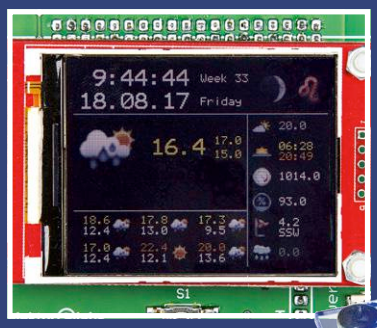
Edition 6/2017 | November & December 2017
www.elektormagazine.com

# elektor

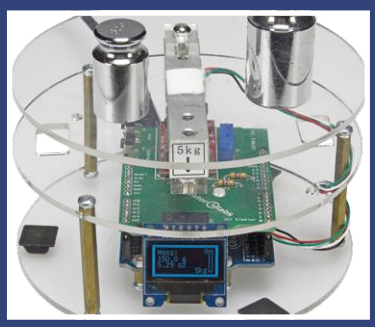
LEARN DESIGN SHARE




**Weather Display**
Visualize current weather data on an LC display



**Function Generator Shootout**
Three pro products compared




**Arduino'd Kitchen Scale**
Weigh up to 5 kilos




**Comparing Raspberries and Oranges** • Datasniffer 2.0 for Märklin Digital • **DIY Kitchen Scale** • DIY MC Element • **Elektor Labs Pipeline** • Elektor Store • **Err-lectronics** • Failed Fuse Indicator • **FPGA-DSP Board for Narrowband SDR** • Function Generator Shootout • **Hexadoku** • Homelab Helicopter • **Hoverboard: Under the Hood** • How to Charge a Battery • **Join the Elektor Video Olympics** • LEDitron Scoreboard • **Magnetic Memory** • Mini Oscilloscope • **Q & A** • Raspberry Pi Internet Radio • **Retronics: The Wang 320SE** • **Spiral MicroTesla** • **Spy Camera Detector** • Temi • **Tips and Tricks** • Traffic Signals for Model Railroads • **Ultimate Bugblaster** • Volume Control for RPi Audio DAC • **Weather Display** • Why We Need Hackers **… and more**

pico® Technology

NEW

# PicoScope® DeepMeasure™

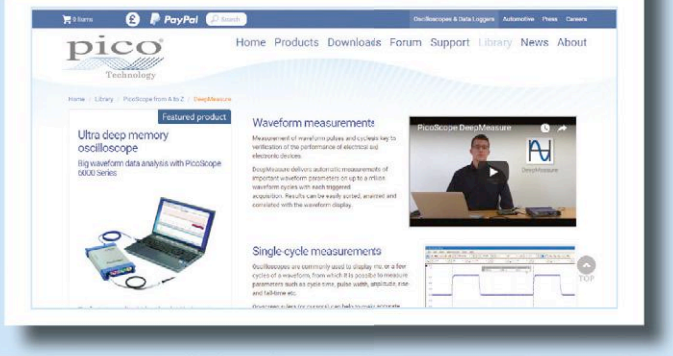
## One waveform, millions of measurements

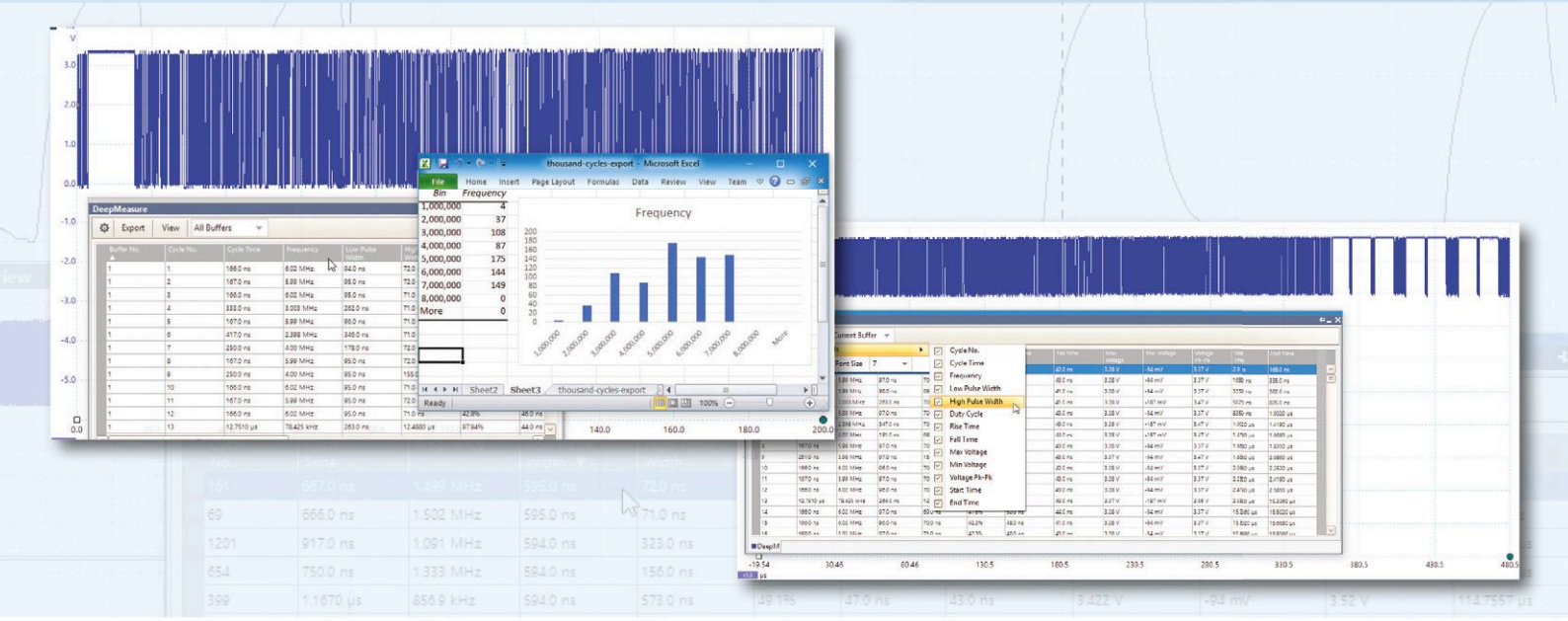
DeepMeasure displays the results of every waveform cycle captured with a PicoScope deep memory oscilloscope. With ten waveform parameters included in the tool, that's ten million measurements on every capture.

**Search** - Quickly find the fastest rise time or longest pulse hidden in waveforms of up to 100 million samples and jump straight to the correct cycle.

**Analyze** - Export captured measurement results to gain insight using data manipulation, mathematical and statistical functions in tools such as MATLAB and Excel.

To find out more view our video at
**www.picotech.com/PS525**







DeepMeasure is included as standard with PicoScope 3000, 4000, 5000 and 6000 Series oscilloscopes.

## www.picotech.com/PS525

Email: sales@picotech.com. Errors and omissions excepted. Please contact Pico Technology for the latest prices before ordering.

Elektor Magazine
**Edition 6/2017**
**Volume 43, No. 486**
**November & December 2017**

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

**www.elektor.com**
**www.elektormagazine.com**

Elektor Magazine, English edition
is published 6 times a year by

**Elektor International Media**
**78 York Street**
**London W1H 1DP**
**United Kingdom**
**Phone: (+44) (0)20 7692 8344**

Head Office:
**Elektor International Media b.v.**
**PO Box 11**
**NL-6114-ZG Susteren**
**The Netherlands**
**Phone: (+31) 46 4389444**
**Fax: (+31) 46 4370161**

Memberships:
**Please use London address**
**E-mail: service@elektor.com**
**www.elektor.com/memberships**

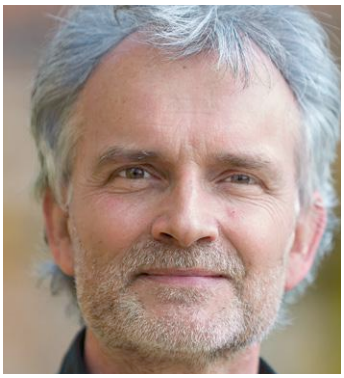
Advertising & Sponsoring:
**Margriet Debeij**
**Phone: +49 170 5505 396**
**E-mail: margriet.debeij@eimworld.com**

**Benoit Simoneau**
**Phone: +44 7891 920 370**
**E-mail: benoit.simoneau@eimworld.com**

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

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

© Elektor International Media b.v. 2017
www.eimworld.com
Printed in the Netherlands



# Hello Engineer [CR;LF]

After a period of noticeable decline there are now clear signs that reverse engineering as an educational help is on the up again albeit in different guise. Where a soldering iron, tweezers and a drawer full of colourful parts were traditionally used to create electronics on a workbench or repair it down to the component level, nowadays we swerve to FPGAs, Chinese PCBs, DHL, e-Bay and flashable firmware. “No hardware if it can’t talk software” is trendy, but essentially nothing new.

About 20 years ago some friends of mine bought a small cartload of boards and other parts off a computer manufacturer in bankruptcy. After assembling and powering up one of the (Z80-based) computers, light smoke started to curl from one of the plug-in boards. While several of the fellows frantically traced the power cord buried in a tangled heap of wires, one team member, fingers over the keyboard, asked quietly if anyone knew the address, in decimal or hexadecimal, of the system address used to monitor the supply voltage? — no answer, more hubbub — Could the address be XORed? Bitbanged? No? Why not? With rapid keystrokes he did manage to launch BASIC from ROM but before what must have been his fourth program line in the editor window, the screen went dead on him and the computer gave up the ghost. By smell only the hardware fans were able to track down an utterly wrong voltage regulator on one of the plug-in boards, and the repair took less than 15 minutes.

Today, circuit repair or reverse engineering is often said to be a waste of time owing to the complexity of modern equipment, the use of black-box technology and, paradoxically, the low cost of new items ordered from China. I could not disagree more. Defective or “surplus to requirements” as it may be, any piece of electronics made to good standards has some component or sub-assembly you can reuse for your creation, or dissect to grasp its operation. The odd smoke signal should not get in the way of success.

Jan Buiting, Editor-in-Chief

## The Circuit

| | |
|---|---|
| Editor-in-Chief: | Jan Buiting |
| Deputy Editors: | Thijs Beckers, Clemens Valens |
| Translators: | David Ashton, Jan Buiting, Martin Cooke, Ken Cox, Arthur deBeun, Andrew Emmerson, Tony Marsden, Mark Owen, Julian Rivers |
| Membership Manager: | Raoul Morreau |
| International Editorial Staff: | Thijs Beckers, Mariline Thiebaut-Brodier Denis Meyer, Jens Nickel |
| Laboratory Staff: | Ton Giesberts, Luc Lemmens, Clemens Valens, Jan Visser |
| Graphic Design & Prepress: | Giel Dols |
| Publisher: | Don Akkermans |

# This Edition

Volume 43 – Edition 6/2017
No. 486 November & December 2017




## Regulars

**12 Peculiar Parts, the series**
Magnetic Memory

**57 Elektor Labs Pipeline**

**67 Homelab Helicopter**

**101 Q & A**
The quality of audio connectors, part 2.

**114 Err-lectronics**
Corrections, Feedback and Updates on projects published

**122 Retronics: The Wang 320SE**
A Time-sharing Calculator (ca. 1970)

**126 Elektor Ethics**
Why we need hackers

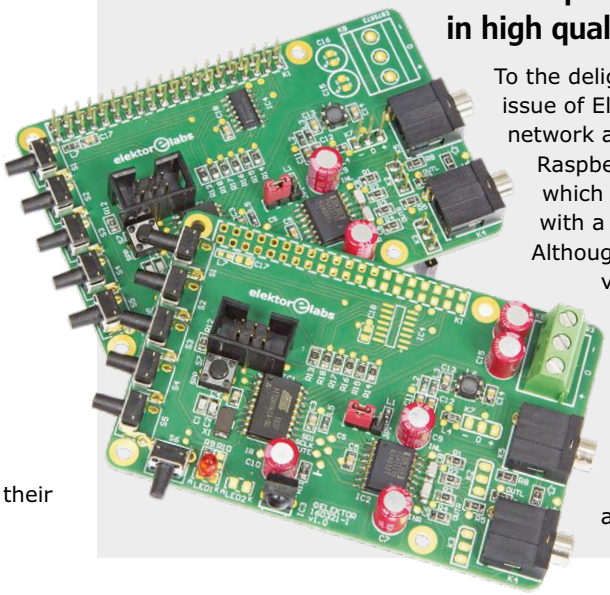
**128 Elektor Store**

**130 Hexadoku**
The original Elektorized Sudoku

## Volume Control for RPi Audio DAC

40

**loud and quiet in high quality**



To the delight of many, the previous issue of Elektor had a high-quality network audio player based on the Raspberry Pi/Volumio combination, which however was not provided with a 'real' volume control. Although it is possible to control the volume via Volumio (what's in a name), but this method seriously affects the high-end characteristics of the DAC for the worse. That is why we now describe a volume control especially developed for the network audio player.

