlling the hood. In my case, four channels were sufficient because my hood has only three speeds. Because each channel needs just 2 mA for control, each of them can be driven directly by the microcontroller. However, the control inputs use inverted logic (0 V = On; 5 V = Off).

Table 1: Captured Hob2Hood command codes.
Note how, for each bit stream, starting at the right, each byte is equal to the previous byte plus one. This is also true when starting at V0 and working your way up to the *Light On* command. If you remove V3, then this pattern is also valid for V4 up to V1.

Command	Bit Stream	Hex Code
Light On	1 00101101 00101100 00101011	2d 2c 2b
Light Off	1 00101010 00101001 00101000	2a 29 28
Speed 0 / V0	1 00100111 00100110 00100101	27 26 25
Speed 1 / V1	1 10010011 10010010 10010001	93 92 91
Speed 2 / V2	1 10010000 10001111 10001110	90 8f 8e
Speed 3 / V3	1 00011110 00011101 00011100	1e 1d 1c
Speed 4 / V4	1 10001101 10001100 10001011	8d 8c 8b

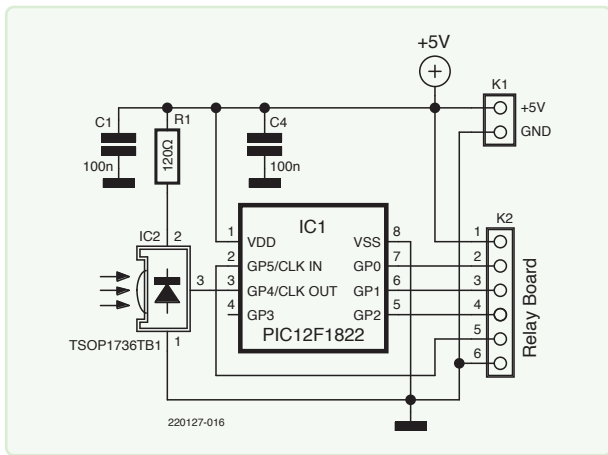Figure 2: A tiny microcontroller and an IR receiver are about all you need. The four-channel relay board connects to K2.
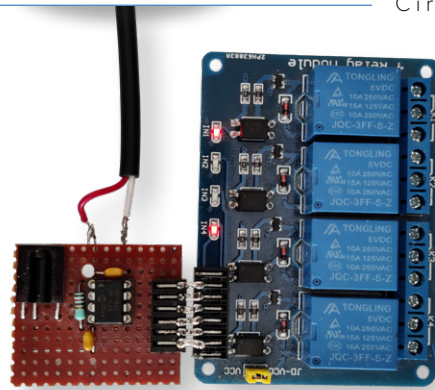

Figure 3: The receiver/decoder prototype connected to the relay board.

Because of the small number of components, the circuit can be built on a piece of prototyping board. See **Figure 3** for the test setup. A 6-pin header connects the decoder to the relay board.

The Software

The corresponding program can be downloaded from the project page on Elektor Labs [1]. Both the source and the compiled HEX file are available there. Also included are a little program to send Hob2Hood data and a receiver/decoder with LCD. Should you wish to modify the program, the B. Knudsen CC5X compiler is also required [2].

My cooker top issues four extraction speed commands (as well as *Off*), but my hood has only three. Therefore, in the program, I connected both *Speed 2* and *Speed 3* to Output 2 and *Speed 4* to Output 3. This can be modified in the program if necessary.
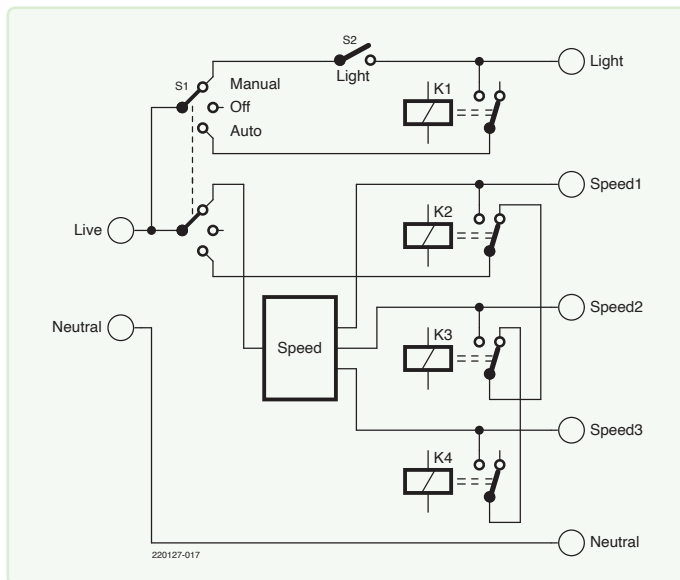
Figure 4: Here is how I connected the relays to the switches in my kitchen hood. It may be different for your hood.

Building Everything into the Hood

Because every hood is different, it isn't possible to provide detailed instructions that fit all. My hood has a 4-position slide switch for ventilation speed, an on/off switch for the light, and a power switch. The relays are simply connected to that, but I replaced the power switch with a 3-position double-pole switch to allow me to choose between *Manual*, *Off*, and *Automatic* modes (**Figure 4**). This may have to be solved differently for other types of hoods.

In the final setup, the relay board is built inside the hood, with the decoder board below it in a plastic box. A short 6-pin cable connects both boards.

While many plastic boxes are opaque to visible light, IR light tends to shine through these enclosures. Do give it a try before installing!

Also, beware of shadowing caused by pans and/or hands. On my cooker top, the IR transmitter is approximately in the middle of the control panel. The camera on your phone (which is sensitive to IR light) may help you find the transmitter on your cookertop. ◀

220127-01

Questions or Comments?

Do you have any technical questions or comments about this article? Contact the author at trainer99@ziggo.nl or the Elektor team at editor@elektor.com.

Related Products

> **Bert van Dam, *50 PIC Microcontroller Projects* (E-book)**
https://elektor.com/18091

> **Joy-IT Sensor Kit X40**
https://elektor.com/18124

WEB LINKS

[1] Project Files at Elektor Labs: https://elektormagazine.nl/labs/hob2hood-interface-investigation
[2] B. Knudsen Data, CC5X compiler: https://bknd.com/cc5x/
[3] AEG commercial: https://aeg.nl/taste/inspiration/mastery-range/mastery-products/hob2hood/
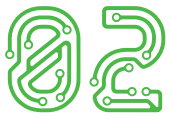
02 Combination Access Switch

Use a "tiny" Circuit for Access Control

By Antonello Della Pia (Italy)

This secure, AVR microcontroller-based Combination Access Switch allows you to control electrical loads or other circuits by entering a (secret) three-digit access code.
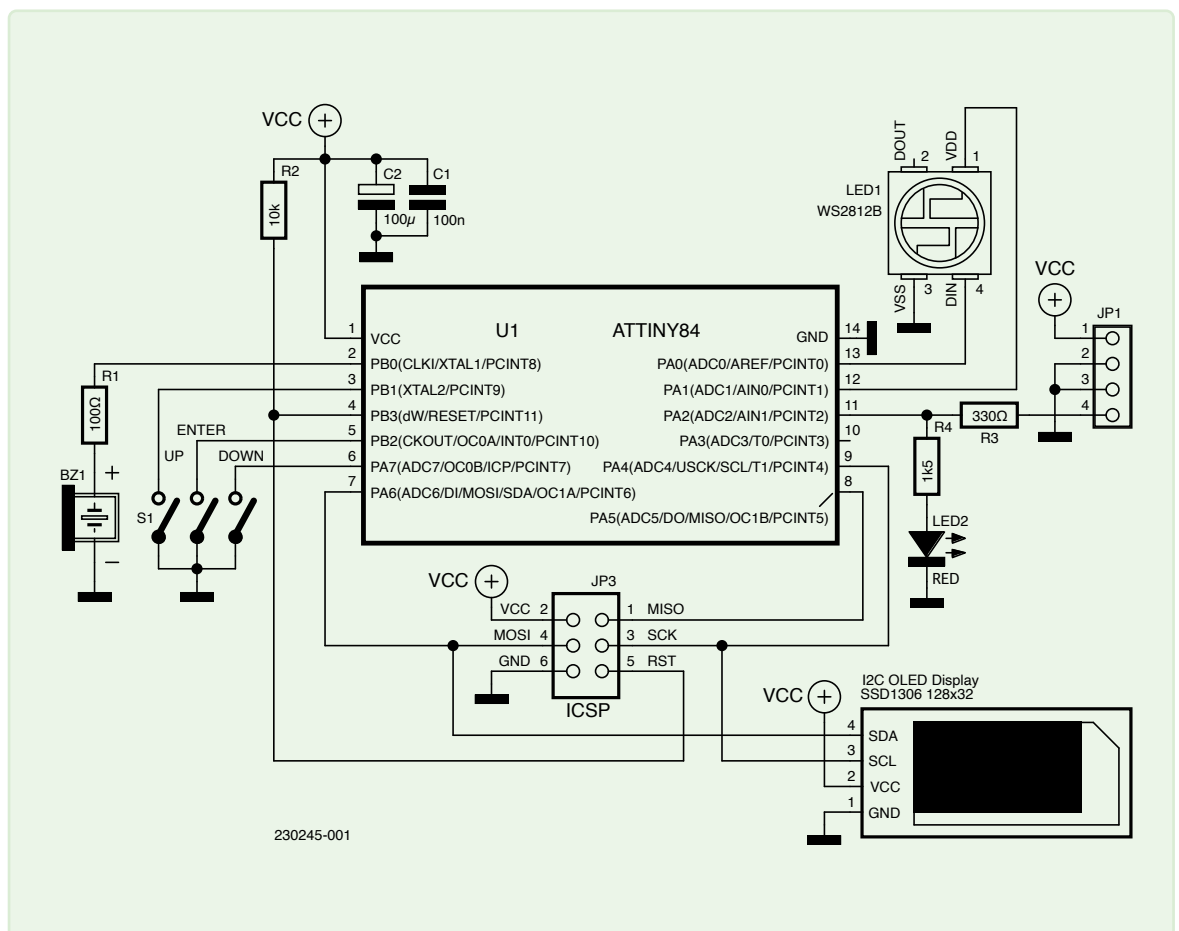
Figure 1: Combination Access Switch schematic.
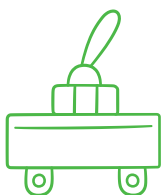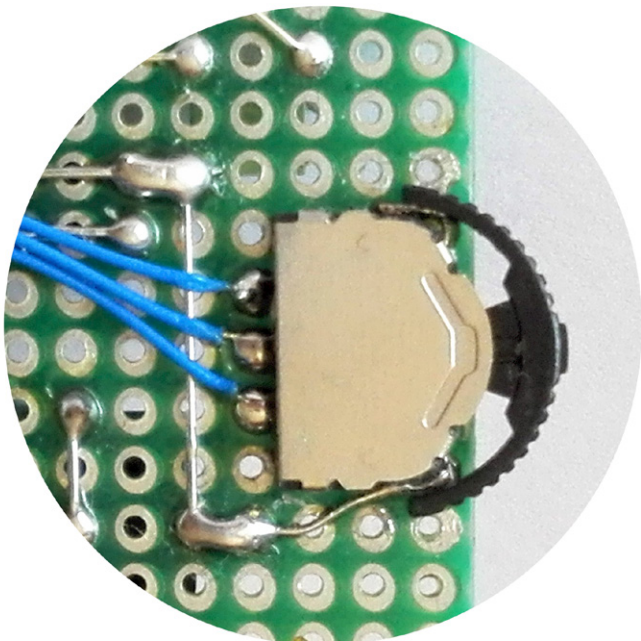
Figure 2: Rocker switch for navigation.

Figure 3: Combination Access Switch assembled on protoboard.

This circuit is intended to be integrated into a more complex design, where the activation of a device is obtained only if an exact numerical sequence, a three-digit access code, is entered. If this happens, a dedicated output of the Microchip Technology ATtiny84A microcontroller goes to a High level. This output can easily be connected to common electrical load control circuits, such as optoisolators, MOSFETs, relays, as well as to another digital or analog circuit.

Each digit can range from 0 to 9, allowing 10^3 combinations. If the code entry takes longer than 30 seconds, the unit will time out. The key code (obfuscated) is stored in EEPROM and can be changed by the user.

Circuit Diagram

The schematic (**Figure 1**) is quite simple, as the hard work is done by the firmware. The input device is a 3-way tactile navigation switch (**Figure 2**), while the user interface is a small, SSD1306-based 0.96″ I²C OLED display, plus a NeoPixel WS2812B 5050 LED and a buzzer, to delight the user with beeps and flashes during operation.

The circuit is always powered. The microcontroller, an ATtiny84A clocked at 8 MHz, normally stays in *Power Down* mode (with a current consumption of a few μA) and wakes up when the user presses the *Enter* button briefly. At this point, the operator can input the code digit by digit, using the *Up* and *Down* buttons, and confirm using the *Enter* button. During the process, the digits flash and the NeoPixel LED blinks red. See the construction in **Figure 3**.