## Features

**6 How to Charge a Battery**
Varieties of rechargeable batteries and their charging strategies

**34 Function Generator Shootout**
Three pro products compared

**89 Join the Elektor Video Olympics**
Frame it and win big!

**54 Mini Oscilloscope**
Display your signals without breaking the bank

**74 Tips and Tricks**
Alkaline batteries

**76 Comparing Raspberries and Oranges**
Orange Pi: a good-value allrounder

**108 Hoverboard: Under the Hood**
Reverse-engineering a fun product

## Projects

**14 Datasniffer 2.0 for Märklin Digital**
Model railway pulse packets with a PC

**18 FPGA-DSP Board for Narrowband SDR**
Part 3: microcontroller board

**26 Failed Fuse Indicator**
… needs to be simple, cheap, reliable and small

**28 Traffic Signals for Model Railroads**
Programming with ladder diagrams

122

## Retronics

**The Wang 320SE:**
a Time-sharing Calculator (ca. 1970)



# Weather Display

## Visualize current weather data on an LC display

elektor magazine


**90 Weather Display**
Visualize current weather data on an LC display

**98 Spy Camera Detector**
Reflections give away the position

**104 DIY Kitchen Scale**
Weigh up to 49.05 kg·m/s² with Arduino

**112 Raspberry Pi Internet Radio**
Tune in and chill out

**116 LEDitron Scoreboard**
With discrete 7 segment displays


## DIY Kitchen Scale

### Weigh up to 49.05 kg·m/s² with Arduino

The starting point for this article was a project posted on Elektor Labs describing the construction of a kitchen scale from an ATmega328 microcontroller, a load cell, a load cell interface board, an LCD, and some software. A bit of cutting and drilling of wood and metal was required to transform it all into a useable weighing scale.



104





**40 Volume Control for RPi Audio DAC**
Loud and quiet in high quality

**50 Temi**
A meteorological marionette

**58 Spiral MicroTesla**
Sparks from a USB charger

**70 DIY MC Element**
Impossible? Of course not!

**82 Ultimate Bugblaster**
No more flabby flyswatters


## Next Editions

### Elektor Magazine 1/2018

• CAN Bus for Multistorey Building Switching • DoubleS-PIder• Transistor Tester with Duo-LED • WiFi Repeater on Solar Power • OLED Displays • Measurement Data Acquisition via USB • Distortion Measurement at 10 MHz • Close-to-Universal IR Remote Control • PlatformIO, the Universal Programming Tool • Easy Parking System • Sandclock to Laserclock Conversion • and more.

Elektor Magazine edition 1/ 2018 covering January & February is published around 3 January 2018. Delivery of printed copies to Elektor Gold Members is subject to transport. Contents and article titles subject to change.

### Elektor Business Magazine 6/2017

**Elektor Business Magazine edition 6/2017 has a focus on Components & Manufacturing and is a special release for the productronica trade show in Munich on 14-17 November 2017.**

Among the contributors are companies and research institutions including Becktronics, Windriver, 4D Systems, Pixus, Visotech, premier Farnell, Botfactory, ROHM, Heraeus Sensor Technology, Weller Tools, Molex, and Microchip.

Elektor Business Magazine edition 6 /2017 is published around 8 November 2017 to Elektor Magazine Gold members in print, and Elektor Green members as a pdf download. The edition will also be available for purchase at www.elektormagazine.com.

# How to Charge a Battery

## Varieties of rechargeable batteries and their charging strategies



By **Dr Thomas Scherer** (Germany)

Knowledge among technical folk about rechargeable batteries and how to charge them properly can often be rather sketchy. And the situation among non-technical folk is even worse, with myths, hearsay and bizarre assumptions being perpetrated left, right and center. All this is despite the fact that rechargeable batteries are a part of many aspects of everyday life, and that most of us have the Internet at our fingertips as a potential source of accurate information. In this article we try to bring together the most important facts.

Fifty years ago the subject of rechargeable batteries did not have the importance it has today. Mostly they were found in cars (sometimes called 'accumulators') to operate the on-board electric starter motor to get an internal combustion engine going using stored electrical energy. But apart from that, in everyday life it was rare to encounter an electrically-operated device that was powered from rechargeable (or 'secondary') cells. For your transistor radio or flashlight, which were the main portable devices available at the time, you would use primary cells, usually based on a zinc-carbon chemistry, or, later, on the superior alkaline manganese dioxide chemistry.

Then things started to move quickly. The 1980s saw a rise in popularity of nickel cadmium (NiCd) rechargeable batteries in sizes compatible with primary cells (in particular in AA size),




Figure 1. Gel lead-acid battery rated at 12 V and 4.5 Ah. Batteries of this type are found in applications such as warning lights used around roadworks.



Figure 2. Inside the battery of my old Prius II hybrid vehicle: 28 modules each comprising six Panasonic prismatic NiCd cells rated at 1.2 V and 6.5 Ah wired in series, giving a total of 201.6 V. The pack can store 1.3 kWh and weighs in at 39 kg.

and, as you might expect, many circuit ideas and construction guides were published for NiCd chargers. Despite their slightly lower nominal terminal voltage it was easy to use these cells to replace disposable primary cells and so save a useful amount of cash. Come the 1990s, NiCd rechargeables started to be replaced by the easier-to-use nickel metal hydride (NiMH) cells, and since 2017 NiCd cells are practically banned in Europe, and can no longer be purchased except for special applications.

Also electronic devices were becoming more and more portable. On the heels of Sony's Walkman (for the benefit of younger readers: a portable cassette player) came MP3 players, laptops started to replace desktop PCs, telephones became mobile and suddenly everyone could have a smartphone to do their computing on the move. Most recently bicycles and, increasingly, cars have begun to go electric. All these devices, and many more besides, need batteries, and the range of different battery types available and their characteristics are as diverse as the range of applications.

**Types of rechargeable battery**

First some terminology. The word 'battery' comes to us via a French word meaning an array of artillery weapons, to which Benjamin Franklin compared his experimental array of Leyden jars. So strictly speaking a 'battery' should comprise more than one cell, but this distinction is only rarely made in everyday usage. The English term 'battery' covers both rechargeable and non-rechargeable types, the more precise terms 'primary cell', 'secondary cell' and 'accumulator' no longer being in common use. Now, with that out of the way, we can look at some of the more common battery types used today.

At over 160 years old, the **lead-acid battery** (see **Figure 1**) is certainly the oldest type of rechargeable battery that still finds practical use. The cells, which typically have a nominal voltage of 2 V, have been used in automobiles since around 1900. In that year the celebrated Lohner-Porsche Semper Vivus [1] was unveiled, the first ever hybrid car (are you listening, Toyota?). Another early application was in telephone and telegraph offices. The lead-acid battery is still difficult to beat in terms of price and robustness, and so, despite its heavy weight and other disadvantages it is still used to power the starter motor for internal combustion engines.

The **NiCd battery** (see the example from a hybrid car in **Figure 2**) also has a long history, going back over 100 years. In particular, in the 1980s, small NiCd cells were the most popular rechargeables for powering consumer electronic devices despite their nominal voltage of only 1.2 V. Their ability to deliver large currents made them the battery of choice for cordless tools and radio-controlled models. Also, Toyota Prius models up to and including the Prius III use this battery technology for their hybrid drive. The biggest disadvantages are poor environmental credentials (cadmium is toxic) and the notorious 'memory effect'.

The better is the enemy of the good: low-cost **NiMH batteries** (see **Figure 3**) came to replace NiCd batteries in consumer devices, having first been manufactured on an industrial scale some 35 years ago. Offering the same nominal voltage of 1.2 V they were an ideal substitute. They are also by and large more environmentally friendly, do not exhibit the memory effect, have reasonable energy density and are economical. On the other hand, they are a bit trickier to charge. Since about ten years ago NiMH batteries with very low self-discharge have been available.




Figure 3. NiMH pack from a handheld vacuum cleaner that has seen better days. The cells are welded together and (used to) provide 9.6 V with a capacity of 1.3 Ah.




Figure 4. Type 5s1p Lithium polymer battery rated at 18.5 V and 5 Ah. The maximum discharge current is 15 C, or 75 A, and the pack is designed for use in radio-controlled models. The small six-way connector allows a charge balancer to be connected.




Figure 5. This typical lithium battery in 18650 format is based on an $LiCoO_2$ chemistry. Such cells are becoming more popular in flashlights and especially in solar garden lights. The nominal voltage is given as 3.7 V rather than 3.6 V.



Figure 6. The 45-V LiFePO4 pack from my Segway clone: 14 type 38120 cells rated at 9 Ah and 10C. In the foreground is the charge balancer.

More recent rechargeable battery technologies are based around the element **lithium**. These chemistries allow the construction of lightweight batteries with a very high energy density, which are important factors in mobile applications such as laptops, tablets and smartphones as well as in electric vehicles. The three main types of practical importance are as follows: **LiPo** ('lithium polymer': see **Figure 4**) whose compactness makes them ideal for mobile applications; and **$LiCoO_2$** (see **Figure 5**) and **$LiFePO_4$** (see **Figure 6**) for electrical drives. Their nominal voltages, at 3.7 V for LiPo, 3.6 V for $LiCoO_2$, and 3.2 V for $LiFePO_4$, are significantly different from those of other chemistries. They do not exhibit the memory effect and have very low self-discharge. On the other hand they are sensitive to environmental conditions and there are strict rules to obey when charging and when discharging them. As well as a range of cylindrical formats lithium batteries are available in customized packs for mobile devices and in prismatic housings with higher capacity. It is interesting to note that the current range of Tesla electric vehicles use 'batteries' comprising many thousands of type 18650 cylindrical cells. The **RAM** (rechargeable alkaline manganese) battery (AA size shown in **Figure 7**) is certainly the newest kid on the block in the world of rechargeable batteries. These alkaline secondary cells have a nominal voltage of 1.5 V and so are ideal as direct replacements for zinc-carbon or alkaline manganese primary cells. However, they are only suitable for use in applications with low discharge current, must not under any circumstances be subjected to deep discharge, and require a special charger: they are decidedly not compatible with NiCd or NiMH chargers!

**Table 1** shows a list of the above rechargeable battery types with additional information such as the expected number of charge-discharge cycles (to the point where the cell's capacity falls to 75 % or 80 % of its original value) and the energy density (more precisely, the 'specific energy'), which is an important figure of merit in many applications. We will not look here at

other battery technologies that are not found in everyday commercial applications or at technologies still at the research stage, or at large-scale energy storage units such as flow batteries.

### Charging strategies

It is a good idea not just to pay attention to, but to follow to the letter the advice given by the a battery's manufacturer with regard to operating conditions (and in particular maximum discharge current) and to make sure that you respect the maximum permissible charge current. Nevertheless, for each type of battery there are some general properties that we can remark on, and we can compare the various allowable and optimized charging strategies. The most important considerations are the ability to withstand deep discharge and overcharging, the charge current limit and the details of the charging strategy; we will also look at aspects such as self-discharge rate and the degree to which the battery suffers from the memory effect.

**Lead-acid batteries**: In the case of sealed gel lead-acid batteries as well as the so-called 'maintenance-free' starter batteries it is important to ensure that they are not overcharged, as otherwise they can outgas. This can lead to a loss of electrolyte, which it is not a straightforward job to replace, and consequently to a shortening of service life and a reduction in capacity. In non-sealed lead-acid batteries, as used, for example, in fork-lift trucks, this is less of a problem as they can be topped up with distilled water. Lead-acid batteries should also not be subjected to deep discharge: the cell voltage should not be allowed to fall below 1.75 V. Relevant for storage is the self-discharge rate of 2% per month or more: for example, a starter battery on a petrol lawnmower may well not make it through the winter without some attention in January. The normal charging procedure is to begin with a constant current until the cell voltage reaches 2.35 V and then switch to constant voltage charging until a minimum current threshold is reached. Unless indicated otherwise the charge current should be at most 0.1 *C* (that is, 10% of the rated capacity in Ah/h). The current threshold when charging stops is typically 0.01 *C*.

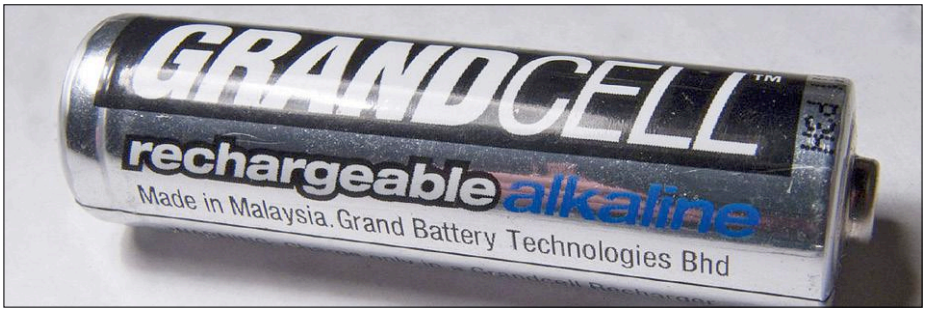



Figure 7. RAM battery cell based on an alkaline manganese chemistry in a standard AA package. These batteries must not be subjected to deep discharge and require a special charger.

In the case of a typical 12 V starter battery rated at 60 Ah this means that charging begins at a constant current of at most 6 A, and then, when the battery voltage reaches 14.1 V, the charging voltage should be kept constant. The current will then fall and when it reaches 0.6 A charging can stop. Continuing to charge at this point ('trickle charging') does not cause any problems. **Figure 8** shows a typical charging curve for this type of battery using the **CC/CV strategy** ('constant current/constant voltage').

A feature of lead-acid batteries is that over time they can lose capacity owing to a process called sulfation. To mitigate this special chargers are available that as well as providing a maintenance charging current, also provide regular millisecond-long pulses of current at over 100 A. This acts to prevent the formation of crystals in the battery and even help to break them down. A number of circuits along these lines have appeared in *Elektor* in the past.

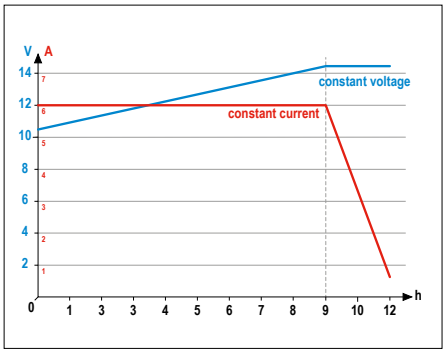



Figure 8. Idealized curves for CC/CV charging of a 12-V car battery rated at 60 Ah. The curves for lithium batteries are similar in appearance.

**Table 1. Varieties of rechargeable battery.**

| Type | Nominal voltage | Cycle life | Energy density | Applications |
|---|---|---|---|---|
| **$LiCoO_2$** | 3.6 V | 500 to 1000 | 180 Wh/kg | Drives, high current |
| **LiPo** | 3.7 V | 300 to 500 | 150 Wh/kg | Mobile devices |
| **$LiFePO_4$** | 3.2 V | 1000 to 5000 | 100 Wh/kg | Drives, high current |
| **NiCd** | 1.2 V | 600 to 1500 | 50 Wh/kg | General purpose, high current |
| **NiMH** | 1.2 V | 300 to 600 | 75 Wh/kg | General purpose |
| **Lead-acid** | 2.0 V | 300 to 500 | 35 Wh/kg | Emergency back-up supplies, starter batteries |
| **RAM** | 1.5 V | 50 to 500 | 50 Wh/kg | General purpose, low current |

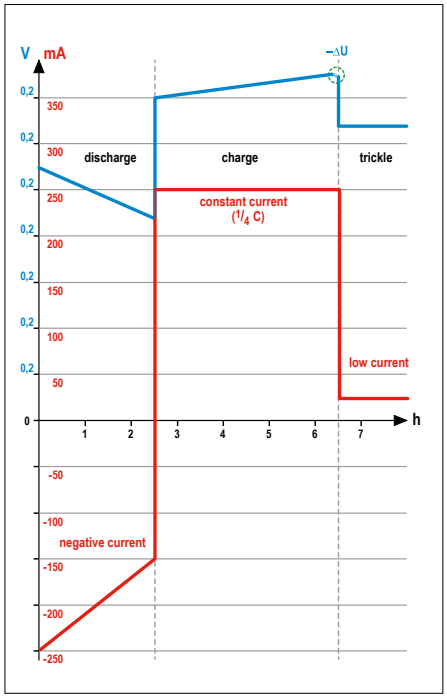



Figure 9. Idealized curves for delta-V charging of a NiCd cell at 0.25 C. In order to reduce the memory effect the charging cycle begins with a discharge phase.

**NiCd batteries**: Simple NiCd chargers have been around for a long time: these simply charge the battery at a low constant current. It is common to find 1-Ah AA-size cells or similar in devices such as electric pepper grinders or handheld vacuum cleaners with a plug-in charger that initially charges at 0.1 *C*, falling back to a continuous charge at perhaps 50 mA. The NiCd chemistry can withstand this overcharging for quite a while, but eventually the cells will be damaged. For this reason, and to improve charging time, a technique known as 'delta-V' charging is used. This technique exploits the fact that a fully-charged battery will turn excess charging power into heat, and the increase in cell temperature leads to a small drop in the cell voltage: see **Figure 9**. The charging circuit thus simply has to detect the point at which the voltage starts to drop.

This type of charger is also capable of fast charging. A charge current of 0.5 *C* or higher can easily be used with cells that are designed to support it. Note also that NiCd cells do not like to be deeply discharged, and should be recharged when their terminal voltage reaches 0.9 V.

One particular aspect of NiCd chemistry is the **memory effect** mentioned above. If a NiCd cell is repeatedly partially discharged and then fully recharged it starts to notice that only a fraction of its capacity is being used and then its maximum usable capacity starts to fall: this is a result of the formation of cadmium microcrystals. It is possible to reverse this process by repeatedly discharging the cell to below 0.9 V. The better microprocessor-controlled chargers take this into account and start each charging cycle by completely discharging the cell.

**NiMH batteries**: Most of what we said above for NiCd batteries goes equally well for their NiMH successors. NiMH batteries can also be charged using the delta-V method, although the voltage drop is less significant, especially at lower charging currents. For this reason higher charge currents are used, and this works well because the internal resistance of the NiMH battery is lower and hence it can be charged more quickly: up to 1 *C* is possible. In the interests of safety fast chargers also contain temperature monitors in case the fall in voltage is not detected. Since NiMH batteries do not suffer from the memory effect, there is no need to start each charge cycle with a discharge phase. NiMH-compatible chargers therefore allow the user to choose whether the discharge phase should be included in the cycle. Also, with the newer NiMH cells that have a very low self-discharge, it is possible to dispense completely with trickle charging at the end of the cycle. Looking at Figure 9, only the part of the cycle between the two vertical dotted lines is required. **Figure 10** shows a universal NiCd and NiMH quick charger for four AAA or AA cells.

**Lithium batteries**: All three types of lithium battery are usually charged using a CC/CV technique like the one used with lead-acid batteries. In comparison to Figure 8 the only thing that changes is the voltage scale; and because of the low self-discharge of less than 1 % per month no trickle charge is necessary at the end of the cycle. Depending on the application of the cells the charging current can be much higher than is possible with other chemistries. In the example of the Tesla cars, we have the ability to charge to 80 % of capacity in 20 minutes, which corresponds to a charging current of 2.7 *C*. On the other hand the final constant-voltage phase lasts relatively longer, and it takes another 40 minutes or more to reach full charge. The USB charger shown in **Figure 11** charges its 18650-size 2.4 Ah battery at a leisurely 0.5 A, or 0.2 *C*. But be careful: if a battery is deeply discharged it must be recharged initially at 0.5 C or less until its voltage reaches a certain threshold.

The memory effect and other such undesirable properties do not affect lithium cells. The only thing to watch is that you do not discharge them too deeply. $LiCoO_2$ cells in 18650 size (18 mm in diameter and 65 mm in length) are available in versions with integrated electronics (as shown in **Figure 12**) that protect against deep discharge, overcharging, and short-circuit. This is an ideal solution for flashlights and the like, at the cost of a slightly increased effective internal resistance. The main difference between the different types of lithium cell lies in the voltage characteristics: **Table 2** summarizes the relevant values. If the man-

**Table 2. Voltage characteristics of different types of lithium batteries.**

| Type | Nominal voltage | Charge termination | 0.5 C threshold | Discharge termination | Deep discharge |
|---|---|---|---|---|---|
| **$LiCoO_2$** | 3.6 V | >4.1 V | 2.9 V | 3.0 V | <2.5 V |
| **LiPo** | 3.7 V | >4.2 V | 3.0 V | 3.2 V | <2.6 V |
| **$LiFePO_4$** | 3.2 V | >3.7 V | 2.4 V | 2.5 V | <2.0 V |

ufacturer does not specify otherwise, you will not go far wrong if you assume that the maximum charging current is 0.5 *C* and that the constant voltage phase should be terminated when the current falls to 0.05 *C*. Terminating this phase is recommended as the cells should not be subjected to the maximum voltage for an extended period. So charging lithium cells is not too troublesome, as long as you can charge them individually. If you have several cells wired in series a charge balancing circuit is essential to ensure that the cell that has the lowest capacity is not overcharged and hence damaged. The LiPo pack in Figure 4 has a special connector for this purpose, and the homemade $LiFePO_4$ pack in Figure 6 includes a particularly high power charge balancer that is capable of diverting large currents away from the individual cells. The technology and circuitry in charge balancers can be complicated, and a full treatment would occupy an article in itself.

**RAM batteries**: Chargers for these batteries use a special technique. They start by charging at a relatively low constant current (in the region of 0.1 *C*). This current is interrupted periodically for a few milliseconds and during this time the no-load terminal voltage of the cell is measured. If this voltage exceeds 1.73 V then charging stops and is not resumed until the voltage falls below 1.69 V. It is very important that the cells are not overcharged and in particular that they are not subjected to deep discharge: the threshold for this is as high as 1.42 V per cell. For this reason RAM batteries are not likely to gain wide acceptance unless they can be made with built-in protection circuitry.

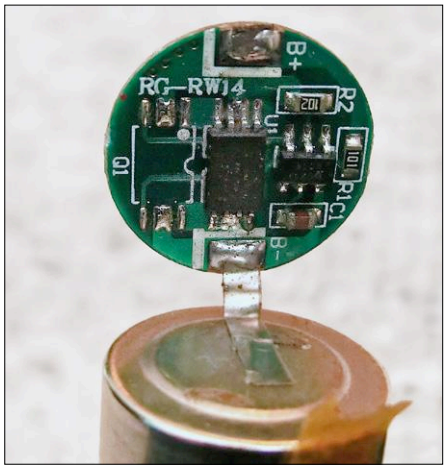



Figure 12. A circuit built into an 18650-size LiCoO2 cell protects it against deep discharge and overcharging.




Figure 10. Low-cost charger for 18650-size LiCoO2 cells with a USB connection and OLED display.




Figure 11. Intelligent quick-charger for four AA- or AAA-size NiCd or NiMH cells working at up to 0.8C. The device can even detect defective batteries. A press of the blue button discharges the cells before the charging cycle starts

**General remarks**

Since the advent of microcontrollers it has been easy to implement complex charging strategies. The information given in this article is in principle enough to let you develop your own charging circuit for any of the cell types we have looked at. One indispensable feature of a homebrew charging device is a cutoff timer. When all else fails and, for example because of a defective cell, the circuit does not correctly detect the point at which it should stop charging, it is a good final line of defense to stop after a certain amount of time has elapsed. This is particularly the case for lithium cells. If charging continues the excess electrical energy will be turned into heat, and potentially cause a fire, especially in LiPo cells: perhaps you remember the unfortunate case of the rechargeable batteries in the Boeing 787 Dreamliner. An allowance of 25 % on top of the calculated charging time is a reasonable place to start. A deluxe charger for lead-acid batteries would include an ambient temperature sensor to allow a compensation of around 3 mV/K to be applied to the charge termination voltage.

For special-purpose battery packs do-it-yourself is invariably the way to go. For packs of standard AA-, AAA- or 18650-size cells there are ready-made commercial alternatives that are manufactured in large quantities that will work out cheaper. ⏮

(160526)

## Web Link

[1] Lohner-Porsche: http://press.porsche.com/news/release.php?id=642


**About the Author**

Dr Thomas Scherer is an electronics engineer and qualified psychologist, and has been writing for *Elektor* for over 30 years. His specialties include analog electronics, microcontrollers and research in psychophysiology. He can almost always be found pottering around in his own spacious electronics lab, experimenting, assembling or soldering something or other.

# Magnetic Memory

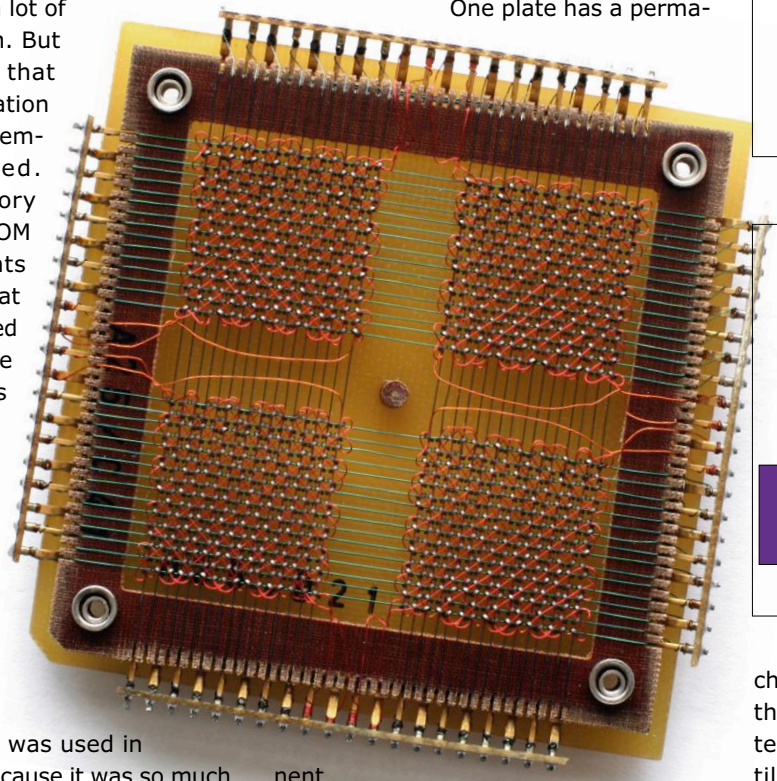
## Peculiar Parts, the series

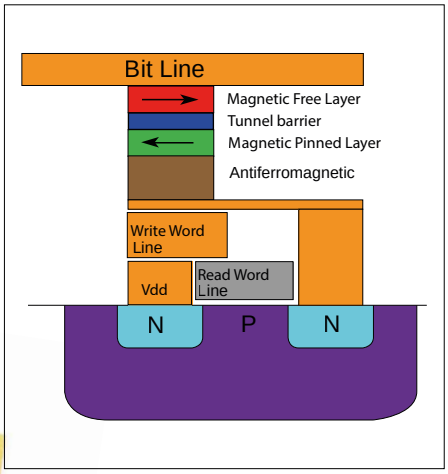
By **Neil Gruending** (Canada)

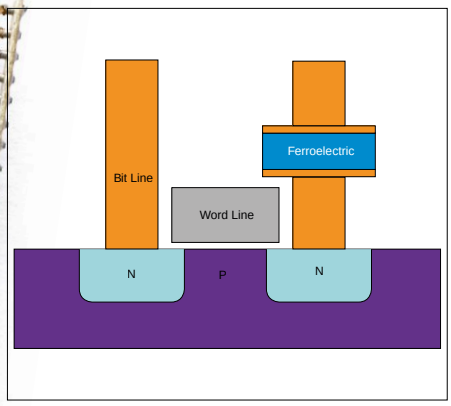
Today we have lots of options for storing information electronically like SRAM, DRAM, Flash and EEPROM. Random access memories (RAM) like SRAM and DRAM store information using capacitance which makes them very fast without a lot of extra complication. But their downside is that all of their information is lost when the memory isn't powered. Nonvolatile memory like Flash or EEPROM keep their contents while powered off at the expense of speed and requiring more complex access methods like memory paging. Enter magnetic memory which combines the qualities of both kinds of memories.