After successfully unlocking the device, the *LOAD_POWER* output (PA2) goes high, the display shows "POWER ON" and the NeoPixel LED starts to flash blue every five seconds. If the user holds the same

button down for more than three seconds, the *LOAD_POWER* output goes Low, the display shows "POWER OFF," and the MCU returns to *Power Down* mode. Pressing and holding the *Enter* button at startup will put the unit into *Set* mode, allowing the user to change the key code. The NeoPixel LED then blinks green.

The 330 Ω resistors protect output PA2 from overcurrent, while output PA1 cuts off the supply to the NeoPixel LED when in *Power Down* mode, saving about 1 mA. The red LED, with R4, is an optional output status monitor. Capacitors C1 and C2 are the usual power supply bypass, while connector JP3 allows the MCU to be flashed using a USBasp programmer. Due to the low current consumption when active (approx. 5 mA), the circuit could also be used in battery-powered devices.

Software

The source code [1] was written and compiled using Arduino IDE 1.8.19 with *ATTinyCore 1.5.2* by Spence Konde installed. Thanks to the built-in *tinyNeoPixel* library, it was easy to drive a WS2812B LED, adding some nice visual effects. The *lcdgfx* library, which drives the display, must also be downloaded and installed.

In the sketch, I used some `#define` directives to create macros to set pin status at runtime, instead of using the usual Arduino `digitalWrite()` function. In addition, we use direct port manipulation to set the pins as Input/Output and set their initial states in the `setup()` function, instead of using the `pinMode()` and `digitalWrite()` functions. In this way, I was able to save a lot of memory, which I could use to add extra features. The size of the compiled sketch is 8106 bytes, so the code can fit into the ATtiny84A's 8192-byte flash memory. The firmware is broken down into several functions, mainly executed in `loop()`.
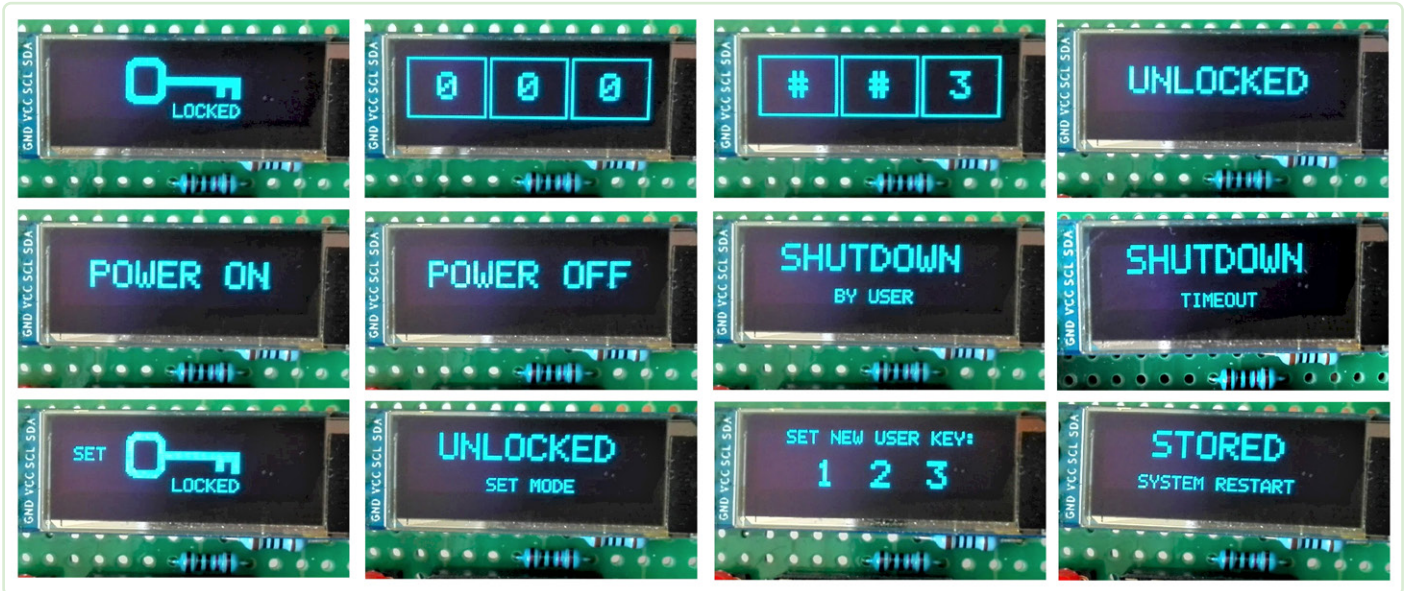
Figure 4: Screenshots from the user interface.

A host of other functions perform the following tasks:

> `getUpDownValue()` checks the buttons' statuses and manages the up/down digit counter
> `unlockDevice()` handles the device unlocking procedure
> `changeUserSettings()` allows the user to change the key code
> `powerOffHandling()` manages shutdown operations
> `printVcc()` shows the VCC value on request when pressing the *Down* button
> `blinkPowerOn()` blinks the blue NeoPixel LED slowly during power on status

These functions call other functions, as required, to manage data in the EEPROM, flash numbers and LEDs, reboot the MCU, play a tone, and manage power saving. The sketch is well-commented and includes further information and useful links. The provided file, *Combination_Switch.ino*, can, of course, be modified and recompiled. The Hex file is also available. But, since a picture is worth a thousand words, let's have a look at the most important screenshots, as shown on the device (**Figure 4**).

Some Notes

From a security point of view, it is important to remember that the device can be electrically bypassed, unless the circuit's enclosure is "armored." On the firmware side, with the same programmer used to load the sketch, you can easily read the Flash memory and the EEPROM. In this case, you get a Hex file, which is not as useful as the original source code, and another file containing the (obfuscated) access code values. To prevent this, AVR microcontrollers have the *Lock Bits Access* feature, which is well-documented by Microchip [2].

Setting both Lock Bits disables programming and verification of Flash and EEPROM. According to Microchip, "the only method to regain access to the memory, after setting the Lock Bits, is by erasing the entire chip with a 'Chip Erase' command." This means that it is still possible to reload a sketch into the MCU.

In this context, my favorite software tool for reading and writing Flash memory and EEPROM, and for setting Fuses on AVR microcontrollers is AVRDUDESS by Zak Kemble [3]. ◄

230245-01

Questions or Comments?

If you have any questions or comments regarding this article, you can contact the author at a.dellapia@elettronicaemaker.it or the Elektor team at editor@elektor.com.

🛒 Related Products

> **Elektor AVR Playground Bundle**
> https://elektor.com/19177

━ **WEB LINKS** ━

[1] Software at this article's web page: https://elektormagazine.com/230245-01
[2] Microchip: Lock Bits Access: https://bit.ly/3MNFf9O
[3] AVRDUDESS – A GUI for AVRDUDE: https://blog.zakkemble.net/avrdudess-a-gui-for-avrdude/
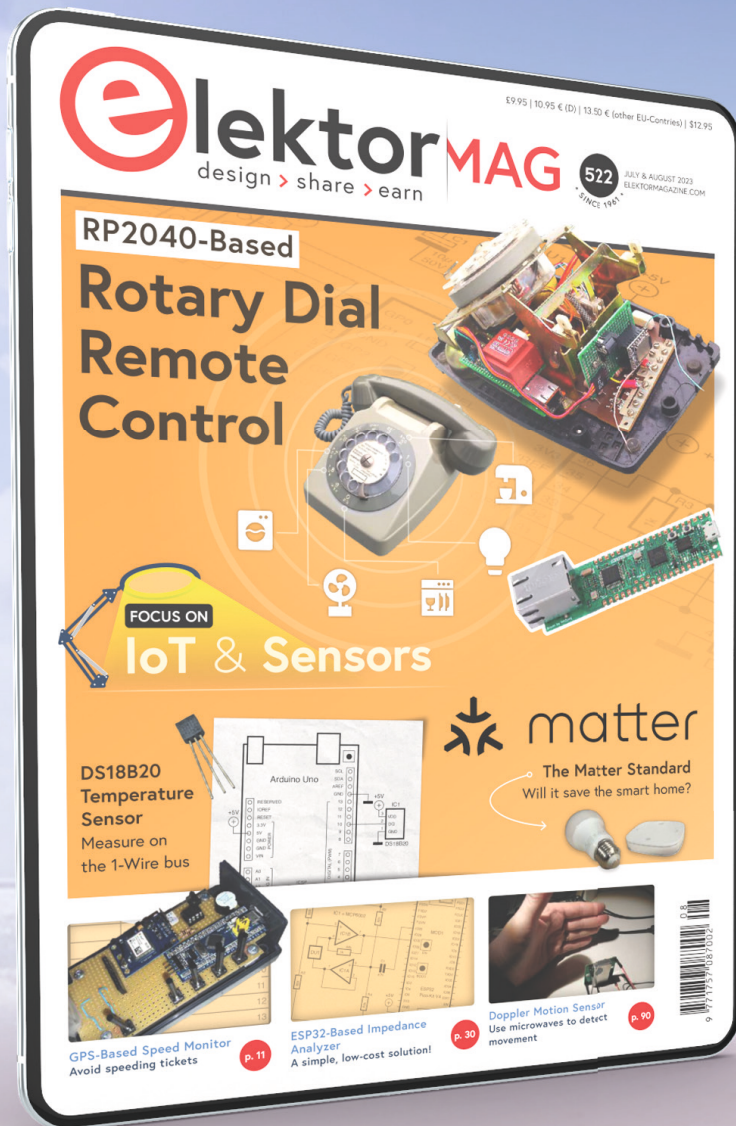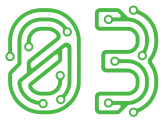
Quick ESR Test

By Burkhard Kainka (Germany)

Need to test the equivalent series resistance
of a capacitor? Here is a simple solution.

The equivalent series resistance (ESR) of a capacitor is its loss resistance, which among other things can cause an electrolytic capacitor to get warm in operation. If the ESR of an electrolytic capacitor is higher than normal, it's a sign that the capacitor is no longer usable. For this quick test, I used a 20 kHz square-wave signal, for example generated by a microcontroller. The combination of a 100 Ω resistor and the capacitor being tested forms a low-pass filter (**Figure 1**).
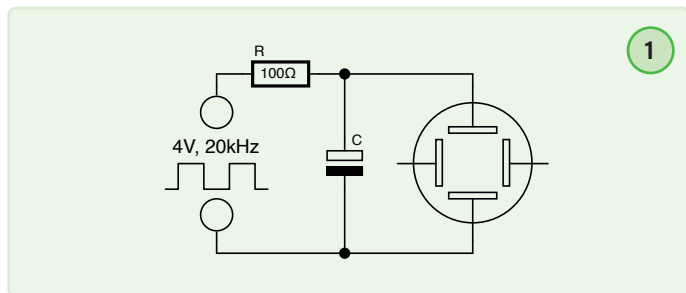
You can view the filtered waveform on an oscilloscope. The sloped parts of the waveform show the expected charge and discharge behavior of the electrolytic capacitor (**Figure 2**). The step changes between these are caused by the ESR. The measured signal amplitude at the microcontroller is 4 V, since the resistance on the port is very low.
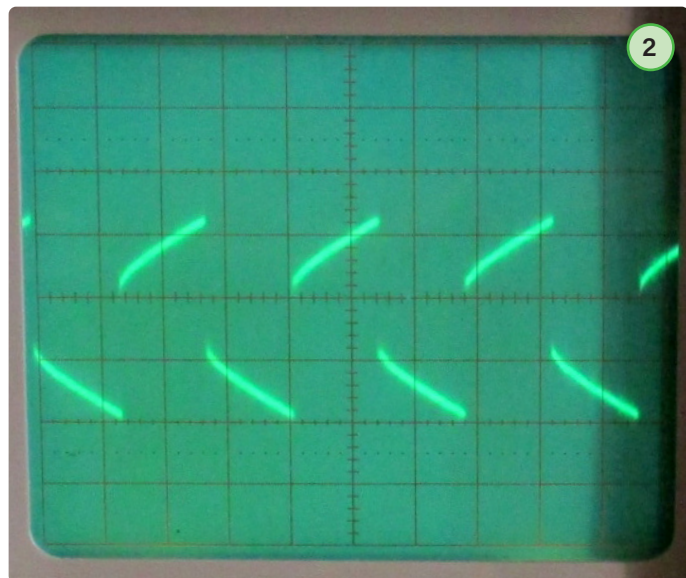
The ESR depends on the size of the electrolytic capacitor. For example, I checked two 100 µF electrolytic capacitors with different case sizes. The larger one showed voltage steps of 10 mV, which is only 1/400 of the 4 V signal amplitude at the microcontroller. This means the ESR is a factor of 400 smaller than the 100-Ω series resistor, so the ESR is 0.25 Ω.

The result was the same when I increased the frequency to a higher value, in this case 200 kHz (**Figure 3**). In both situations, the smaller capacitor showed larger voltage steps of 15 mV. This means it had an ESR of around 0.375 Ω. A larger 1000-µF electrolytic capacitor showed steps of only 2 mV, meaning it had an ESR of 0.05 Ω. ◄

Translated by Kenneth Cox — 230129-01

About the Author

Burkhard Kainka (b-kainka.de) is a radio amateur and well-known Elektor article and book author. After spending many years as a physics teacher, he became a self-employed developer and author in 1996.
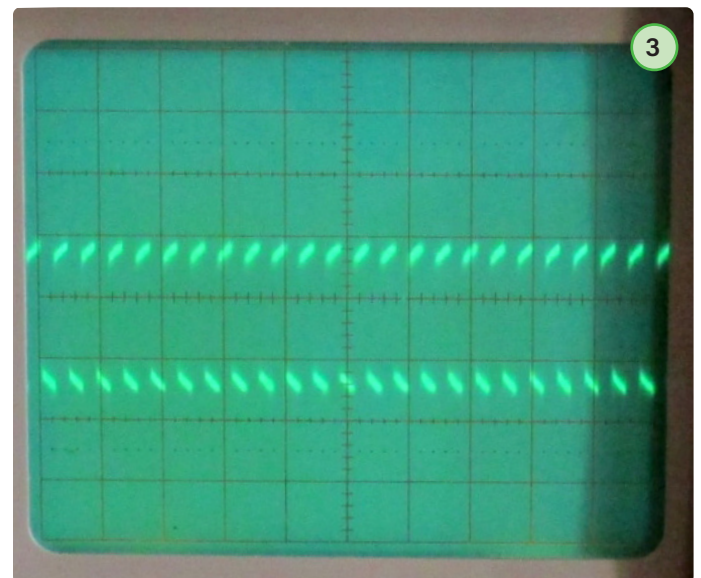
04 Human Interface for DIY Home Alarm System

LCD, Voice, and RFID with Raspberry Pi Control
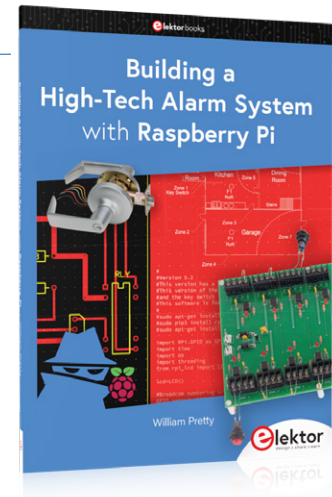
By William Pretty (Canada)

To implement a modular alarm, no matter if it is for a single house or for a business or restaurant, William Pretty's recently published book shows how to combine a Raspberry Pi skillfully with small auxiliary electronic circuits. Instead of installation instructions, the book presents food for thought that will enable you to find a solution to your needs. Here's a sample chapter from the book, concentrating on the system's three-element human interface.

Editor's Note: *This article is an excerpt from the 115-page book,* Building a High-Tech Alarm System with Raspberry Pi *(Elektor, 2023). The excerpt was formatted and lightly edited to match Elektor Mag's editorial standards and page layout. The author and editor have done their best in such editing and are happy to assist with queries. Contact details are in the* **Questions or Comments?** *box.*

In this article, we will discuss how the alarm panel communicates with the natural world: by display, by voice, or by RFID Card Reader. The three elements jointly form the human interface of the High-Tech Alarm System described in the book.

Display

First, we will cover the 2×16 LC Display (**Figures 1 and 2**). This display can display two rows of 16 characters. There are two methods of sending data to the display, and we will be using the type with a serial interface because it is both easy to connect to the Raspberry Pi and easy to program.

The pinout of J3 on the alarm system motherboard is identical to that on the adapter board soldered to the back of the display. If we consider the pin of the adapter closest to the top edge to be pin 1, then pin 1 is connected to pin 1 (GND) of J3 on the alarm panel motherboard. Pin 2 is connected to pin 2 (+5 V) of J3. Pin 3 is connected to pin 3 (SDA) of J3, and pin 4 is connected to pin 4 (SCL) of J3 (**Figure 3**).

The first thing we must do is to enable the I²C interface on the Raspberry Pi. The easiest way to do this is to configure a development system with an HDMI display, a keyboard, and a mouse. My system is running the latest version of the OS, called Buster.
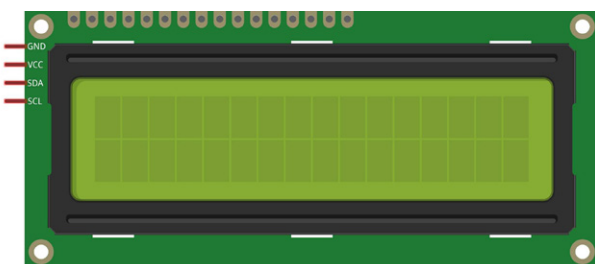
Figure 1: Front view of the 2×16 LCD module.
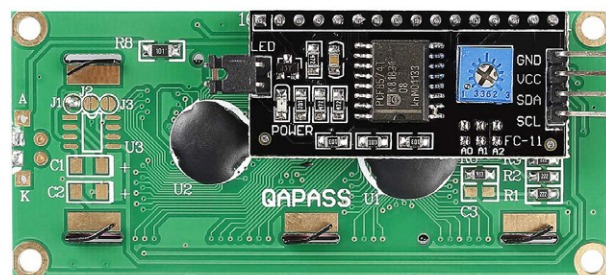
Figure 2: Rear view of the 2×16 LCD module.
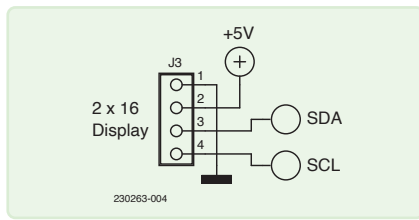
Figure 3: Motherboard J3 connector pinout.

Figure 4: Raspberry Pi configuration menu.

Click on the Raspberry Pi icon on the top left of the screen and you should see the menu shown in **Figure 4**.

Next, select *Raspberry Pi Configuration* and you will see a pop-up window as in **Figure 5**.

Select the *Interfaces* tab and use the radio button to make sure that the I²C interface is enabled. We will have to reboot the system for any changes to take effect, so now is a good time to power down the Raspberry Pi and connect the display to the Raspberry Pi.

You will notice that a breakout adapter is used to connect the Raspberry Pi to my breadboard (**Figure 6**). It's highly recommended as it will make your building and evaluating a lot easier. As a bonus, it comes with a 40-conductor ribbon cable, which we can eventually use to connect the Raspberry Pi to the motherboard of the alarm system.

It is now time to power up the system and start writing our first Python script. Before you can do that, you must install a library and some tools. The first thing to do is make sure that PIP is installed. You do this with the following command:

```
sudo apt-get install python3-pip
```

This will install PIP3, if it was not already included with Buster. Once you have PIP3 installed, you can install the serial LCD library to employ:

```
sudo pip3 install rpi_lcd
```


Figure 5: Interfaces to enable or disable.

Figure 6: Raspberry Pi breadboard circuit and connections.
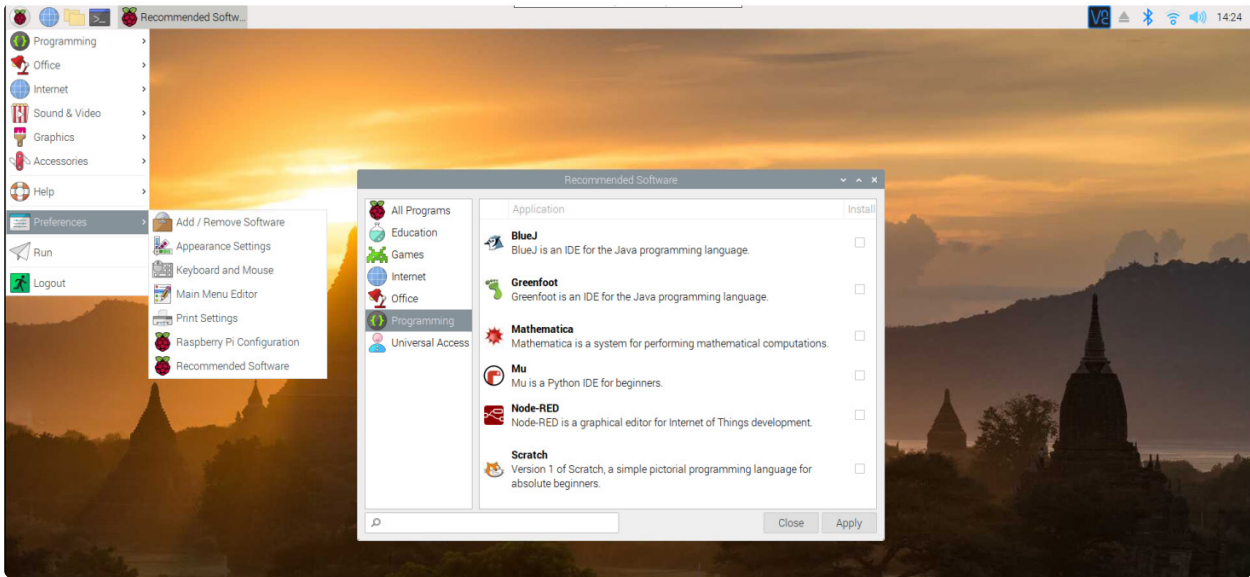
Figure 7: Installing Geany.

With these elements installed, it is time to pick an IDE (Integrated Development Environment). My personal favorite is Geany. You can install it from the drop-down menu, like in **Figure 7**.

I like Geany because once you save the file you are editing as a *.py* file, it will help you with formatting such as indenting and assorted colors for different components of your program, such as variables and comments. It also does simple syntax checking. Before we can use Geany as an IDE, there is some setting up to do. As shown in **Figure 8**, click on the *Build* menu item on the taskbar and select *Set Build Commands*.

If all's going well, you will see the *Set Build Commands* pop-up window as in **Figure 9**. Next, we need to enter the following information into the IDE:

> Under the *Python commands* section, line 1 (*Compile*) should read: `python -m py_compile "%f"`
> Under the *Execute commands* section, line 1 (*Execute*) should read: `python3 "%f"`

From here on, it's assumed that we are using Geany to write and run our programs.

Create a new file by clicking on the far-left icon on the taskbar.

This will open an empty file. At this point, you should save the file as a Python script with a name such as *lcd_test.py*. This will tell the IDE that you are writing a Python program, and it will help you with formatting and component colors. Enter the following code:

```
from rpi_lcd import LCD
lcd = LCD()
lcd.text("Hello", 1)
lcd.text("World", 2)
```

If you now click on the paper airplane icon, the program will run in the IDE environment — see **Figure 10**.

If there are no typing errors, the program will run, and you should see the following displayed on the LCD:

```
Hello
world
```


Figure 8: Pull-down menu for selecting Set Build Commands.
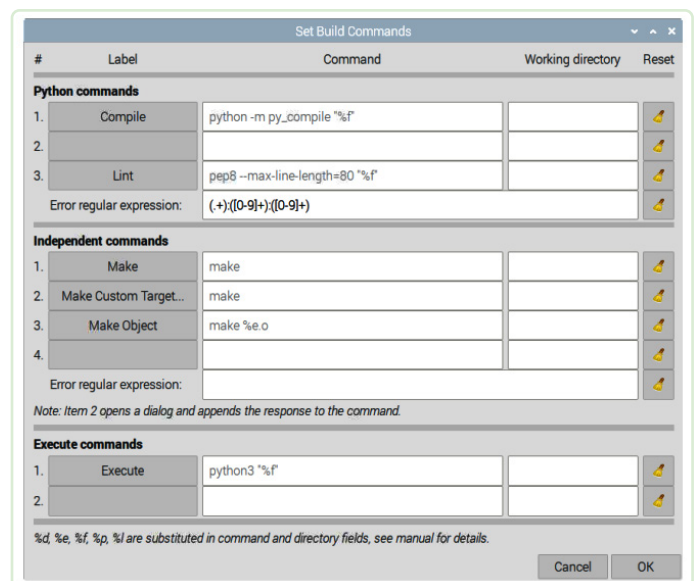
Figure 9: Pop-up window for the various command options.

If you see these words displayed on the LCD, then the screen is connected properly, and everything is working properly. You may have to adjust the contrast using the potentiometer on the back of the LCD.

If you don't see anything on the display and your electrical connections are correct, then it's troubleshooting time. The first thing to check is the actual address of the LCD on the I²C bus. We do this by opening a terminal window and entering the following command:

```
i2cdetect -y 1
```

If your connections are correct, the display should appear in the address table (**Figure 11**). If you don't see the display address anywhere in the table, then the display is not being detected for some reason, and it may be damaged.

The default address of the LCD should be 0x27 (hex). If you see a different address, you will have to modify our code slightly.

The LCD object can be created with the address as parameter:

```
lcd = LCD(address=0x3f)
```

or whatever address is shown by *i2cdetect*.

Voice Output

The alarm system software is capable of speaking to you in one of four languages; English, French, German, or Dutch. I am an English speaker, so I had to use an online translation program to get the other three languages. I apologize if the translation is not the best. The software package that we'll be using parses a text string and "speaks" whatever is in the text string using a phoneme-based library. That program is called eSpeak, and here is how we go about using it. The first thing we have to do is to install the eSpeak package using the following command:

```
sudo apt-get install -y espeak
```

This version of eSpeak is the command-line version. In the Buster version of Raspian, the Python3 version of eSpeak broken, so we will be using a workaround. We will be piping the output of espeak to Alsa Audio, so we will have to install that package as well:

```
sudo apt-get install alsaaud*
```

I will be covering the basic commands used by the alarm system in this article. A more complete description of the commands and languages is available from [1]. The program shown in **Listing 1** will allow the user to test the eSpeak software with various voices and languages.