One of the first incarnations was magnetic core memory that was used in the 50s and 60s because it was so much faster than other storage methods like Williams tubes which were based on CRTs. The name comes from the fact that the memory was constructed using magnetic toroids called cores with wires threaded through them for read and write operations. Each toroid would be set to a 1 or 0 by controlling its magnetic direction although some extra circuitry was required because every read operation would clear the core's magnetization. This storage method proved to be very reliable and could even withstand an EMP pulse. But the downside of the toroids was that it was very difficult to manufacture and that made it relatively expensive so it was eventually superseded by SRAM which became available in the 60s. The Computer History Museum in Mountain View, CA, has a fine collection of magnetic-core memory devices on display, also check out their coasters with an MM image — buy one from the Gifts Shop (*Ed.*)

A modern version of magnetic memory is magnetoresistive RAM (MRAM) which has been developed over the last 30 years. Early versions used ferromagnetic plates with an insulating layer between them. One plate has a permanent magnetic field and the other plate's magnetization is writeable to store a bit. The logic state of the cell is then determined by measuring its resistance which changes depending on the magnetic orientation of the writeable plate to the permanent plate. Newer versions of MRAM use the spin transfer torque of electrons in their memory cells to lower their power consumption.






Another variation is ferroelectric RAM (FRAM) which has been developed in parallel with MRAM. Conventional DRAM memory uses one transistor and one capacitor per memory cell which can be made into a FRAM cell by adding a ferroelectric material in place of the regular dielectric. The ferroelectric material changes the normally linear behavior of the cell to one that has magnetic hysteresis which gives FRAM its nonvolatile properties. Writing data is pretty straightforward but reading a FRAM cell requires that a transistor put the cell into a known state and then the cell is monitored to see if the ferroelectric material causes the current to flow. This will also clear the cell so it needs to be rewritten. Both MRAM and FRAM are being produced today by various companies. MRAM devices tend to focus on density and speed whereas FRAM concentrates on low power. Either way, they are very interesting parts!

(160016)

Image credits (1, 2, 3): wikimedia

**Please contribute your Peculiar Parts article, email neil@gruending.net**

THE FUTURE OF STEAM EDUCATION

# Inspire Your Classroom

pi-top

www.pi-top.com

POWERED BY RASPBERRY PI

**pi-top** provides access to world-class educational resources for the Computer Science (CS) and STEAM (Science, Technolgy, Engineering, Art and Math) curriculum, designed to improve digital literacy with an all-inclusive simple to deliver solution.

OCR Oxford Cambridge and RSA

The only maker product endorsed by ocr - a leading uk awarding body




MODULAR ADD-ON BOARDS

## pi-top

The modular build-it-yourself Raspberry Pi powered laptop

- 10 HOUR BATTERY LIFE
- 13.3" HD SCREEN
- MODULAR COMPONENTS

## pi-topCEED

The modular all-in-one Raspberry Pi powered desktop

- ADJUSTABLE VIEWING ANGLES
- 14" HD SCREEN
- MODULAR COMPONENTS

The OCR* endorsed **pi-top**OS (Operating system) platform comes pre-installed on the SD card shipped with every unit. pi-topOS software suite lets you - browse the web, - check emails, - create and edit Microsoft Office compatible files It includes **pi-top**CODER with step-by-step activities guides and comes with the revolutionary educational game CEED**universe**.

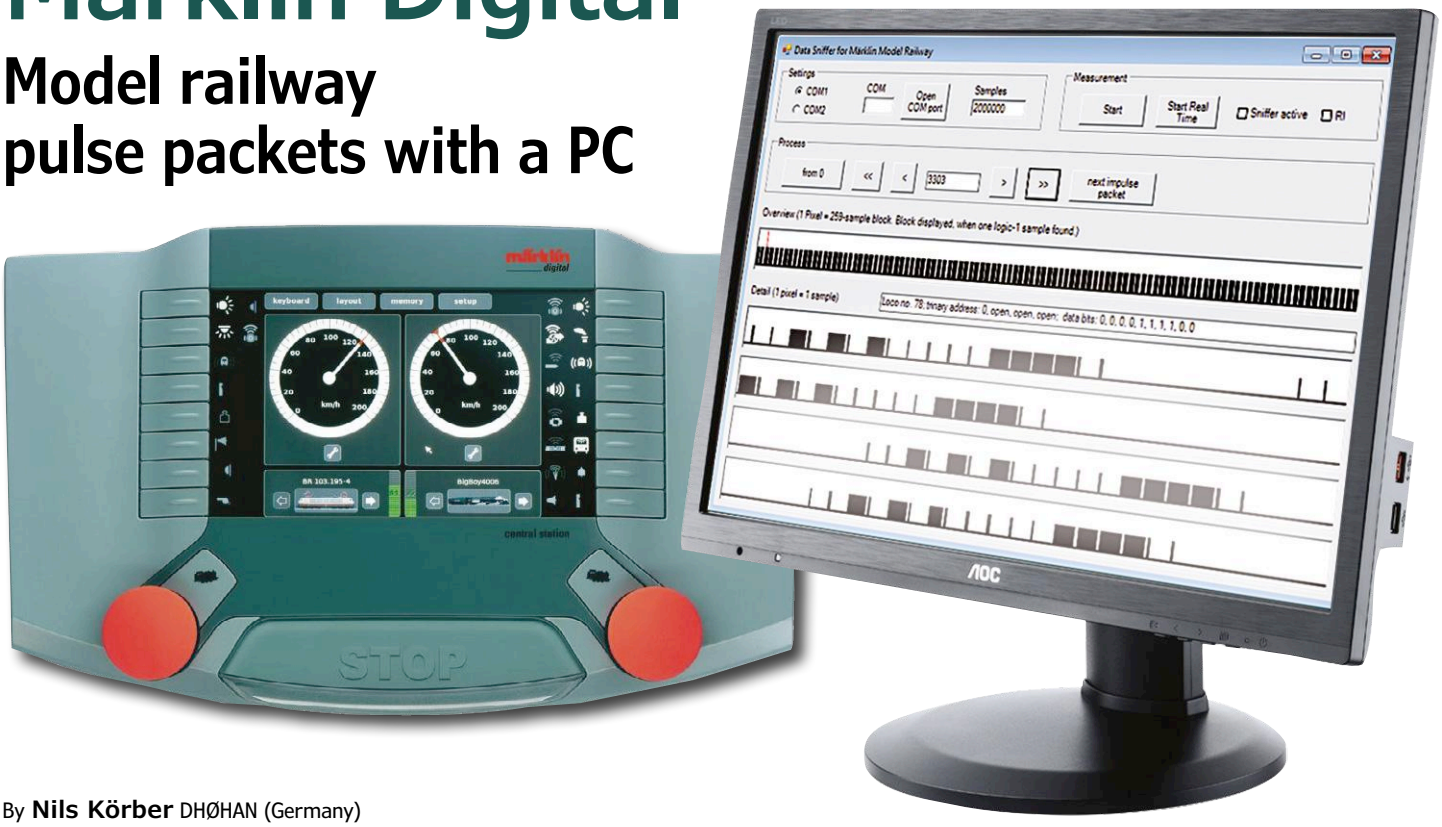
elektor

www.elektor.com



# Datasniffer 2.0 for Märklin Digital

## Model railway pulse packets with a PC




By **Nils Körber** DHØHAN (Germany)

It doesn't matter if they are generated by a DELTA-Control, Control Unit, Digital Railrunner, EEDTS-Pro or some other model railway controller, at the rails, all control signals in 'Motorola format' look pretty much alike. For testing and tracking down errors it helps if you can see these digital pulses live on a PC screen. In fact what you need is a Data Sniffer!

Those of you with a good memory may recognize the above header. The Data Sniffer (or Data Monitor) was first introduced in the May 2002 edition of Elektor. Thirteen years on, te software for that project is no longer compatible with the operating systems used on modern PCs. Welcome to the update!

### Some background and the concept

The control protocol used by Märklin digital model train controllers has already been pretty well covered (in the Elektor May 1999 edition for example) so there's no need to go over the theory in detail and we will just concern ourselves with the basics: The data is transmitted in the form of pulse packets which contain 18 individual pulses or 9 pulse pairs. The signal voltage levels on the rails are approximately –15 V and +15 V, the exact levels are dependent on the type of control unit you are using and the loading on the system which is dependant on the number of connected units.

Depending on circumstances a single pulse is interpreted as a bit while a pulse pair is interpreted as a 'trit' (trinary bit). Their logic value corresponds to the mark/space ratio of the pulses i.e. their pulse width. A typical data stream seen on the rails consists of bits and trits, for example:

**Bit '0'**
6 µs at +15 V, 182 µs at -15 V

**Bit '1'**
182 µs at +15 V, 26 µs at -15 V

**Trit '0'**
26 µs at +15 V, 182 µs at –15 V
26 µs at +15 V, 182 µs at –15 V



**Trit '1'**
182 µs at +15 V, 26 µs at –15 V
182 µs at +15 V, 26 µs at –15 V

**Trit 'open'**
182 µs at +15 V, 26 µs at –15 V
26 µs at +15 V, 182 µs at –15 V

In old Motorola format a complete pulse packet is made up of four trits which defines the locomotive address, one trit for the function and four trits for engine speed and direction of travel. The new Motorola format replaces the last four trits by eight bits giving information for speed, direction and four functions f1, f2, f3 and f4. The pulse packet therefore still retains the same length i.e. approximately 3.75 ms.

For the Data Sniffer to read the signal present on the rails from the digital train controller requires a level shifter circuit to reduce the signal level before it can be connected to the PC's serial interface. The data sniffer program samples the input signal at quite a high sample rate and stores the results during the measurement window which lasts for a few seconds. At the end of the measurement window the measured states are evaluated and then shown in detail on the PC display.

The measurement accuracy is dependant on the speed and consistency of the sampling rate. Here we must take into account some limitations (read more on this in 'Software'). Assuming we have a constant sampling rate we need to sample the waveform twice within the shortest pulse width of 26 µs to be sure that it has been detected. This gives us a minimum sampling rate of around 83.5 kHz. This rate should be achievable with a fairly up to date PC even with Windows running. The author's PC uses an Intel Pentium clocked at 2 GHz which, under Windows 7 is able to achieve a sample rate of 160 kHz. PCs with higher specs will get even better rates and more reliable measurements.

## The Hardware: one voltage divider

The control signals present on the rails are input to the PC via the serial interface port. The signal level can be in excess of 15 V so we use a simple voltage divider network (**Figure 1**) consisting of just two resistors valued at 4.7 kΩ. This reduces the signal levels so that they are compatible with the RS232C/V.28 standard.
For the serial interface the RI (Ring Indicator) and GND (Ground) connections are used. On a 25-way sub-D connector RI is connected to pin 22 and GND to Pin 7, for a 9-way sub-D-connector RI connects to pin 9 and GND to pin 5. The two-wire cable to the PC can be relatively long without causing any problems it is only important to observe the correct polarity of the two wires. On Märklin layouts the 'B' connection goes to the middle rail.

Note that with the PC connected, the power supply to the rails is no longer floating but is earthed via the GND connection at the PC. If this arrangement is not compatible with other circuitry you have installed on your layout it will be necessary to use a mains isolator with them in order to isolate them for the power controller.

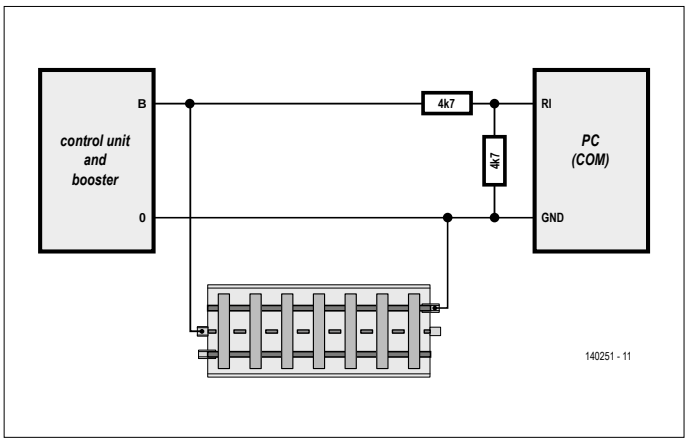



Figure 1. The train controller and PC working together.

## Software

The Data Sniffer program 2.0 was written using the free Microsoft Visual Basic 2008 Express. The Dynamic Link Library RSCOM.dll was used to provide access to the serial interface. It was written by Burkhard Kainka whose name regular readers will recognize. It can also be downloaded for free from the Internet.

## The user interface

The user interface shown in **Figure 2** is divided into five areas:

• Settings
When the port COM2 or COM1 is available the Data Sniffer will automatically select the port at start up. Alternatively you can use the *Open COM port* button to select the port. The number of samples used for the measurement can be defined here.

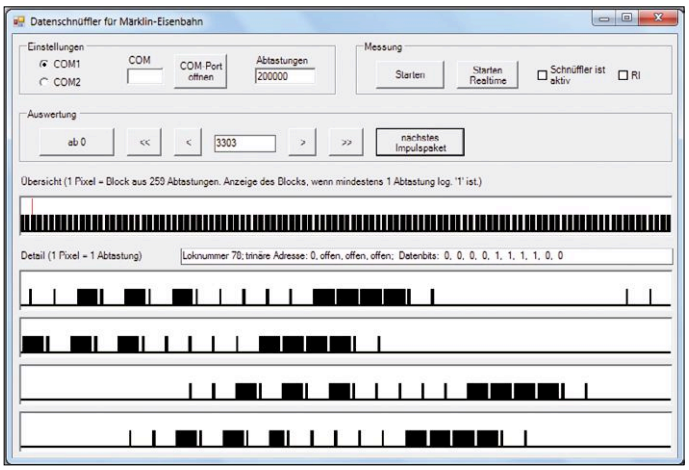



Figure 2. The Data Sniffer program user interface.



## Measurement accuracy stands or falls on the sampling rate

• Measurement
The *Start* button begins measurement. The *Start Real Time* also begins a measurement but this time the activity of other Windows processes are suppressed by masking their interrupts so that the measurement accuracy is improved with a more consistent sample rate. While the measurement is active the *Sniffer Active* box will display a tick. Between measurements the state of the RI input is sampled and displayed in the RI field.

• Process
The buttons available in this area control the detailed data display region. The number of the displayed packet is also shown here. A special feature is the *next pulse packet* button: This button positions the detail window over the next pulse packet (actually, just before it) and evaluates the information contained therein. The results are displayed as text in the Detail window above.
• Overview
The complete measurement is shown here with samples arranged in blocks. A block will be displayed when a 1 (corresponding to a 15 V level) is detected in the block. In addition a red cursor shows where the section shown in detail begins.

• Detail
The Detail display in the lower half of the display shows the sampled data values in four contiguous windows. The first displayed sample can be selected using buttons in the 'Process' field. There are always 3088 samples displayed.
Figure 2 shows a pulse packet in old Motorola-Format for Locomotive number 78. The data bits shown 0000111100 or data trits 00110 indicate that *function* is off and the *speed* value is *6*. In this case the displayed region begins with sample 3303 from a total of 200,000 samples.

### Features and system requirements

In order to get reliable measurements it will be necessary to select the *Start Realtime* option. This will mask disturbances from other Windows processes. During the time that measurements are made it is advisable to avoid other interactions with the PC also, such as moving the mouse.

The Data Sniffer program is available as a complete VB2008 setup. This means that there is no need to install Visual Basic in the PC; all you will need is Microsoft .NET-Framework which comes pre-installed in most modern computers. You will need RSCOM.dll [3], which you can store in the Data Sniffer folder or in your computer's Windows system folder (i.e. C:\Windows\System32). The DLL registration will usually be taken care of in the background but sometimes it will be necessary to register it manually.

Should you want to take a closer look at the program or make changes if necessary you can install Visual Basic 2008 Express Edition or higher and open the VB2008 project.
The source code and compiled program of the Data Sniffer are available for free download from Elektor's web site [4].
The Data Sniffer requires a PC with a free serial interface port, preferably COM1 or COM2. Alternatively you can use a standard 'USB 2.0 to RS232 serial' adapter cable of which there are numerous examples available on a popular online auction site for less than five Euros. A computer with a built-in serial interface has the advantage of a higher data sampling rate.
The system is compatible with Microsoft Windows XP, Vista, 7 or 8.

NB: Supply power for the model engines can either be floating (no connection to ground potential) or grounded via one of the outside rails (not the center rail!). ⏮

(140251)

### Web Links & Literature:


[1] Data Monitor for Märklin Model Railway Systems, Elektor May 2002

[2] Digital Control for Model Trains, Elektor May 1999

[3] RSCOM.dll: www.elektronik-labor.com/RS232/RSCOM.zip

[4] www.elektormagazine.com/140251

[5] The author's homepage: www.koerber-home.de




reichelt elektronik

Technology connects.

- More than 45 years of experience
- 24-hour shipping
- More than 85,000 products

# UPGRADE NOW FOR 3-WAY SAVINGS!

V-TAC® INNOVATIVE LED LIGHTING

1. Permanently low purchase price!
2. Reduced energy costs by up to 85 %!
3. Reduced maintenance costs due to long life!



## LED BULB

- 9 W, 806 lm
- 2,700 K
- 20,000 h

Per item only

Order no.: VT-7260

**1.28**

A+

## High-quality and low-priced LED technology from reichelt!

### LED SPOT



- 3 W, 210 lm
- 3,000 K
- 20,000 h

Order no.: VT-7126 | GU 10 | A+ | **0.52**

### FILAMENT LED



- 4 W, 400 lm
- 2,700 K
- 20,000 h

Order no.: VT-1885 | E-27 | A+ | **1.48**

### SPOT



- 10 W, 850 lm
- 4,500 K
- 20,000 h

Order no.: VT-5899 | A+ | **5.26**

### SENSOR SPOT



- 20 W, 1,700 lm
- 6,000 K
- 20,000 h

Order no.: VT-5802 | A+ | **15.09**

### TUB LIGHT



- 36 W, 2,800 lm
- 4,500 K
- 20,000 h

Order no.: VT-6200 | A+ | **15.09**

LED technology for indoors and outdoors – discover now and save:
**www.reichelt.co.uk/ledsavingweeks**

**SAVE NOW ▶**

Daily prices! Price as of: 22. 9. 2017 Prices in £ plus statutory VAT, plus shipping costs · reichelt elektronik, Elektronikring 1, 26452 Sande (Germany) Onlineshop languages:

PAYMENT METHODS: VISA MasterCard PayPal

SHOP CONVENIENTLY ONLINE!

**www.reichelt.co.uk**

ORDER HOTLINE: +44 203 808 95 25

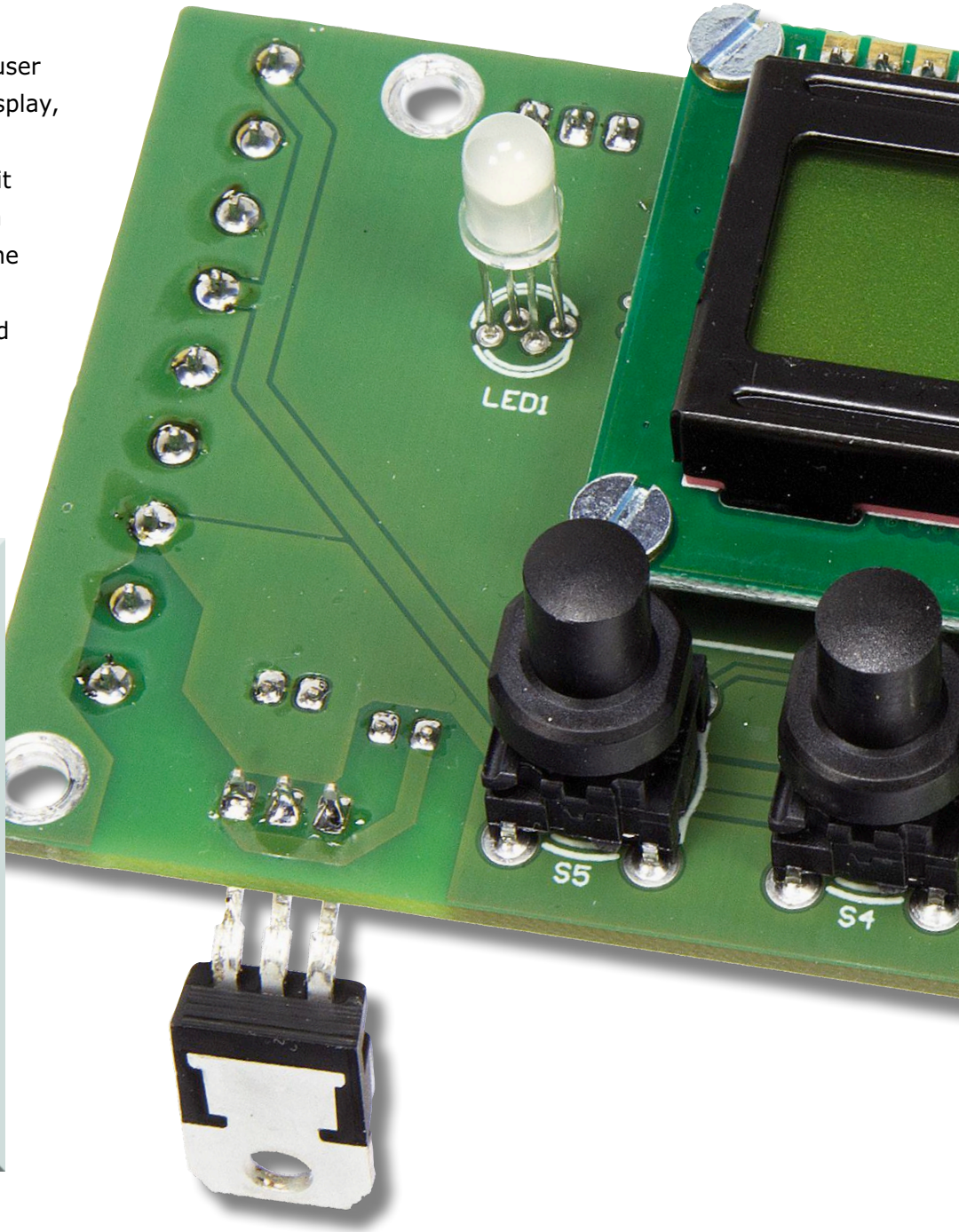
eLABS PROJECT

# FPGA-DSP Board for Narrowband SDR

## Part 3: microcontroller board

By **Daniel Uppström** (SM6VFZ, Sweden)
and **Ton Giesberts** (Elektor Labs)

Presented in this installment is a user interface (UI) or front end with display, knobs and buttons to control the FPGA-DSP radio. Based on an 8-bit ATmega128A microcontroller from Microchip the board can replace the Raspberry Pi that we used up to now. An audio amplifier is included too, making it possible to listen to the radio without ear-warming headphones.




**PROJECT INFORMATION**

FPGA DSP SDR
Radio Ham Radio RF
Microcontroller

entry level
intermediate level
➡ expert level

4 hours approx.

SMD soldering tools,
drill press,
metal work tools

£90 / €100 / $110 approx.