The first thing we have to do is to import the *os* package so that we can pipe commands to Alsa. Next, we import *time* so that we can sleep for five seconds between statements.

eSpeak tends to talk rather fast, so we slow it down with the command `os.popen('espeak -s100')`. This sets the rate to 100 words per minute. You can play with this command to suit your language.

These lines cause eSpeak to actually "speak" the text string sent to it:

```
speak = "Alarm detected in Zone 1" #English
os.popen('espeak -ven+f5 "' + speak + '"--stdout | aplay 2>
/dev/null').read()
```

The `espeak` command tells eSpeak to speak English with female voice #5. This is piped to the Alsa application (`aplay`). There are several male and female voices available (+f1 through +f5 and +m1 through +m7). I found the male voices sounded too much like a robot. Feel free to assess them all.
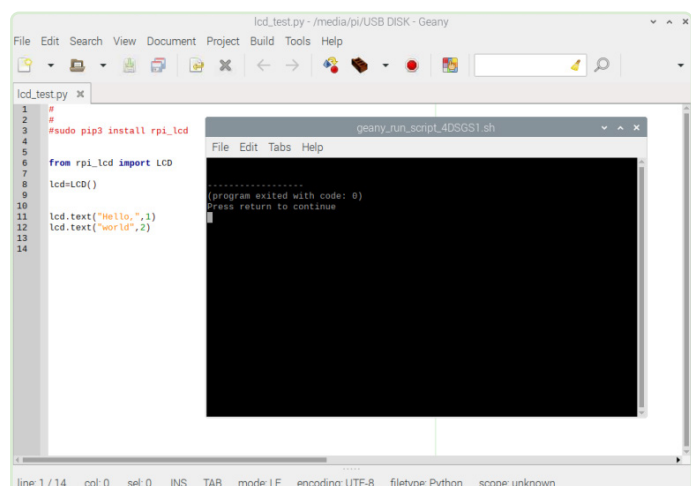
Figure 10: Running the test program.
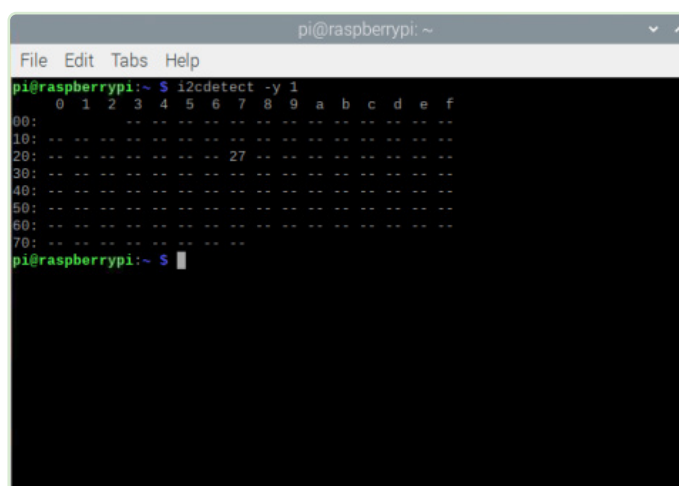
Figure 11: The i2cdetect tool in action.

Listing 1: Language selection and test program for eSpeak.

```
#
# sudo apt-get install -y espeak
#
# In the latest version of the OS (Buster), this program is broken
# sudo apt-get install -y python3-espeak
#

import os
import time
os.popen('espeak -s100')

while True:

        print("English")
        speak = "Alarm detected in Zone 1" # English
        os.popen('espeak -ven+f5 "' + speak + '" --stdout | aplay 2> /dev/null').read()
        time.sleep(5)

        print("French")
        speak = "Alarme detectee dans la zone 1" # French
        os.popen('espeak -vfr+f5 "' + speak + '" --stdout | aplay 2> /dev/null').read()
        time.sleep(5)

        print("German")
        speak = "Alarm in zone eins entdeckt" # German
        os.popen('espeak -vde+f5 "' + speak + '" --stdout | aplay 2> /dev/null').read()
        time.sleep(5)

        print("Dutch")
        speak = "Alarm gedetecteerd in zone een" # Dutch
        os.popen('espeak -vnl+f5 "' + speak + '" --stdout | aplay 2> /dev/null').read()
        time.sleep(5)
```


Figure 12: A USB-controlled RFID card reader suggested as part of the human interface section.

RFID Card Reader

This project uses an RFID card reader that plugs into a USB port on the Raspberry Pi (**Figure 12**). The reader appears to the operating system as a human interface device (HID). Basically, the Raspberry Pi thinks that the card reader is a keyboard. Before we can use the reader to arm and disarm the system, we must create a database of known RFID cards. In our case, it will be a plaintext file, and we will create it using the text editor that comes with the system. To do this, we select *Accessories* from the dropdown menu, then *Text Editor* — see **Figure 13**.

Next, open a text file called *cards.txt*. This will be our database of acceptable cards. Adding cards to the database is a simple matter of swiping the cards across the card reader. The reader acts like a keyboard and sends a text string that corresponds to the serial number of the card, followed by a carriage return. An example of this output is given in **Figure 14**.

Once you have all the cards entered into the file, save the file in the same directory as the Python script. Now it is time to create a special database file called *super.txt*, which will contain "supervisor" cards.
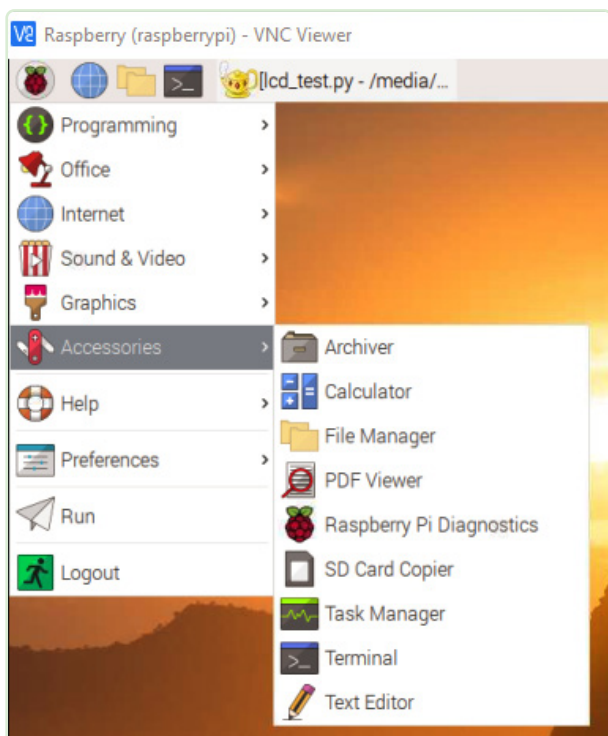
Figure 13: From Accessories, select Text Editor.
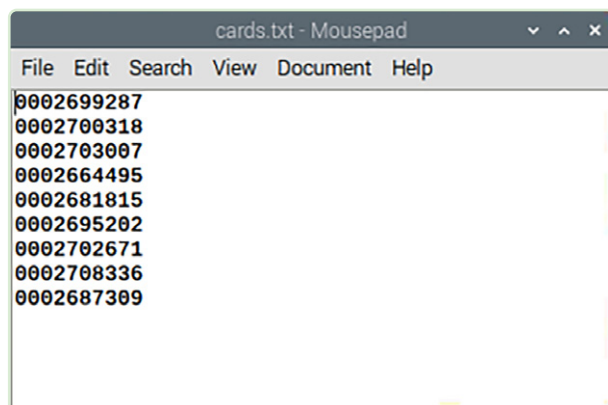
Figure 14: RFID idents written to a text file by swiping cards.

Only supervisor cards can arm the system. Once the system is armed, it can be disarmed by any valid card or key. Once the system is no longer in the *System Idle* state, it can be armed or disarmed using the key switch. Rotate the key to the left to disarm the system. Rotate the key to the right to arm the system. The key can only be removed if it is in the center position.

If you run the script, the LCD will say "System Idle." This means that the system is waiting to be armed. If you tap the card reader with a valid supervisor card, the LCD will display "System Armed" and the alarm system voice will also respond by saying "System Armed." The system is now capable of detecting intruders and fires. To disarm the system, tap a card again. The system will now say and display "System Disarmed."

The fire-detection system is different from the burglar alarm portion of the panel. The panel will still detect fires using the sensors connected to the fire zones even though the burglar alarm is disarmed. If a fire (heat rise) or smoke (smoke detector) is detected, the alarm siren will sound. The siren can only be silenced by tapping the card reader twice with a supervisor card or turning the alarm key switch to the *Disarm* position.

Here is a typical scenario. You disarm the burglar alarm because you are going to bed for the night, and you don't want someone to trigger the alarm because they got up for a glass of water. Unfortunately, someone has left something on in the kitchen, which started a fire. Depending on the type of fire, either the heat rise (flame) detector or the smoke detector will trigger an alarm. You now have time to investigate the source of the alarm and to take appropriate action. ◄

230263-01

Questions or Comments?

Do you have technical questions or comments about this article? Send an email to the author at bill.pretty@mail.com or contact Elektor at editor@elektor.com.

Related Product

> **William Pretty,** *Building a High-Tech Alarm System with Raspberry Pi*, **Elektor 2023**
https://elektor.com/20443

WEB LINK

[1] eSpeak, open source software speech synthesizer: http://espeak.sourceforge.net

05 Fitness Timer

By Guy Weiler (Luxembourg)

"Use it or lose it" is the mantra of fitness gurus; spend too much time slouching and your fitness and health will suffer. The programmable Fitness Timer presented here is your own personal trainer to help keep your workout on track.

There is no doubt that many of us spend too much time sitting crouched over a keyboard. Health experts advise that regular exercise is important for keeping fit, but person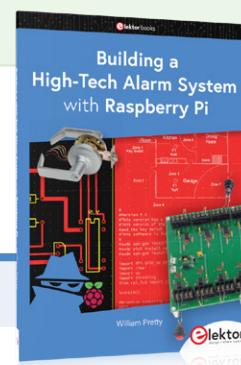al trainers are a prerogative only of the affluent. Not everyone is good at multitasking; it's difficult to stick to an exercise regimen when you are trying to strike the pose and count down the seconds at the same time… how many reps was that? It takes away some of the fun of keeping fit. That's where the Fitness Timer comes in!

Functionality

The Fitness Timer generates three tones. A lower tone signals the start of the exercise at first. After, for example, 45 seconds, a higher tone indicates the completion of one repetition. Then, a short pause (e.g., 15 seconds) follows, and the next repetition starts. After several repetitions, a triple-tone signals the end of the exercise.

There are two push buttons to choose between two exercise programs. In total, there are five programs, which are predefined in software. Using a rotary encoder with a button, the two push buttons can be assigned to two of these five different exercise programs. To do so, press the rotary encoder, select the program, press the desired push button (B1 or B2) and press the rotary encoder again.

The selected program is then shown on an OLED display, for example:

3×45 15 1

The last digit indicates which of the two push buttons can be used to select the program (**Figure 1**). When a button is pressed, the program starts after a 5-second pause, allowing you to get into position for the first exercise.

All programs can be freely defined in software.

Hardware and Software

In my tinkering box, I happened to have a LOLIN/WEMOS D1 mini Pro (V2.0) and a D1 mini Shield (V2.1) which has a 0.66" OLED display and two I²C-readable push buttons on board.

Figure 1: The display shows the number of repetitions, the designated durations, and break periods between exercises.
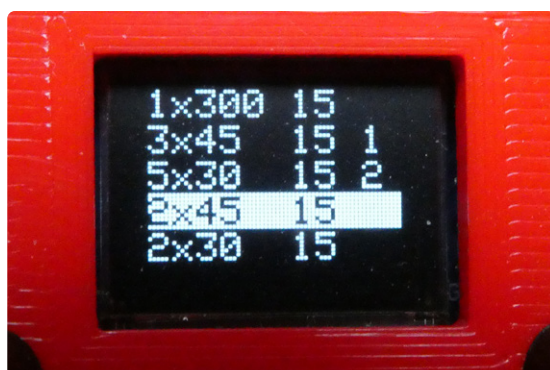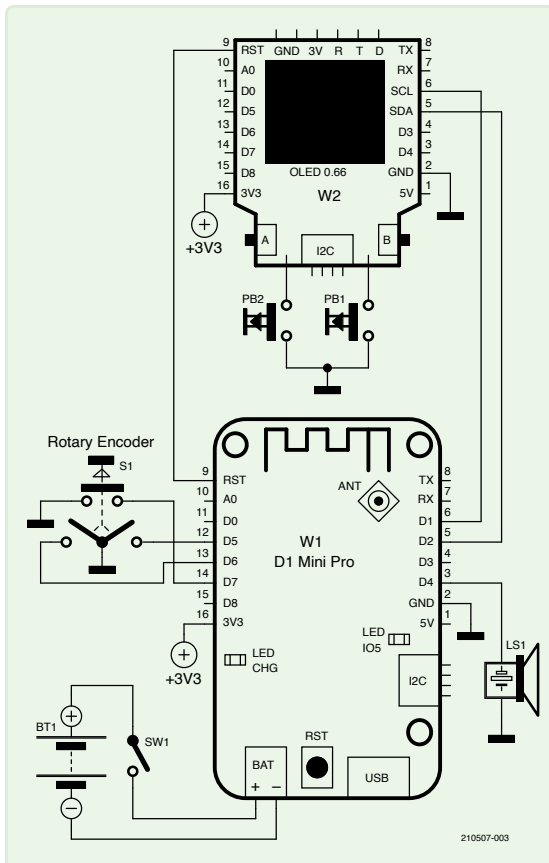
◄ *Figure 2: Fitness Timer circuit diagram.*

About the Author

Guy Weiler has been passionate about electronics since elementary school. To share his enthusiasm for the subject, he chose a career in teaching after studying microelectronics and telecommunications. Guy Weiler is a fan of the open-source philosophy; you can find all his projects and courses on weigu.lu.