## Features

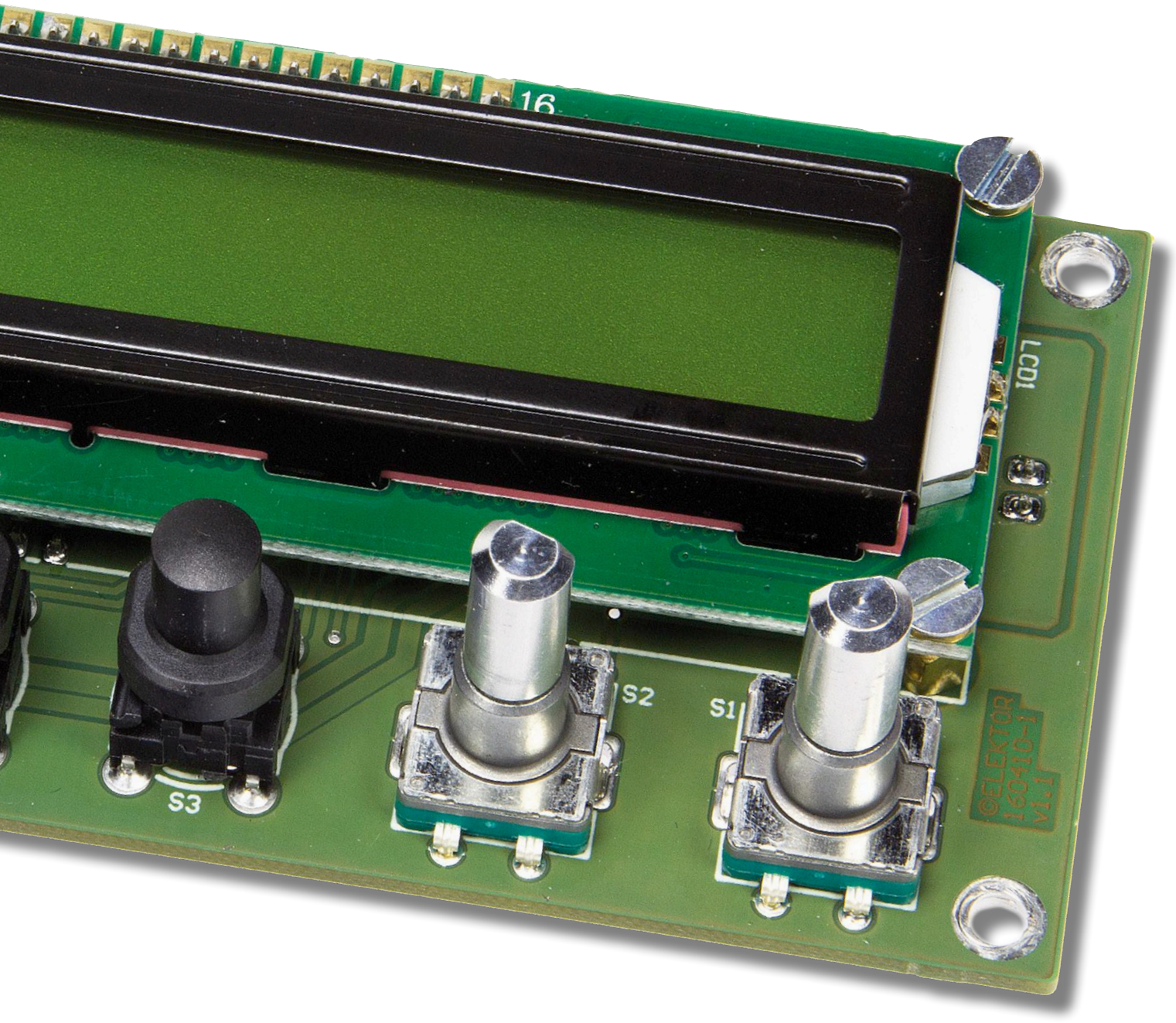
- 2×16 LCD + RGB LED
- 2 rotary encoders, 5 pushbuttons
- Clarity control
- Squelch
- Multifunctional
- On-board audio amplifier

The microcontroller board described in this article was specifically designed as a controller for the FPGA-DSP radio, as well as an alternative to the Raspberry-Pi-based controller. Where an RPi requires additional hardware like a monitor, keyboard and mouse, this MCU board comes with a 2×16 LCD, three pushbuttons, two rotary encoders (with pushbutton), a 2-watt audio power amplifier, a 5-volt voltage regulator and everything else needed to create a stand-alone radio. An (incomplete) overview of the board is shown in **Figure 1**. The versatility of this MCU board allows it to double for many other applications.
**Figure 2** shows the complete circuit diagram of the MCU board.

### I²C

The MCU board and the FPGA-DSP board communicate over an I²C bus, exposed through K3. When the controller starts up, a few settings are sent to the FPGA to define its basic operation. Further settings like frequency, audio volume and mode of operation (AM, SSB, CW, etc.) can then be controlled by the user. The controller continuously reads status information from the FPGA. This data incudes things like received signal strength (RSSI) and whether the radio is in receive or transmit mode.

I²C is simple, requires few wires, and allows adding more nodes to the same bus. The SDA and SCL ports are of the open-collector/drain type and resistors R2 and R3 pull them up to 5 V. This might seem incompatible with the 3.3 V used for the FPGA I/O's, but the resistors will allow only a few milliampères to flow and the body diodes in the FPGA will ensure non-destructive voltages.
Resistors R4 and R5 and the two ferrite beads L4 & L5 attenuate any high-frequency noise at the interface.
Solder jumper JP1 on the FPGA-DSP board must be left open to select I²C mode (instead of UART mode).



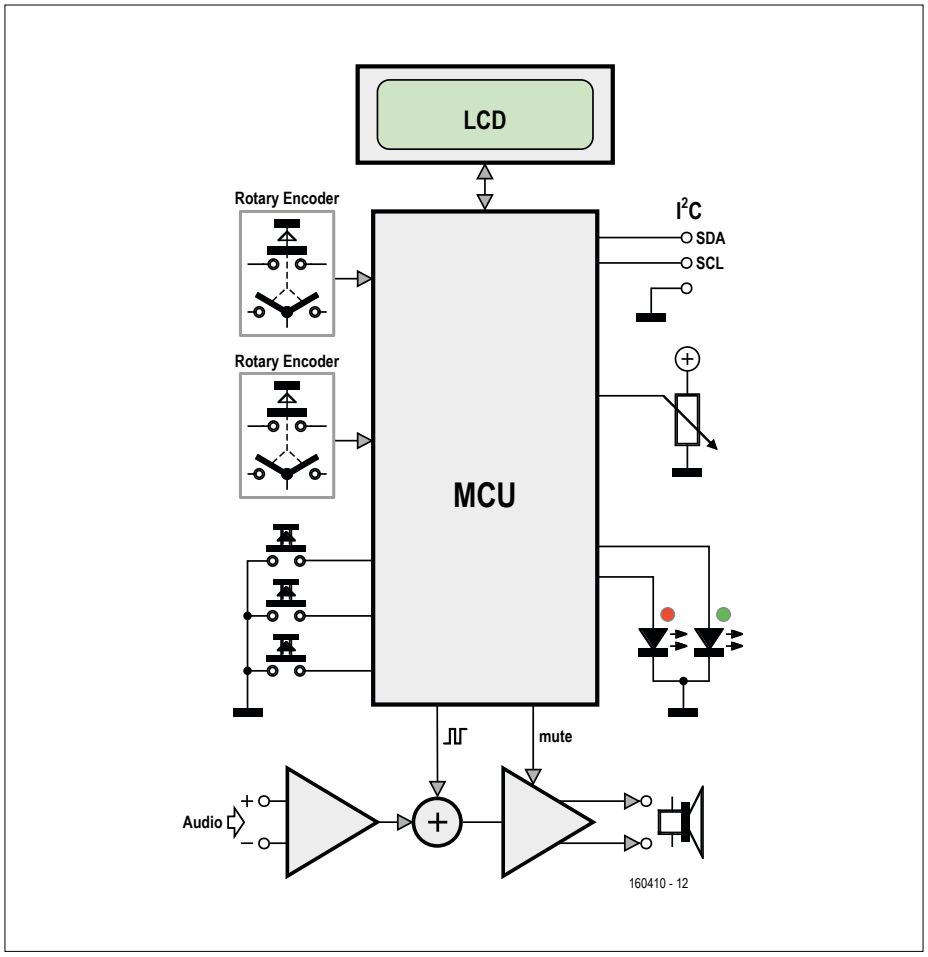



Figure 1. Even though this drawing only shows the main controls and peripherals of the MCU board, it is clear that this board is at ease in many other applications.

### Knobs & buttons

Pushbuttons S3, S4 and S5, together with encoders S1 and S2 and the LCD form the main user interface or UI of the radio.

Rotary encoder S1 is intended for tuning the radio's operating frequency. To go faster, press it while turning.

The second rotary encoder, S2, sets the volume. Push and rotate it to set the squelch level. This will mute the radio's audio output when the signal strength drops below a certain level.

Pushbutton S5 allows to quickly step through frequency bands, whereas S4 toggles between the modes AM, LSB, USB, CW and CWN (narrow CW filter).

Pushbutton S3 is intended for accessing a configuration menu, but at the time of writing this function wasn't implemented in the firmware.

The display, LCD1, lets the user see what the current settings are. It is a standard HD44780-compatible alphanumerical LCD with two lines of 16 characters; its contrast is set by trimmer P2. The LCD is used in full 8-bit mode to get the most out of it.

### Big-ass tuning knob

When doing serious radio amateur work with lots of tuning up and down the band one quickly gets fed up with a cheap mechanical encoder like S1. To make life more comfortable it is therefore possible to connect a better — but more expensive — optical encoder, with a nice, big knob to get that smooth feeling associated with expensive high-end shortwave transceivers. Not only do these encoders spin better, we also implemented an accelerator for it that makes fast rotations result in larger frequency steps. The Bourns ENA1J-B28-L00128L is a good example of a reasonably priced optical encoder compatible with the MCU board's firmware. Connect such an encoder to K1.

### Clarifier

When using the radio for transmitting, it is often desirable to fine-tune the frequency of reception without altering the transmission frequency. This is known as *clarifying* and is achieved with potentiometer P1. When it deviates from its middle position an offset is added to the reception frequency, indicated on the display with a plus or minus sign.

### RGB LED

Today no device is complete without an RGB LED, and so there is one on the MCU board too: LED1. However, here it is not used to create a colorful indicator, but simply to show the main operating modes:

1. Transmit (red)
2. Receive (green)
3. Off (blue)
4. Mute (black)

It may seem strange that Off is considered an operating mode, but in this case it is. In fact, LED1 will light blue when the radio is turned off using S6, while the 5 V supply remains available — a kind of stand-by mode. When the radio is on, the LED will be green in receive mode and red otherwise. If the squelch is set and the audio signal is muted, the LED will not light up at all.

### Power supply

If a voltage source capable of providing at least 7 V is available, it is possible to mount voltage regulator IC5 to obtain the precious 5 V needed for the rest of the board. If this is done, care should be taken to allow IC5 to dissipate its heat. The source connected to K7 can be turned on and off by S6 mentioned before provided it is of the double-pole variety. K5 and K6 can be used to either supply the board with 5 V or to distribute the 5 V to other boards. Note that K5 is ahead of S6, K6 is behind it, meaning that the latter is directly connected to the rest of the circuit.

### Sound & music

The FPGA-DSP board contains an audio CODEC with a class-AB speaker output capable of delivering a maximum output power of 250 mW into an 8-Ω load. Because small speakers usually have low efficiency and 250 mW is often only enough for headphones, a small amplifier was added to the MCU board to provide more power. A class-AB amplifier was used instead of class-D to avoid additional RF noise.

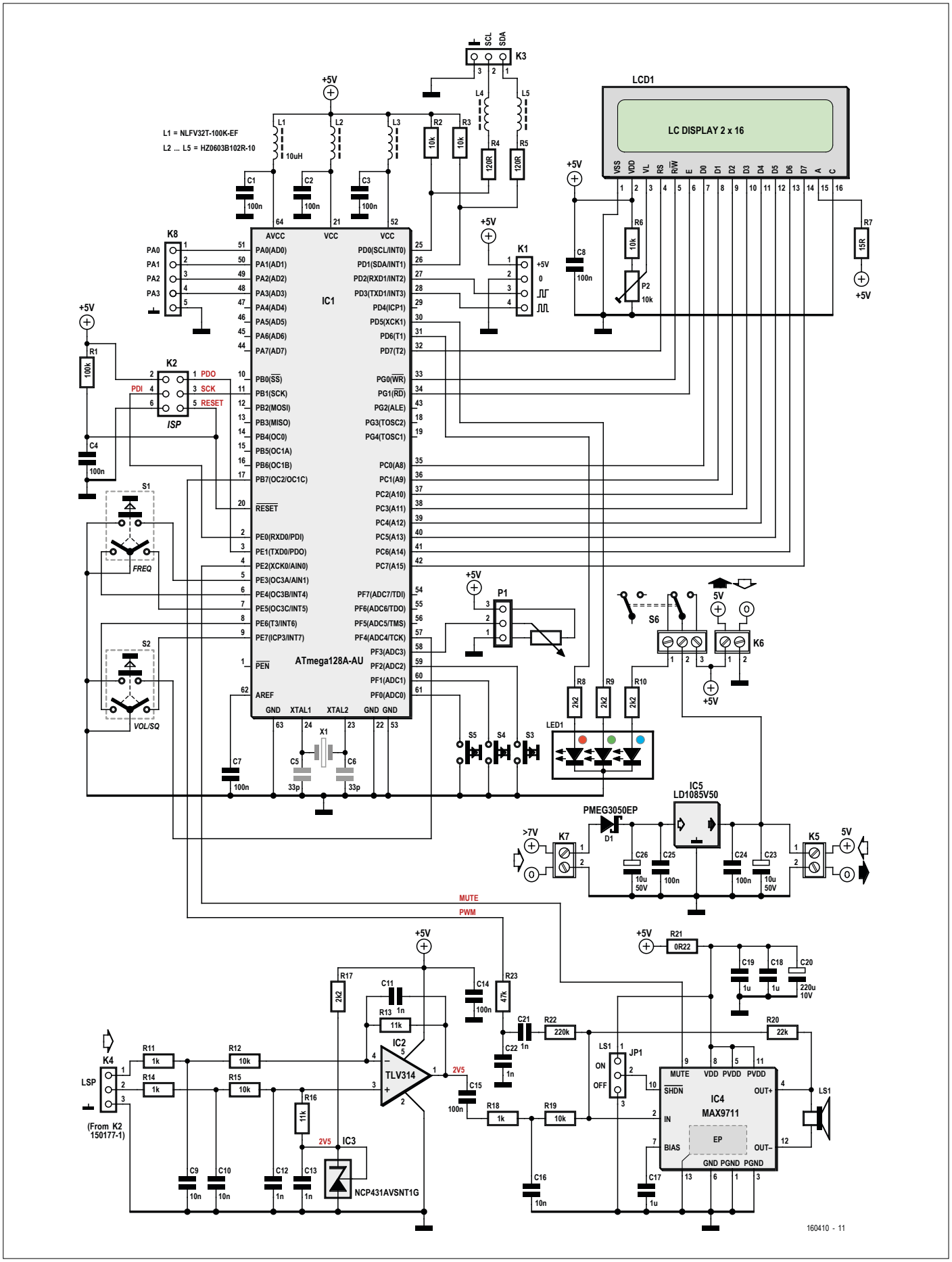



Figure 2. The lower part of this schematic shows the single-channel audio amplifier of the MCU board. IC2 converts the symmetric input signal to asymmetric, whereas IC4 does the inverse. The MCU's PWM output is mixed with the input signal.