Questions or Comments?

If you have any technical questions or comments about this article please contact the Elektor editorial team at editor@elektor.com.

Related Products

> **WEMOS D1 mini Pro**
  https://elektor.com/19185

The board has a connection for a lithium battery, so I didn't need to develop a separate power supply. In addition, I needed a rotary encoder with a push button function, a switch, a piezo speaker, and two push buttons.

The wiring can be seen in the circuit diagram in **Figure 2**. I soldered the two buttons in parallel with the two push buttons on the OLED shield because there are limited GPIOs available on the D1 mini Pro processor board. I then designed an enclosure (using FreeCAD) and 3D-printed it (**Figure 3**).

The Fitness Timer can also be easily re-engineered using different hardware by adapting the software accordingly.

The software and FreeCAD files are all available on GitHub [2]. More information about the Fitness Timer can be found on my website [3]. ◄◄

Translated by Martin Cooke — 210507-01
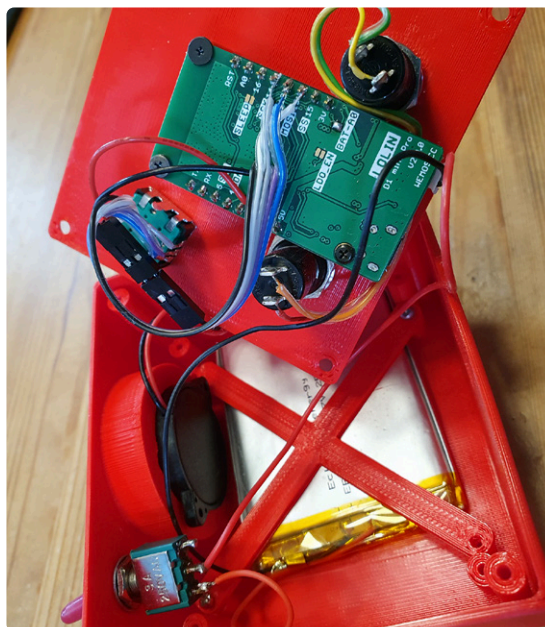
Figure 3: The enclosure was designed using FreeCAD.

◄

WEB LINKS

[1] Project at Elektor Labs: https://elektormagazine.com/labs/fitness-timer-do-your-fitness
[2] Project at GitHub: https://github.com/weigu1/fitness_timer
[3] Project on the author's website: http://weigu.lu/microcontroller/fitness_timer/index.html
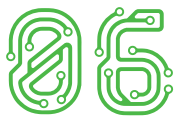
06 Ni-Cd Battery Rejuvenator

Don't Throw Those Old Batteries Away Yet!

By David Ashton (Australia)

Got dead Ni-Cd batteries lying around? You can still revive those! This clever circuit, featuring a 555 timer, brings them back from the dead. By delivering a "zap" to the battery followed by a controlled discharge multiple times, it eliminates dendrites and restores the battery's normal operation.

You may have old equipment about that uses nickel-cadmium (Ni-Cd) batteries, and does not work anymore, and you can't find suitable replacements. There are two things that go wrong with Ni-Cd batteries: They can lose their ability to hold a charge, or they appear to short-circuit. The loss of ability to charge is usually caused by overcharging, usually in equipment with an unlimited "5-hour" charger. Leave it charging for longer than this and your batteries are unusable and cannot be recovered.

Understanding Battery Failure

The short-circuited batteries are caused by whiskers of cadmium called "dendrites," which grow across the electrodes and short them out; this usually happens when batteries are left discharged for long periods. A quick "zap" at around 3 to 5 times the usual charge rate will usually burn out these whiskers and restore the battery to normal operation, but, often when this is done, the battery often rapidly returns to its shorted state.

If a Ni-Cd battery has a capacity C (usually given in Ah or mAh), you would usually charge it at $0.1C$ (i.e., 100 mA for a 1 Ah battery), or maybe higher — up to $0.3C$ — if you limit the charge by time or battery temperature rise. Obviously, a sustained charge at $3C$ or more is not a good idea.

What we do here is to "zap" the battery at around $3C$ to $5C$ for a short time (about 5 to 10 seconds) and then discharge it at a slower rate (around 0.1 to $0.2C$) until it discharges to about 1.1 volts (just under the nominal voltage of 1.2 volts), and repeat this process many times. A 555 timer is ideal for both these requirements.

The Circuit

The schematic is shown in **Figure 1**. RV1 should be adjusted for 2.2 V on pin 5. This will make the 555 trigger when pin 2 goes below 1.1 V. The relay, RLA, is normally off, and the battery under test (BUT) will discharge through Rd (selected for $I = 0.1C...0.2C$) until

Table 1: Suitable values for Rc and Rd for a 5 V supply against common battery capacities.
For batteries of intermediate capacities, interpolate between the nearest values.

Battery Capacity	100 mAh	300 mAh	500 mAh	1 Ah	2Ah	3 Ah	4 Ah	5 Ah	7 Ah
Max I$_{SUPPLY}$	0.5 A	1.5 A	2.5 A	5 A	10 A	15 A	20 A	25 A	35 A
Rc Ω **Rc W**	10 Ω 3 W	3.3 Ω 10 W	2.5 Ω 15 W	1 Ω 25 W	0.5 Ω 50 W	0.33 Ω 75 W	0.25 Ω 100 W	0.2 Ω 125 W	0.15 Ω 170 W
Rd Ω **Rd W**	120 Ω 0.25 W	33 Ω 0.25 W	22 Ω 0.25 W	12 Ω 0.5 W	6 Ω 0.5 W	3.9 Ω 0.5 W	3.3 Ω 0.5 W	2.2 Ω 1 W	1.8 Ω 1 W

Figure 1: Battery rejuvenator circuit diagram. Consult Table 1 for suitable values for Rc and Rd.

Figure 2: Shrink-wrapped battery packs.

Figure 3: Using dressmaking pins to measure the voltage of a (rather flat) Ni-Cd cell in a pack.

Figure 4: Using dressmaking pins to connect to one cell in a Ni-Cd Pack.

the voltage gets below 1.1 volts. The 555 will then trigger and operate relay RLA for a period determined by RV2 and C1, which should be set for about 10 seconds. When RLA operates, the BUT will charge at a rate determined by Rc, which should be for a current of about 3...5C of the battery, when short-circuited. When the timer times out, RLA is restored to the discharge position and will remain so until the battery voltage falls below 1.1 V again. As this is a fairly simple circuit, strip-board is adequate for construction, but make sure to use suitable wires for the high-current connections.

If you experience strange results, it is probably due to noise on the 5 V line. In this case, try supplying the circuit from another power source, such as a 5 V 3-terminal regulator (7805) from the PC power supply's 12 V line. The optional *Hold* switch will hold the circuit in Discharge mode (e.g., for changing batteries).

The 5 V supply should be capable of supplying at least 5C of the battery — so for a 1 Ah battery, if Rc is 1 Ω it will supply 5 A to a shorted battery. An old PC power supply is ideal, as they can usually supply at least 15 A, but a lot more is needed for larger batteries.

Initially, the circuit will give the 5C charge for 10 seconds and will often almost immediately give another 10 s charge, so Rc needs to be capable of supplying this current indefinitely — in this case 5 × 5 / 1 = 25 W! Make it up of 4× 10 W resistors, and you'll be well within spec. After half an hour or so of this treatment, the discharge times should get longer and longer as the battery recovers when the dendrites are all burned off. Recovered batteries should be given a full normal charge as soon as they are removed from the rejuvenator.

Table 1 lists suitable values for Rc and Rd (5 V supply) against common battery capacities. Required supply currents for a 5C "zap" are also shown — for larger batteries, these are high. You can reduce this to 3C if you wish, as it usually works as well as 5C, except for on stubborn batteries. The wattages given are for continuous charge currents — they will dissipate these powers if a battery remains stubbornly shorted. If this happens, the battery is usually beyond redemption and should be disposed of. Don't try to recover batteries that have leaked; throw them out. Don't be tempted to modify the circuit to recover two or more batteries at once — you may end up overcharging one of them.

Tips and Tricks

Sometimes you may encounter a battery pack such as in **Figure 2**: Two or more cells enclosed in heat-shrinkable wrap. You measure and obviously one cell is dead, but which one? You don't really want to take the pack apart to find out. Here's a trick that can help you: Get a dressmaking pin (my wife had some; they're handy things!) and you can usually insert it under the plastic to make contact with the cell case (**Figure 3 and 4**). To connect to the positive tip of the cell, just use another pin on the case of the next cell up. You can usually make a good enough connection for measurement, and it will usually be good enough for the rejuvenator as well, but be careful with high currents. The holes will not usually expose the cell metal, and you can tape over them if you're worried. Hint: The groove around the top of the battery is usually the easiest place to get the pin in.
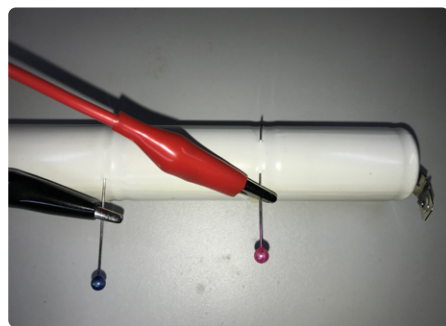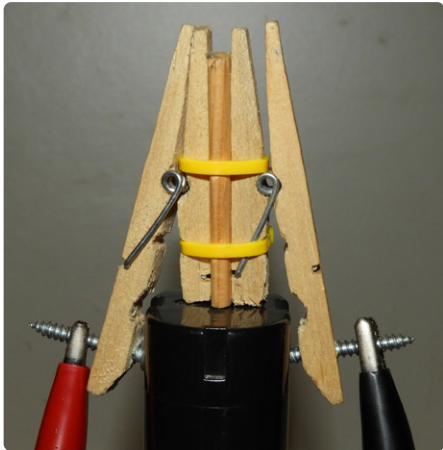
Figure 5: Battery test clip for power tool battery.

Once you've rejuvenated your battery pack, how do you test it, especially if it's a plug-in power tool battery with connections that you can't easily attach crocodile clips to? Using some clothes pegs, cable ties, screws and a bit of thin wood, you can make a test clip like this one (**Figure 5**). Using a bit of imagination, you can adapt this technique to make a jig to use on flat battery packs as well. ◀

230365-01

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

About the Author
David Ashton was born in London, grew up in Rhodesia (now Zimbabwe), lived and worked in Zimbabwe, and now lives in Australia. He has been interested in electronics since he was "knee-high to a grasshopper." Rhodesia was not the center of the electronics universe, so adapting, substituting, and scrounging components were skills he acquired early (and still prides himself on). He has run an electronics lab, but has worked mainly in telecommunications.
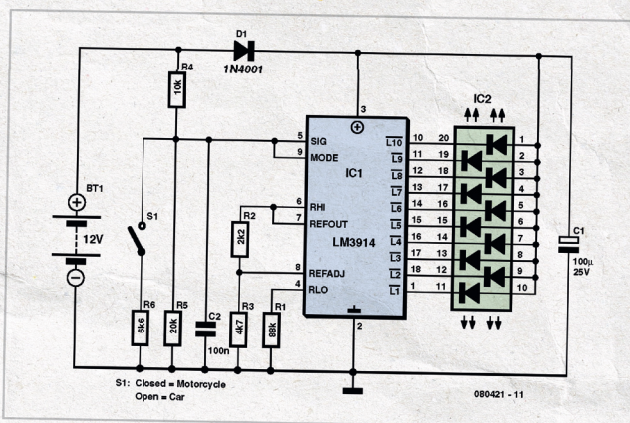
Quiz: Circuits from the Past #01

Car & Motorcycle Battery Tester

Joseph Zamnit

Going camping nowadays involves taking lots of electronic equipment whether for day to day running or for fun and entertainment. Most of the time a charged lead-acid battery and a power inverter would be used to ensure a smoothly organised holiday where ideally the missus and the children cheerfully use their electric and electronic gear!

With rechargeable lead-acid batteries it's invariably useful — if not essential — to determine whether the power source you're hauling along on your travels is losing capacity and needs to be topped up. The same circuit would also come in handy when going on a car or motorbike trip as it can check the status of a 12 V (car) or a 6 V (motorcycle) battery. Although the circuit draws so little power that it will not noticeably load the battery under test, it should not be left connected permanently.