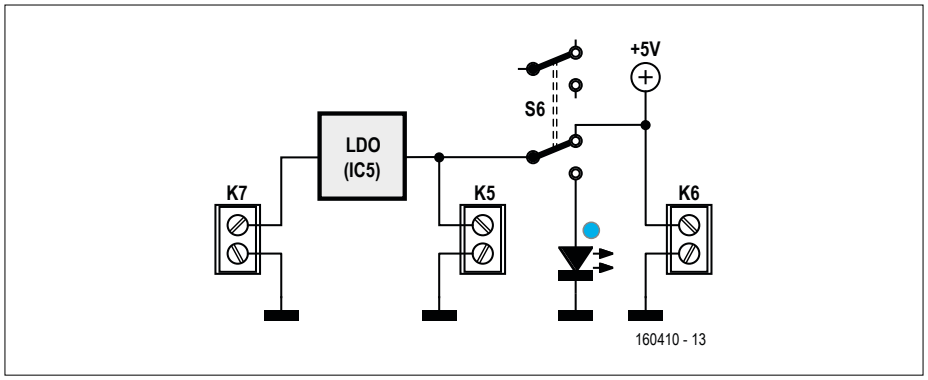



Figure 3. Close-up of the power connections. K7 is an input, K5 is either a 5 V input or output while K6 is a (switched) 5 V output. The second pole of S6 is available for switching some other device or signal, like the source connected to K7.

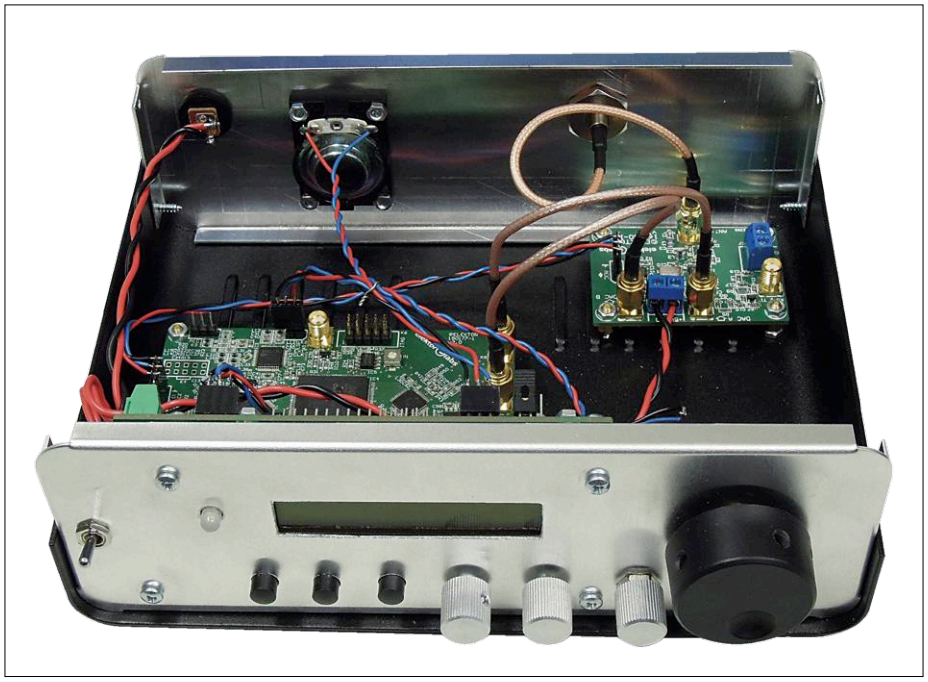


Figure 4. A look inside our finished prototype. Notice how the antenna input cable loops to accommodate its slightly too great length.

For our AF amplifier the MAX9711 was chosen (IC4). It has a differential output stage and its measured output power is close to 2 watts into a 4-Ω load with only a single 5-V power supply. Its input is single ended whereas the FPGA-DSP board has a differential audio output. For this reason IC2 makes the audio signal single-ended before sending it to IC4. It also acts as a low-pass filter with a cutoff frequency of 8.6 kHz, defined by C9 to C12 together with C16. This bandwidth is more than enough for voice; the bandwidth can be extended by decreasing the values of these capacitors. The resistors and capacitors at IC2's input (R11 to R16 and C9 to C12) should have 1% tolerance or better to get the highest common mode suppression. IC3, a 2.5-V reference device, lifts the non-inverting input of IC2 to half the supply voltage.
The feedback of the MAX9711 is inverting allowing the power amplifier to be easily used as an adder for different input signals. This gives the processor a means to generate blips and beeps — as feedback to the user — through a simple low- and high-pass filter added for this purpose, R23/C22 and R22/C21 respectively.
The processor can also mute the audio amplifier, because its port PE2 is connected to the mute input of IC4.
Be careful when connecting a speaker as both output pins carry a signal. Accidentally connecting one of them to ground equals a short circuit and will probably damage the MAX9711 (we didn't try to find out).
To connect K4 to output K2 of the FPGA-DSP board twist/braid the three wires; at the FPGA-DSP board side the ground terminal is not connected.
A very detailed description of the audio amplifier can be found at [4].

**And finally, the processor...**

Since most of the circuitry has been described by now, there is not much left to say about the microcontroller, the brains and the heart of this MCU board. Quartz crystal X1, together with its load capacitors C5 and C6, can optionally be mounted if an extra-stable clock is required, but for the current firmware it is sufficient to use the RC oscillator you get for free with your ATmega128A. This may also produce less radiation since no clock signals leave the processor (through a pin).
The MCU is programed by a standard AVR ISP adapter connected at K2, while connector K8 exposes some leftover GPIO ports for experimentation and extension purposes.

**Some words about the software**

The firmware for the MCU board is open source, and can be downloaded from the project web page [1][4], check [6] for the most-recent version. By default the firmware is configured for shortwave amateur radio use with the simple radio board presented in a previous installment, but it can easily be adapted for other frequency ranges or for something else altogether.
Except for the LCD driver, everything is contained in the file main.c while some configuration parameters are available in the file build_settings.h. The code is straightforward and easy to read.
Two timers assure a smooth user experience. Every 100 ms the Band (S5) and Mode (S4) buttons are read together with the clarity control P1, and the RSSI value is updated.
The rotary encoders generate interrupts when they are turned and they are serviced when this happens. A 10-ms timer controls their maximum speed and provides contact debouncing. The tune encoder S1 (ROT1 in the software) functions in parallel with the optional optical

encoder connected to K1 (ROT2 in the software). Since the optical encoder does not have mechanical contacts, it does not need to be debounced, which makes it much more responsive. Fast spinning can now be detected too, and will result in larger frequency steps.

### Putting it all together

Since the purpose of the MCU board is to provide a means of building a stand-alone radio, let's have a look at the mechanical side of how this can be done. A suitable and fairly cheap enclosure that fits the FPGA-DSP radio was found at Conrad Electronics, reference GSS03. Its dimensions are 200×150×70 mm. It's a very simple design with identical top and bottom shells made of robust 1.5 mm steel with a few ventilation slots. The top and bottom are screwed to two identical 1 mm thick aluminum front and back plates with four self-tapping screws. There are no additional board support rails. All the boards fit easily in this enclosure. A bigger version of the enclosure might be interesting if future extensions are a possibility. Dimensions of the GSS04 enclosure are 250×200×70 mm, making it possible to mount the speaker on the front.

### Cabling

Although a bit too long, two commercially available 15 cm SMA male-to-male RG405 cables can be used to easily connect the RF board to the DAC outputs of the FPGA-DSP board without requiring special tools to attach SMA connectors to shielded cables. Such cables can be found online, for instance on eBay.

For an antenna input for the RF board we used a 15 cm long RG316 BNC female-to-male SMA cable. Since this cable is also longer than strictly needed, we allowed it to loop to the back of the enclosure (**Figure 4**). An external antenna can now be connected through a BNC cable.

The MCU board is mounted on the front panel and placed just far enough to the right to leave sufficient space for the power switch. Potentiometer P1 was placed next to the optical rotary encoder at a distance similar to the one separating S1 from S2.

Use pinheader sockets and thin stranded wires to connect the potentiometer and optical encoder to the corresponding pinheaders on the MCU board (P1 and K1 respectively). Be careful when wiring the optical encoder as connecting its power supply the wrong way around will probably permanently damage it. If the encoder seems to turn in the wrong direction, swap the two data lines.

The connections of the I²C-bus and speaker signals (K3 and K4 of the MCU board to K7 and K2 of the FPGA-DSP board respectively) and the output of the RF board (K4) to the input of the FPGA-DSP board (K1) is also done using sockets and thin stranded wires. Twist or braid all wires. Using sockets for the connections will make potential future changes and/or additions much easier to execute, but feel free to solder all wires directly to the pin headers.

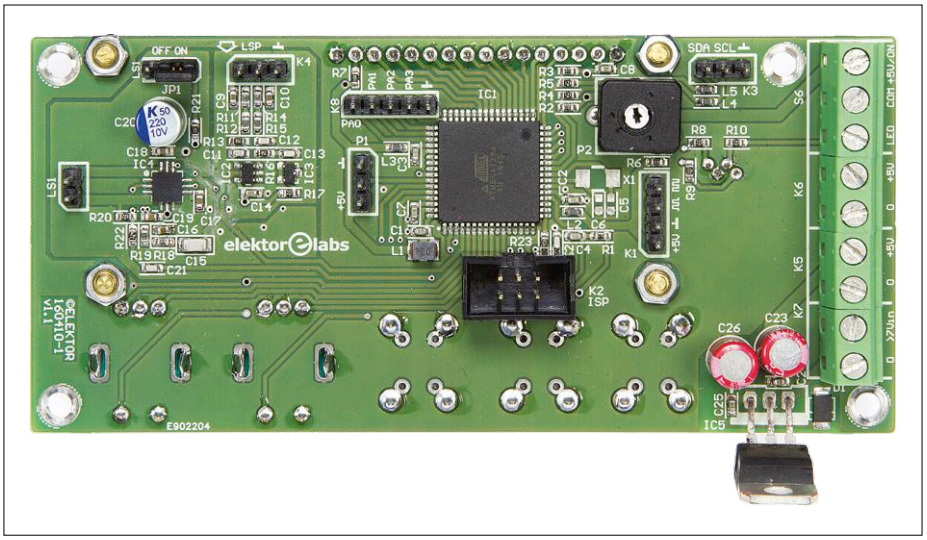



Figure 5. The pins of voltage regulator IC5 are folded so that the IC can be easily fixed to the bottom of the enclosure.

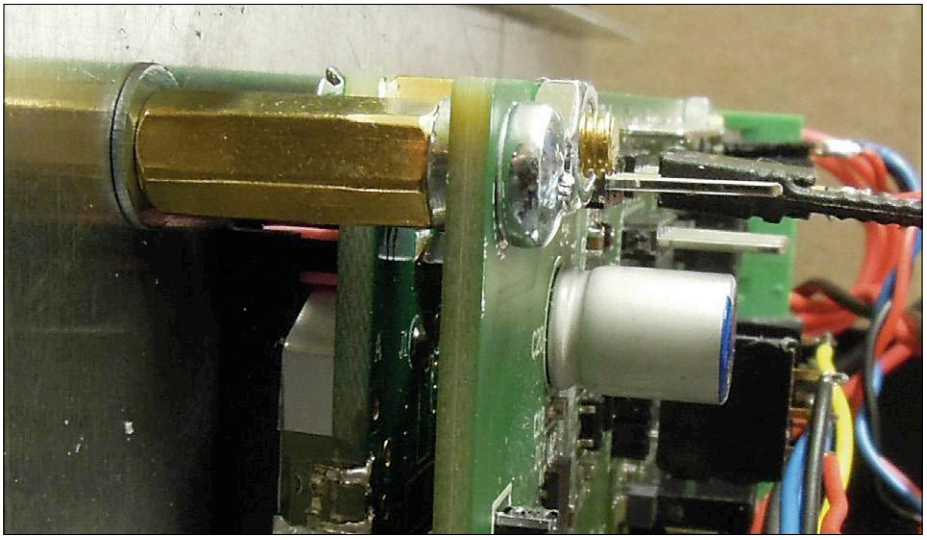


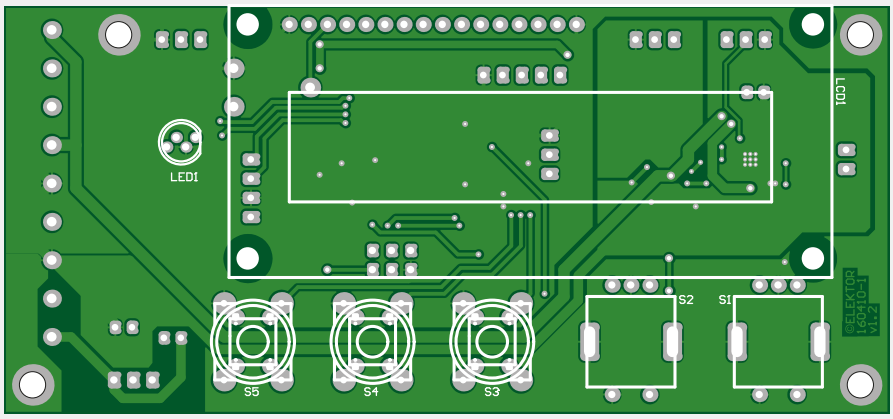
Figure 6. Washers and stand-offs were used to precisely position the MCU board behind the enclosure's front and so create a tight fit for the LCD.