The circuit employs the familiar LM3914 (IC1) to display the voltage level. The LED readout creates a battery status readout: when the top LED lights, the battery is fully charged. When the bottom LED lights, the battery needs imminent charging! Switch S1 selects between 12 V and 6 V operation. A series diode, D1, protects the bargraph driver from reverse supply voltage. A colour-coded display with individual LEDs could be used instead of the common-anode bargraph display for better indication of the state of the battery.
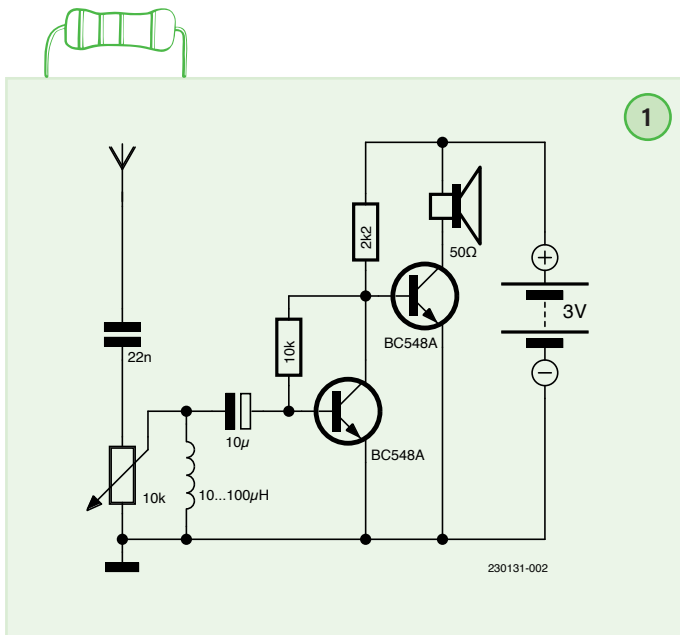
(080421-I)

Visit elektormagazine.com/quiz-23-1
to play now!

Take the quiz by August 31st, 2023.

07 All-in-One Shortwave

By Burkhard Kainka (Germany)

Interested in shortwave radio? Check out this simple two-transistor receiver with extremely wide bandwidth.

This broadband radio without any variable capacitor operates with a 20-m long wire outdoor antenna strung up to a tall tree. In the evening, the shortwave band comes alive. Then you can hear a lot of transmitters at the same time, and the various stations fade in and out, sometimes becoming very loud for a short time. This makes shortwave fun. Instead of constantly turning the dial to find a strong station, you simply wait and see what comes out.

The circuit diagram in **Figure 1** shows a simple two-transistor amplifier. However, strong RF signals are demodulated by the nonlinear base-emitter characteristic of the first transistor. This receiver performs so nicely that it should be built into an enclosure. The initial tests showed that the received signal was sometimes unpleasantly loud due to the frequent signal level variations, so I added a volume control to the circuit. Unconventional but effective: the potentiometer is connected directly to the antenna terminal. With so few components in this receiver circuit, it can be built in point-to-point style by soldering the components directly to the potentiometer, as shown in **Figure 2**. ◀

Translated by Kenneth Cox — 230131-01
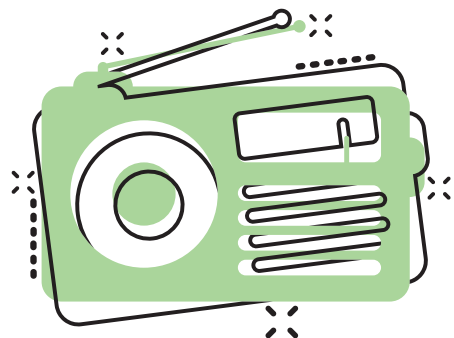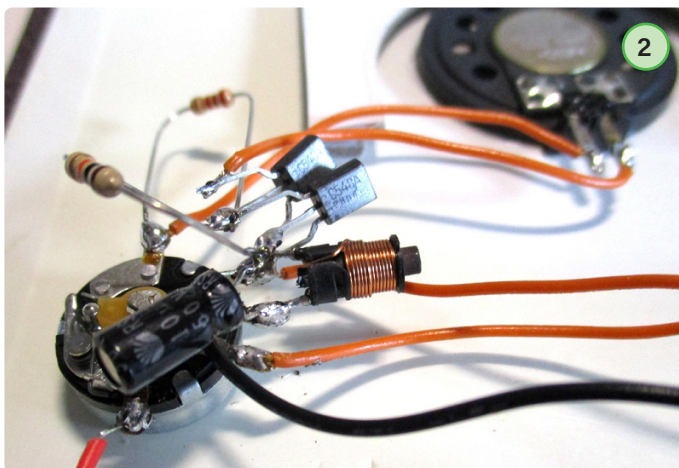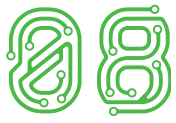
08 QucsStudio

A Powerful Circuit Simulator

By Mario Rotigni (Italy)

Looking for a free, yet powerful circuit simulator? Check out QucsStudio and its capabilities.

Many free circuit simulators are available today on the web. Each one has its benefits that make it best suited for a given application. QucsStudio is a circuit simulator that evolved out of the old project called Qucs. It is aimed at creating a complete development environment for electrical engineers. It has features rarely found in free simulation tools, and it deserves some attention. Many of the features are obtained by a clever interface versus specialized free programs. Let's take a quick tour of QucsStudio and its capabilities. (Refer to [1] for the official list of features for the current release 4.3.1.)

Features

QucsStudio has the classic SPICE simulation analysis, DC, AC, Transient, Noise. It also has native support for S-Parameters analysis, which is not easy to find in free simulators. Raising the operating frequencies, the capacitance and inductance, the classic parameters used to describe the properties and behavior of electrical interconnections become impractical to use (and measure). The insertion loss and return loss become more significant. These entities are better represented by the Scattering Parameters that describe the behavior in frequency of the active or passive network under analysis. Unfortunately, S-Parameters are not familiar to and not easy to approach for people with an electronics background (non-RF or microwave). Working out some extraction of S-Parameters for simple networks in QucsStudio could smooth the learning curve of this useful technique.

Another interesting feature, not easily found in free tools, is the support of digital simulations. In *Truth Table* mode the simulator calculates all the outputs' combination of a combinatorial network (without timing information). In *Timelist* mode the dynamic answer of the network is evaluated. The simulator also accepts logic networks described in the Verilog or VHDL languages and evaluates the dynamic response, plotting waveform and timing, as it does in Timelist mode.

The system simulation enables the analysis of data transmission channels, for example, taking care of the modulation eventually adopted. (The modulator can be represented by a mathematical model.) The eye diagram and the bit error ratio (B.E.R.) can be extracted, not only for electrical interconnections, but also for optical channels.
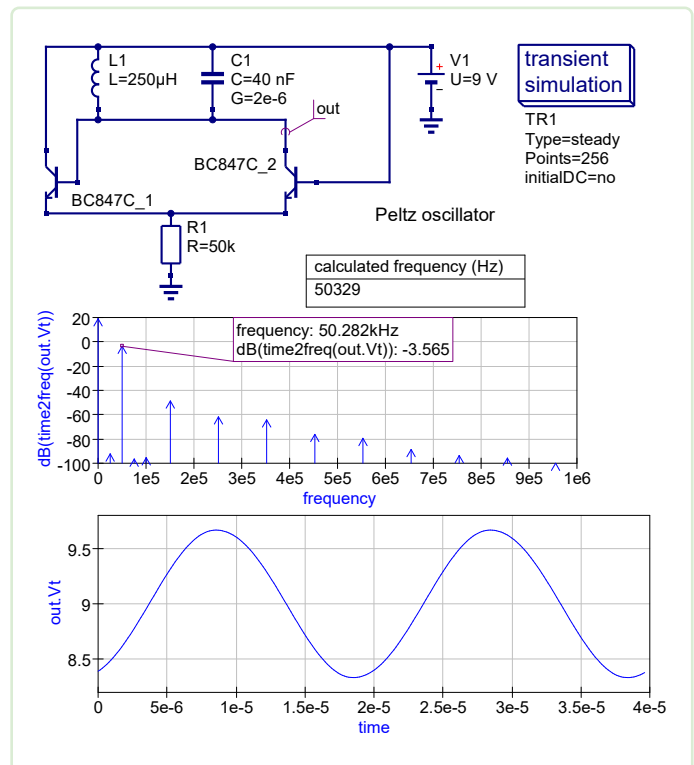

Figure 1: Transient simulation of a Peltz oscillator.

Another rare item in the free simulator, the electromagnetic 3D simulation, reads as input not only the schematics but also a layout and performs the extraction of the network answer in frequency, as S-Parameters and Smith-Chart. The effects of the physical interconnections, lost when working at the schematic level, can be captured here. Filters and patch antennas are provided as examples. It is also possible to compare the EM results with the S-Parameters calculated just from the schematics. Once the schematics/layout is optimized, it is possible to export the PCB for manufacturing in Gerber file format.

QucsStudio has integrated a *Transmission Line Calculator* mode that is able to work on stripline, microstrip, waveguides and cables. The calculated structure can be simulated, of course, to check its performance and sensitivity to the manufacturing parameters. QucsStudio also can be used to develop Octave scripts to perform complex mathematical operations. Octave can work as a powerful postprocessor for the QucsStudio simulation result as well.

Getting Started

Once you install QucsStudio, it is recommended that you visit the *Doc/Screenshot* page on the website. Watching the short videos is an effective way to explore the program's capabilities and start using it. The "Getting Started" chapters in the help section (accessed using F1) and the examples provided on the website's *Doc/Examples* are also very useful as an introduction.

Since QucsStudio is a free, self-supported tool, you can expect some limitations compared with professional tools. However, QucsStudio has much to offer, and it can be used by anyone looking to learn before moving on to commercial CAD tool suites. It is an evolving solution, so keep an eye on it. ◄

230252-01

🛒 **Related Products**

> **Elektor CheatKard (Electrical Engineering Reference Cards)**
> https://elektor.com/20080

> **D. Ibrahim, *Circuit Simulation with TINA Design Suite & TINACloud* (Elektor, 2022)**
> https://elektor.com/19977

QucsStudio
A Free and Powerful Circuit Simulator
Michael Margraf
http://qucsstudio.de

About the Author
Mario Rotigni was born in Bergamo, Italy, in 1958 and received a diploma in electrical engineering in 1977. He discovered electronics as a hobby when he was 14. In 1978, electronics became his profession, where he worked in R&D on the design of process instrumentation operating in very hostile electromagnetic environments. After designing an Automatic Test Equipment for micro-controllers, he joined a major semiconductor manufacturer, where he has held various positions in engineering, design, and now in R&D in the Automotive Product Group.

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

━ WEB LINKS ━
[1] QucsStudio Features: http://qucsstudio.de/Features/

09 The Arduino GIGA R1 WiFi Reviewed

A Portenta H7 for Makers?

By Clemens Valens (Elektor)

The Arduino GIGA R1 WiFi is a powerful board. With its two ARM Cortex-M7 and -M4 cores, it offers plenty of processing power. As it has audio and video ports, it's great for multimedia and gaming applications or machine learning and AI. Its form factor — identical to the Mega — ensures pin compatibility with existing shields. Let's have a closer look.
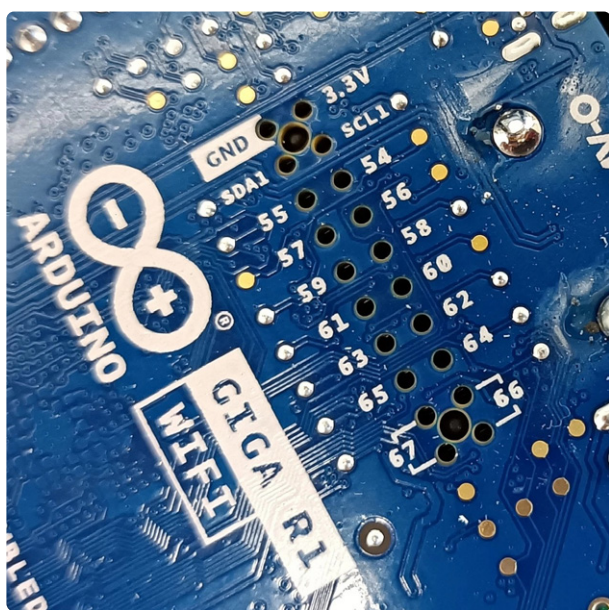
According to the Arduino blog, the Arduino GIGA R1 WiFi is the most powerful Arduino board ever. For makers, that is, because the GIGA R1 is based on the same microcontroller as the Portenta H7, the STM32H747. The Portenta X8 has one of these too, but with fewer pins. On the X8, it is even pushed into a corner because it's a kind of coprocessor for the big i.MX 8M SoC. (And, what about the RP2040 Nano Connect with its RP2040 plus ESP32?) So, the GIGA is not the most powerful Arduino board ever, even though it is very powerful indeed.

Isn't the Arduino GIGA R1 WiFi a Portenta?

As the GIGA has the same dimensions as the Arduino Mega, we kind of understand the reasoning behind the naming scheme. But its shape is about the only thing the GIGA has in common with the Mega. Its specifications are, unsurprisingly, very similar to the almost three times smaller Portenta H7 [2].

They both have a dual-core microcontroller, comprising an ARM Cortex-M7 running at 480 MHz and a Cortex-M4 clocked at 240 MHz. This MCU integrates 2 MB of flash memory and 1 MB of RAM. Both boards add 16 MB QSPI Flash memory and 8 MB of SDRAM to this. Again, both boards have a Murata module for wireless connectivity, meaning Wi-Fi and Bluetooth Low Energy (BLE).

Extension Headers Galore

Both boards also have a USB-C connector, but, while the H7 features DisplayPort over USB-C, the GIGA does not. However, the thing that makes the GIGA a board targeted at makers is its size. Because it is relatively big, it can have 0.1″-pitched pin headers (the H7 has two high-density micro headers) that let you plug extension shields and jumper wires into it.

Besides the Mega-style extension headers, the GIGA also has a display connector and a camera connector (**Figure 1**), both with a pitch of 0.1″. The special thing here is that these connectors are accessible from both sides of the board. This means that they remain accessible even with an extension shield plugged into the top of the GIGA.

A tiny 2×5 JTAG connector is available for programming, testing, and debugging the board and its software. I didn't try it, but the

Figure 1: The camera connector (shown here) and display connector are accessible from both sides of the board.

Arduino IDE 2.0 is probably capable of using it; don't worry, you don't need it, as the GIGA is a 'normal' Arduino board that can be programmed over its USB port (USB-C in this case).

Immature Software Support

To use the GIGA in the Arduino IDE you must first install the *Arduino Mbed OS GIGA Boards* package. At the time of writing, this package knows of only one board, so you can't pick the wrong one. You can configure the memory map and choose the processor core in the *Tools* menu (**Figure 2**). The included example collection is a mix of examples for other Mbed OS boards, including the Portenta.

Out of the box, the board flashes an RGB LED next to the audio jack, and so I decided to try the *MultipleBlinks* example. Compilation is surprisingly fast, and after uploading the sketch to the board (the RGB LED breathes green while doing this — nice), the LED blinked as intended. However, if I have understood the sketch correctly, it accepts On/Off commands over the serial port for the blue LED, but at the time of writing (mid-March), this didn't work for me. I couldn't control the LED and I didn't receive any acknowledgment messages in return.

Arduino GIGA R1 WiFi, a Powerful Board

With its two ARM Cortex-M7 and M4 cores, it offers plenty of processing power. As it has audio and video ports, it is great for multimedia and gaming applications. Its form factor being identical to the Mega ensures pin compatibility with existing shields but with 3.3 V signal levels. The board can be powered from up to 24 V, which is practical, and it comes stuck into a transparent base for support (**Figure 3**), protecting it a bit against short circuits.

On the software side, things seem still a bit immature, with only a few examples specifically targeting or even mentioning the GIGA R1. But, I am confident that this will be fixed in future releases of the boards package.

Retailing at almost € 70, the board is a bit expensive, if you ask me. Cheaper and smaller alternative Cortex-M7 boards are the Teensy 4.1 (NXP iMXRT1062) and the Daisy Seed (STM32H750) from Electrosmith. ◄
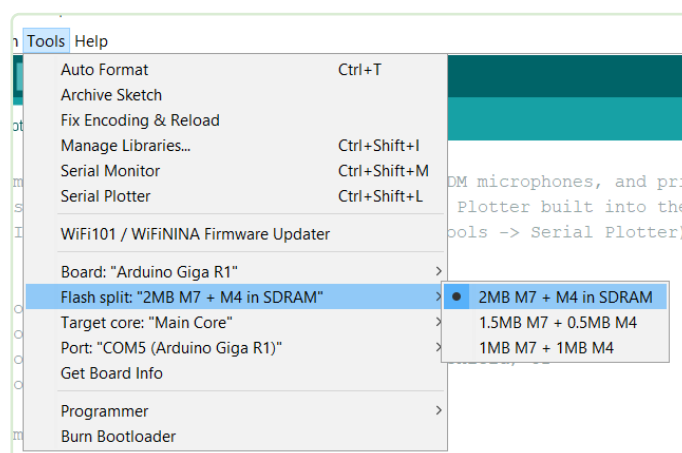
230347-01

Figure 2: You can configure how to divide the MCU's memory between the two cores.
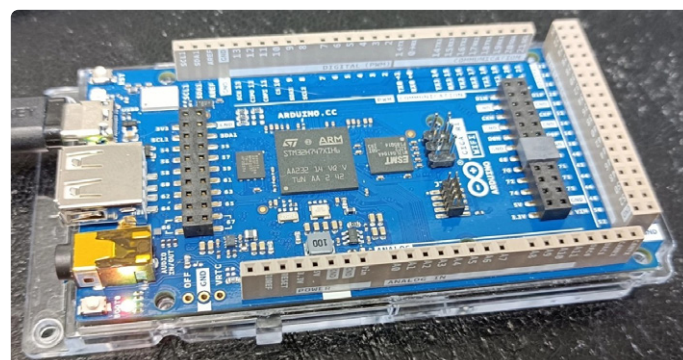
Figure 3: The Arduino GIGA R1 WiFi comes equipped with a transparent plastic bottom protection plate.

Questions or Comments?

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

Related Products

> **Arduino GIGA R1 WiFi**
https://elektor.com/20436

> **Arduino Pro Portenta H7**
https://elektor.com/19351

WEB LINKS

[1] GIGA R1 WiFi on Arduino blog: https://blog.arduino.cc/2023/03/01/step-up-your-game-with-giga-r1-wifi/
[2] Mathias Claussen, "Arduino Portenta Machine Control and Arduino Portenta H7," Elektor Guest-Edited Edition 2022: https://elektormagazine.com/magazine/elektor-268/61219

Elektor infographics

IC Market Growth (2022-2027)

High growth
Medium growth
Low growth
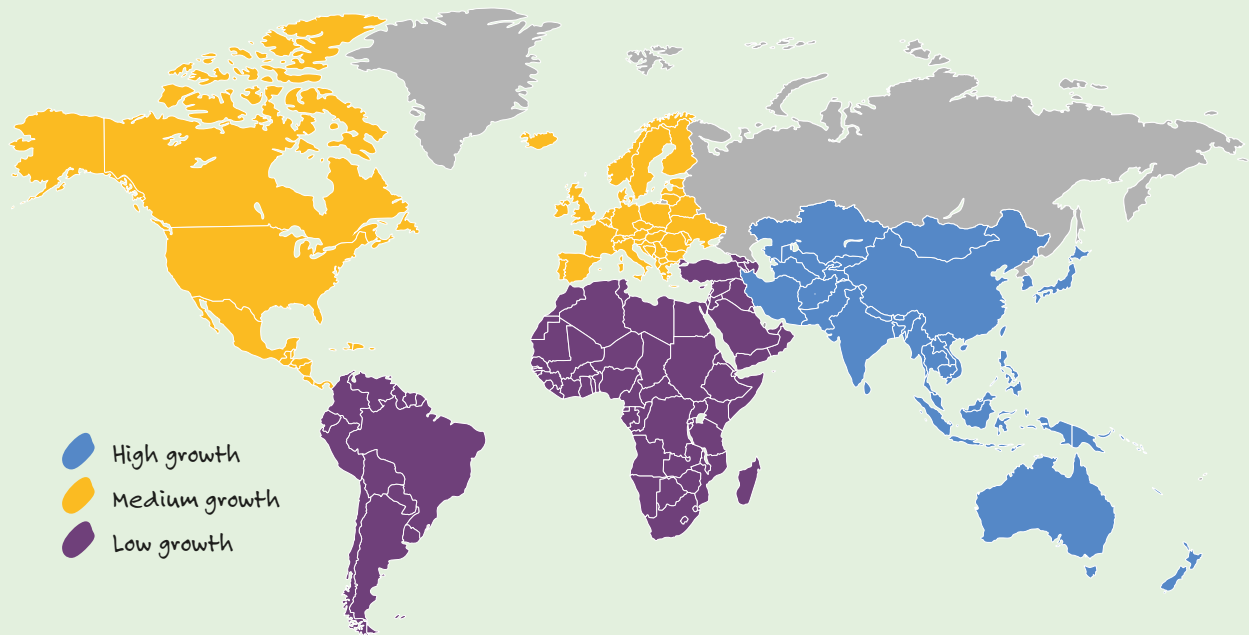
How Do You See **ICs?**

Integrated circuits (IC) are used in a wide variety of electronic devices, from digital clocks to complex microprocessors. With ICs, engineers can pack various electronic functions onto a single solution, thereby reducing the complexity of many designs. The global IC market has been on a steady growth path since 2009 — way back before Apple iPads were even on the market! According to Statista.com, the market reached $343.1 billion in 2017 and it hit $479.9 billion in 2022. [1]

IC Companies to Watch

> Analog Devices
> Infineon Technologies
> NXP Semiconductors
> STMicroelectronics
> Texas Instruments

The PCB Market: Where Is the Growth?

$19.05 billion

Projected global PCB market growth (2022-2027) [3]

4.25%
Projected YoY growth [3]

6.2%
Expected annual (2023-2030) growth of flexible PCB market [4]

Printed circuit boards (PCBs) play a key role in the work of professional engineers, students, and makers alike. Whether you need 1,500 PCBs for a new IoT solution that you intend to deploy in an industrial factory or you require a few PCBs to prototype a DIY project, it helps to stay informed about the global PCB market and any potential innovation coming down the pipe.

Electronic Design Automation Software (EDA Solutions)

Electronic Design Automation (EDA) design tools include, but are not limited to, the following: schematic capture, layout, simulation, test, and verification. Recent research reports indicate that the EDA market is growing, and it has been doing so since 2020, despite the challenges posed by the COVID-19 pandemic. For instance, Grand View Research recently reported that the global EDA software market hit $11.1 billion in 2022. [5]

Global EDA Software Market Share
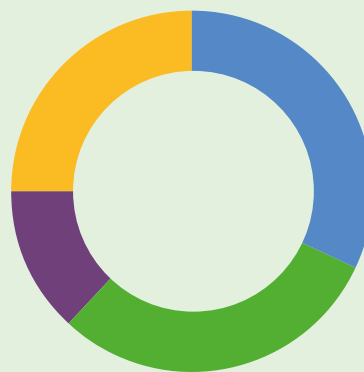
- 32% Synopsys
- 30% Cadence
- 13% Siemens EDA
- 25% Others

Source: Trendforce [6]

9.1%

Projected global EDA CAGR (2023-2030) [5]

EDA Software Companies to Watch
> Altium
> ANSYS
> Cadence Design Systems
> eInfochips
> Siemens EDA
> Synopsys

WEB LINKS

[1] Statista, "Integrated Circuits Semiconductor Market Size Worldwide from 2009 to 2023," March 2023: https://bit.ly/statista-ic-2023

[2] Mordor Intelligence, "Integrated Circuits Market Size & Share Analysis (2023-2028)," 2023: https://bit.ly/mordor-ic-2023

[3] Technavio, "PCB Market by End-user, Product, and Geography," Nov 2022: https://bit.ly/technavio-pcb-market

[4] Reliable Research Reports, "Global Flexible PCB Sales Market Report 2023," May 2023: https://bit.ly/reliable-research-flex-pcb

[5] Grand View Research, "Electronic Design Automation Software Market Size Report, 2030": https://bit.ly/grandv-eda

[6] Trendforce, "New US EDA Software Ban May Affect China's Advanced IC Design, Says TrendForce," 2022: https://bit.ly/trendf-eda
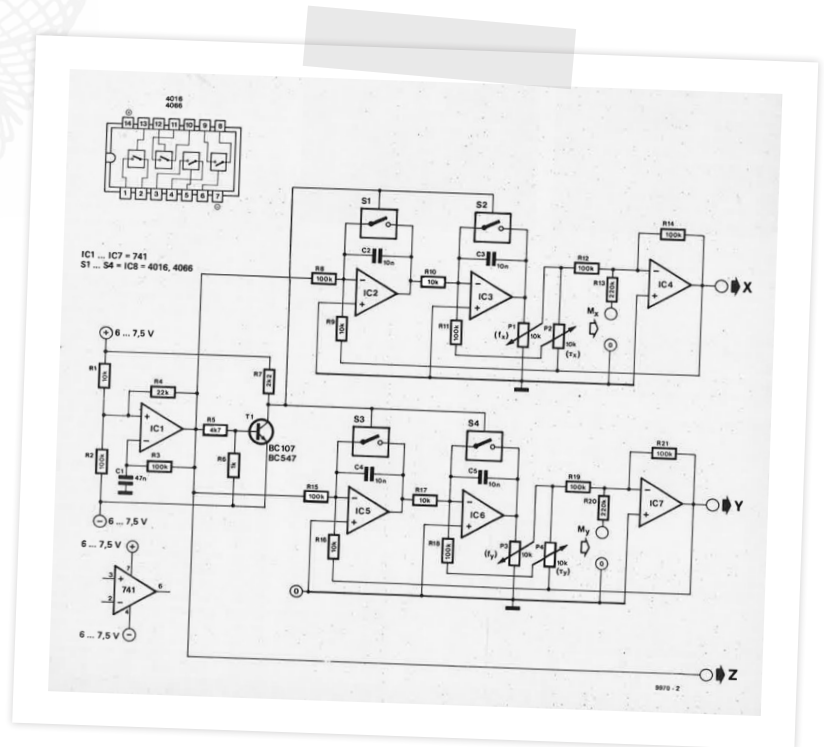
18 Oscillographics
with the Elektor "Spirator"

Now Try This on Your Digital Scope...

By Jan Buiting (Elektor Retronics)

Back in 1978, Elektor Magazine wrote: "An oscilloscope can be used not only as a test instrument; with the aid of the following circuit, it can be made to generate a multitude of fascinating and attractive geometrical patterns." Interestingly, 45 years on, this bunch of opamps may challenge the latest oscilloscopes with all the advantages of "digital."

Figure 1: Simple to build even in 2023, the Elektor "spirographics" generator consists of two damped-sinewave oscillators which are triggered by an astable multivibrator. Both the frequency and damping of the sinewaves can be varied independently. The oscillator output signals are used to control the X- and Y-deflection of an oscilloscope trace; the result is a fascinating display of "Lissajous" figures. Caution: Best connected to an analog, CRT-based oscilloscope!

In a big article in Elektor's September 1978 edition [1], M. Zirpel described a "spirographics generator" or "spirator" he said was capable of generating visual patterns similar to those produced by hand using the Spirograph™ outfit, as well as the type of figures often produced by computer graphics.

The patterns are derived from certain basic geometrical functions and are known as *Lissajous figures*. They are to be found in nature, for example in the path described by an object fixed to the end of a rope which is oscillating. In geometrical terms, a Lissajous figure is obtained when a point describes a sinewave on both the X and Y axes.

The circuit of the spirator reproduced in **Figure 1** from the original 1978 article [1] produces two sinewave voltages, the frequency of each being independently variable. Both sinewaves are damped, i.e., after the waveform has been started, the function will decay exponentially to zero.

The circuit also offers the particularly aesthetic possibility of displaying moving patterns by varying the frequency and/or damping of one or both oscillators. To this end, the circuit provides extra control inputs (M_x and M_y) for low frequency modulation signals (from, say, a sinewave generator).

Depending upon the type of oscilloscope used, it may be necessary to invert the Z signal, in which case the signal can be taken from the collector of T1. If the picture is not completely flicker-free, then the value of C1 should be reduced accordingly.

Z signal? Does my 100 MHz, 5 GS/s digital oscilloscope have an input for that? Nope, but all CRT (cathode ray tube) based scopes did. "Z axis" means CRT spot intensity modulation, besides movement in the X and Y directions on the scope screen. So why not dig out your best analog scope and watch it outperform the latest digital ones in at least one way?

Returning to the late 1970s, the "spirator" turned out to be so popular in hobby circles that Elektor published a tweaked version and a matching
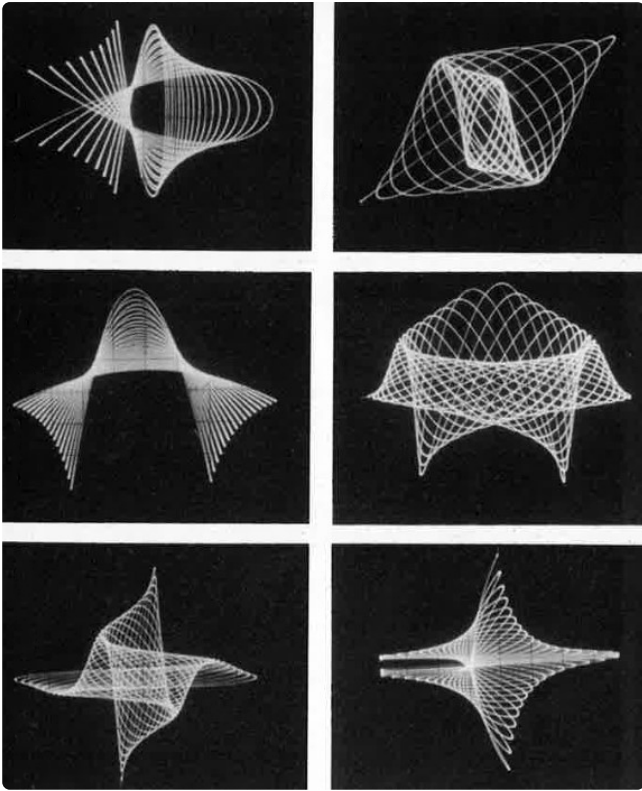
Figure 2: These photographs give some idea of the type of patterns that can be obtained from the spirator on an analog, CRT-based oscilloscope. (Scanned images from Elektor September 1978 magazine.)

PCB in the January 1979 edition [2]. Rumor has it that these animated, swirling, oscillographics weren't used scientifically or educationally as much as for fun and general distraction watching the workshop oscilloscope screen during lunchtime, possibly enjoying a good smoke. ◄

230356-01

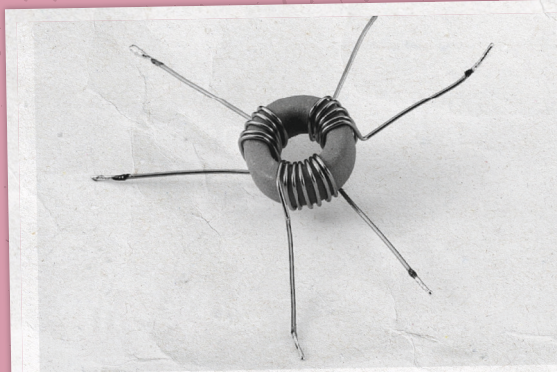🛒 **Related Product**

> **Elektor Archive 1974-2022 (USB Stick)**
> https://elektor.com/20372

WEB LINKS

[1] "Oscillographics," Elektor 9/1978: https://elektormagazine.com/magazine/elektor-197809/58217
[2] "Oscillographics on Board," Elektor 1/1979: https://elektormagazine.com/magazine/elektor-197901/58273

Quiz: Circuits from the Past #02

the signal source.

A drawback of the splitter is that the output voltage is 34% below the internationally specified level. However, most S/PDIF (Sony/Philips Digital Interface Format) inputs can cope with this perfectly well.

Place the transformer directly at the output of the signal source. The reason for this is that the input and output impedances of the transformer are not exactly 75 Ω. With the transformer directly at the source and provided the coaxial cable at the computer end is terminated correctly into 75 Ω, all will be well.

[994044]

T. Giesberts

The circuit in the diagram enables the digital audio output of, say, a compact-disc (CD) player to be linked to two different appliances simultaneously. It is, of course, considerably less expensive than the proprietary active splitters on the market.
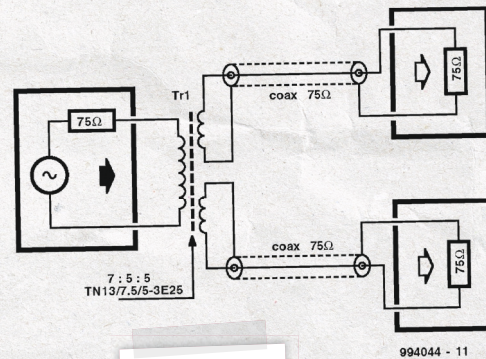
The circuit is in effect a small transformer that can be wound easily on a Philips Type TN13/7.5/5-3E25 toroidal core. The wire should be 0.5 mm dia. enamelled copper wire. The primary winding is seven turns and there are two secondary windings, each of five turns. The bandwidth of the transformer is 40 kHz to 16 MHz. When both outputs are loaded, there is a voltage of 0.33 V_{p-p} at each output. When one of the outputs is open-circuited, the volt-

Visit elektormagazine.com/quiz-23-2
to play now!

Take the quiz by August 31st, 2023.

The Elektor Store
Never expensive, always surprising

The Elektor Store has developed from the community store for Elektor's own products, such as books, magazines, kits and modules, into a mature web store that offers great value for surprising electronics. We offer the products that we ourselves are enthusiastic about or that we simply want to try out. If you have a nice suggestion, we are here: sale@elektor.com.

Elektor ±40 V Linear Voltage Regulator Kit
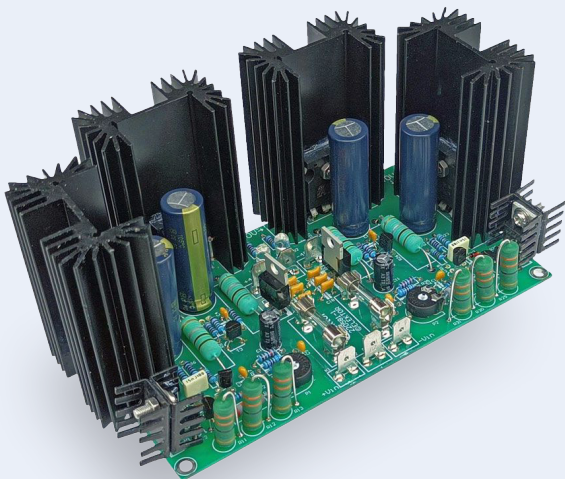
For those who object to a switch-mode power supply for the Fortissimo-100 Power Amplifier, this kit yields a linear, symmetrical, voltage regulator marked by low dropout voltage, high output current, and excellent stability – all obtained from discrete components.

Price: €69.95
Member Price: €62.96

🛒 **https://elektor.com/20439**

Elektor Cloc 2.0 Kit

Cloc is an easy to build alarm clock without SMT parts. It has a double 7-segment retro display with variable brightness. One display shows the current time, the other the alarm time. Cloc connects to a Wi-Fi network, so it can access a timeserver somewhere on the planet. It can connect using DHCP or with a fixed IP address.
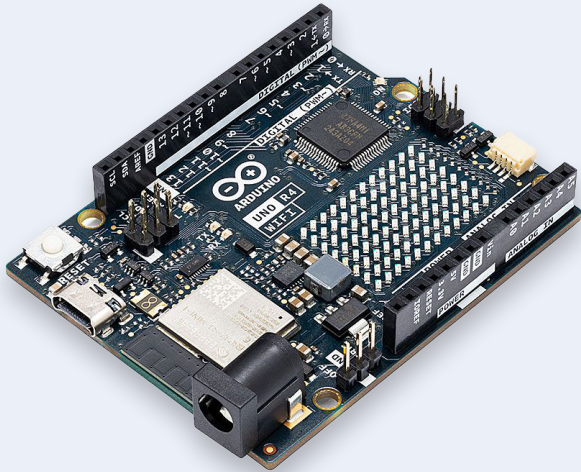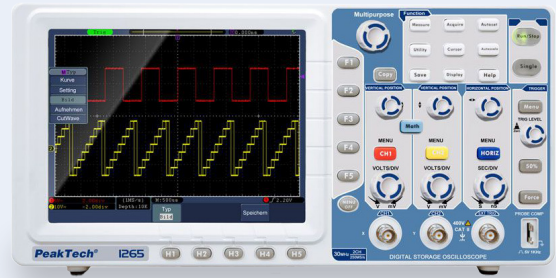

Price: €74.95
Member Price: €67.46

🛒 **https://elektor.com/20438**
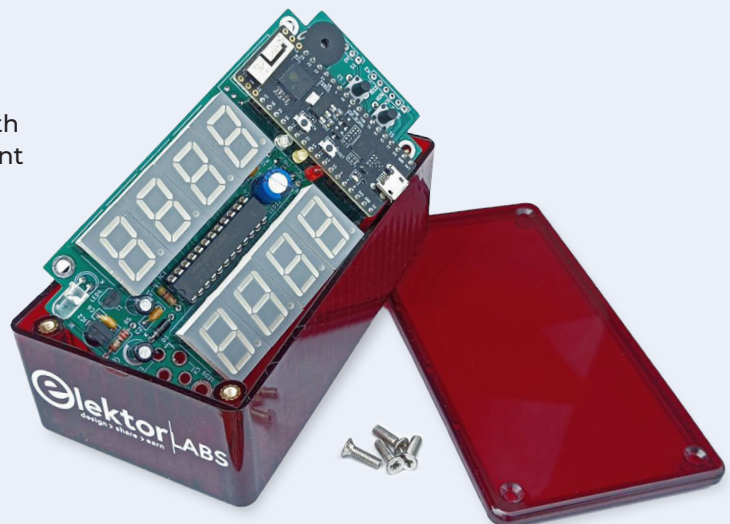
Arduino Uno R4 WiFi

Price: €29.95

[cart icon] **https://elektor.com/20528**

PeakTech 1265 2-ch Oscilloscope (30 MHz)

Price: €459.00
Member Price: €413.10

[cart icon] **https://elektor.com/20509**

EM900 Thermal Imaging Camera for Android (USB-C)

Price: €239.00
Member Price: €215.10

[cart icon] **https://elektor.com/20492**

YDLIDAR Tmini Pro – 360-degree Laser Range Scanner (12 m)

Price: €109.95
Special Price: €98.96

[cart icon] **https://elektor.com/20514**