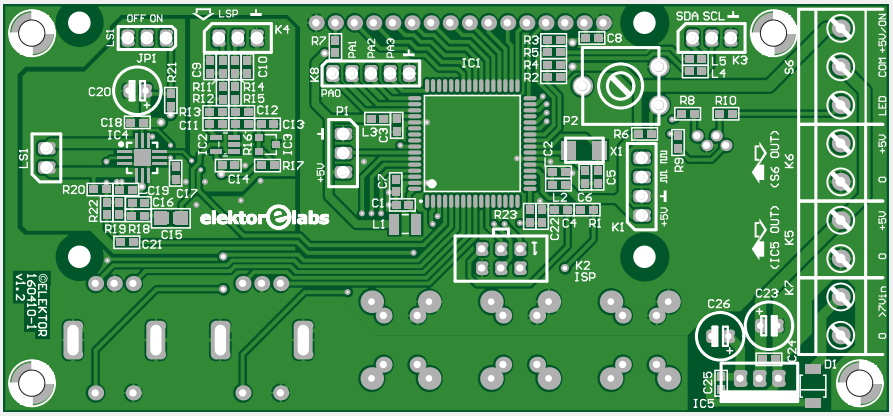
### Power supply, part two

In our prototype we used the 5-volt regulator IC5 on the MCU board (**Figure 5**). It is fixed to the bottom of the enclosure with an insulating bush and washer. The bottom of the enclosure is made of steel and conducts heat less efficiently than aluminum. The higher the input voltage, the higher the power loss in IC5, and you might consider adding a real heat sink for it if the input voltage will be more than 8 V.

The DC power input jack on the back is connected with thick (0.75 mm² or more) stranded wires to K7 of the MCU board. Use the same wire for all other power supply connections too. The swit-

## COMPONENT LIST




### Resistors

Default: 1%, 0.1W, 0603
R1 = 100kΩ
R2,R3,R6,R12,R15,R19 = 10kΩ
R4,R5 = 120Ω
R7 = 15Ω
R8,R9,R10,R17 = 2.2kΩ
R11,R14,R18 = 1kΩ
R13,R16 = 11kΩ
R20 = 22kΩ
R21 = 0.22Ω
R22 = 220kΩ
R23 = 47kΩ
P1 = 10kΩ linear potentiometer with solder lugs
P2 = 10kΩ horizontal trimpot

### Capacitors

Default: 0603
C1,C2,C3,C4,C7,C8,C14,C24,C25 = 100nF
C5,C6 = 33pF (optional)
C9,C10 = 10nF, 1%, U2J
C11,C12 = 1nF, 1%
C13,C21,C22 = 1nF
C15 = 100nF, 1206
C16 = 10nF
C17,C18,C19 = 1µF
C20 = 220µF, 10V, 6.3mm diameter, 2 or 2.5mm pitch
C23,C26 = 10µF, 35V, D 6.3 mm diameter, 2 or 2.5 mm pitch

### Inductors

L1 = 10µH, 250mA, 0.2Ω, 1210
L2,L3,L4,L5 = ferrite bead, 1kΩ @ 100 MHz, 200mA, 0.6Ω, 0603

### Semiconductors

D1 = PMEG3050EP
LED1 = RGB, 5mm, common cathode
IC1 = ATmega128A-AU, programmed
IC2 = TLV314IDBVT
IC3 = NCP431AVSNT1G
IC4 = MAX9711ETC+
IC5 = LD1085V50, TO-220

### Miscellaneous

K1 = 4-pin SIL pinheader, vertical, 0.1" pitch
K2 = 6-pin (2x3) pinheader, vertical, 0.1" pitch
K3,K4,JP1 = 3-pin SIL pinheader, vertical, 0.1" pitch
K5,K6,K7 = 2-way terminal block 630V, 0.2" pitch
K8 = 5-pin SIL pinheader, vertical, 0.1" pitch
LCD1 = HD44780-compatible, 2x16 character LCD
LS1 = mini speaker, 4W, 4Ω
S1,S2 = rotary encoder + push-on switch, Alps EC11E183440C
S3,S4,S5 = pushbutton, Multimec RA3FTH9
S6 = toggle switch, DPDT, 28VDC, 5A
X1 = 8MHz quartz crystal, 18pF, 5x3.2 mm (optional)

For IC5: Insulating kit, thermal pad TO-220 + bush TO-220
For IC5: M3 screw, nut, 2 pcs. washer
For JP1: shunt jumper, 0.1" pitch
For K1: 4-way SIL pinheader socket, vertical, 0.1" pitch
For K3,K4,P1: 3-way pinheader socket, vertical, 0.1" pitch
For K8: 5-way SIL pinheader socket, vertical, 0.1" pitch
For LCD1: 16-pin SIL pinheader, vertical, 0.1" pitch
For LS1: 2-pin pinheader, vertical, 0.1" pitch
For LS1: 2-way pinheader socket, vertical, 0.1" pitch
For P1: 3-way pinheader, vertical, 0.1" pitch
For S1,S2,P1: Knob, round, 12mm, shaft 6mm
For S3,S4,S5: Cap, black 19mm, Multimec 1S09-19.0
For S6: 3-way PCB screw terminal block 630V, 0.2" pitch
PCB # 160410-1 from Elektor Store

ched 5-volt power supply output K6 of the MCU board is connected to K10 of the FPGA-DSP board and from there connected to K6 of the RF Board. Screw terminal blocks have been used for power supply connections on all boards for a sturdy and secure connection.

If an external 5-volt power supply is used, the onboard voltage regulator can be omitted along with D1 and C23 to C26, and K5 is the input (also see Figure 3). Be careful, there is no polarity protection here! The 5 V on K5 is switched by S6. If this is unwanted, then K6 can act as the 5 V power supply input, but this is less practical as K6 is also supposed to be the distribution point of the supply voltage to the other PCBs.

### Audio

The miniature speaker is placed on the rear panel and connected by two wires to connector LS1 on the MCU Board. The speaker can also be mounted on either the bottom or top shell, whatever you prefer. If nothing is going to be placed on top of the enclosure, putting the speaker there would give a more direct sound.

The small speaker we used (ABS-230-RC) is small and easy to mount. However, its low-frequency response could be better; at 300 Hz the loss of sound pressure is already noticeable. Feel free to use a bigger 4 Ω speaker (2 W nominal minimum). Enough space is left on the front to add a 3.5-mm (maybe even a 6.3-mm) headphone jack. Connect left and right together if you have a stereo connector. If it has a switch, use it to disable the speaker. Put a resistor in series (start with 100 Ω or so) with the headphone output to limit the maximum output power and decouple any capacitive load of the cable.

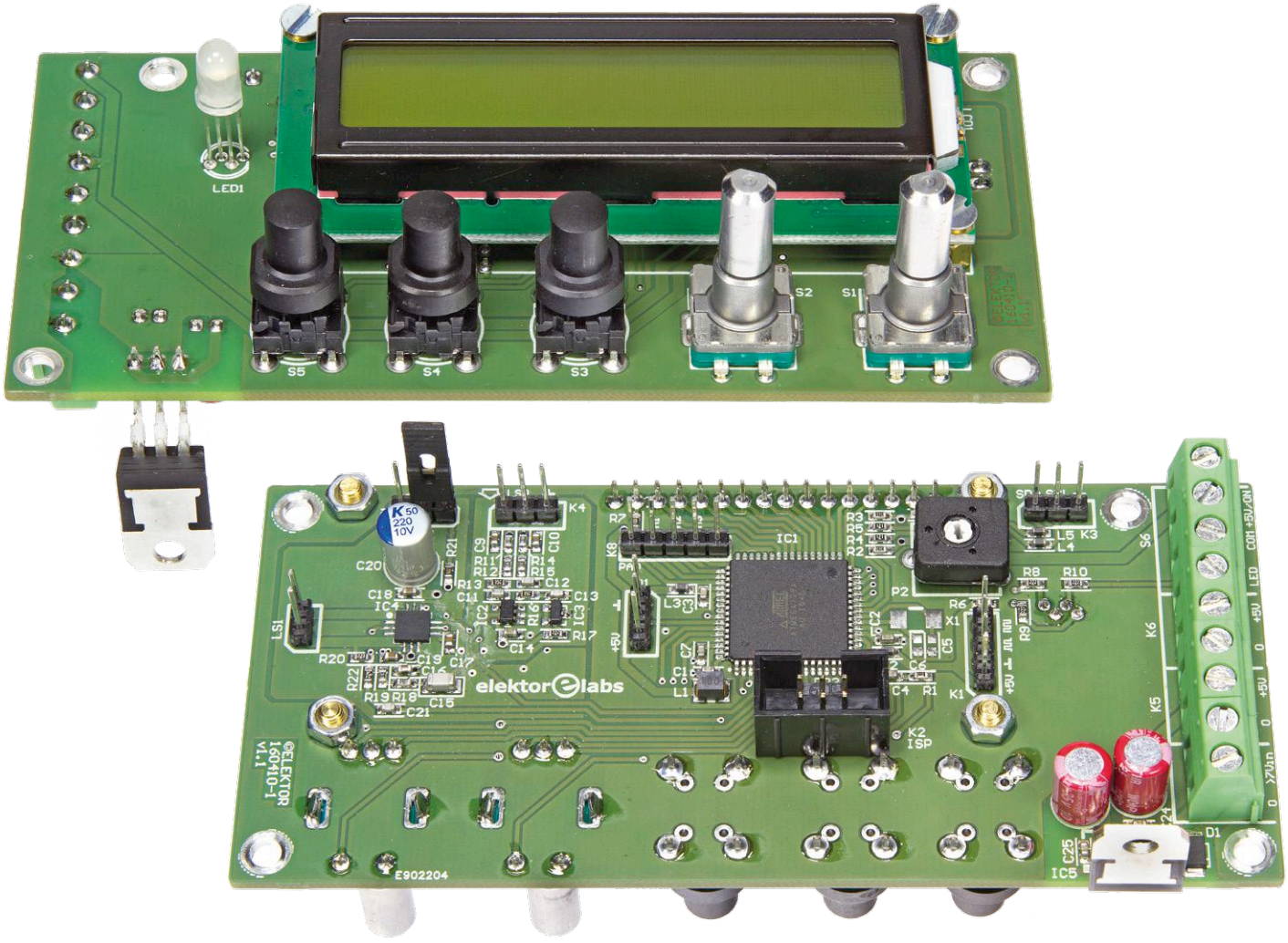
**Mounting the LCD**

Use a 1:1 printout of the bottom side component print of the MCU board as a drill template. (Be careful, a 1:1 laser print isn't always exactly 1:1.) The viewing area of the LCD is also printed on the bottom overlay, but it is for the (standard) module we have in our Store (120061-74/SKU 16414). LCD modules from other manufacturers may be slightly different. Always look at the datasheet for the exact location and size of the viewing area, and check that the pinout is the same.

The MCU board is mounted on four 12 mm M3 female-female standoffs extended with an M3 washer and locking ring (**Figure 6**). Together they place the board at exactly 13.4 mm behind the front plate, leaving just enough room to mount our LCD module 5 mm above the board. Again, if a module from another manufacturer is used this distance may have to be adjusted.

The LCD itself is fixed to the MCU board with four 5 mm M3 male-female standoffs and four short 4 mm screws. ⏮

(160410)




## Web Links


[1] www.elektormagazine.com/160410

[2] www.elektormagazine.com/150177

[3] www.elektormagazine.com/160160

[4] www.elektormagazine.com/labs/microcontroller-board-for-fpga-dsp-radio-160410

[5] Author's blog: sm6vfz.wordpress.com

[6] For the latest firmware: https://github.com/danupp/radiocontrol-mega128



**FROM THE STORE**

→160410-1
bare PCB
for Microcontroller Board

→120061-74
2×16 alphanumeric display

# Failed Fuse Indicator

By **Hans-Norbert Gerbig** (Germany)



You switch it on and... ping! Or perhaps a gentle 'pffft' or just silence, but nothing more after that. Who hasn't had this experience at some time? The usual suspect is the fuse. In the same way the normal 'quick fix' is to change the fuse for another one, even though this is not really advisable without knowing the precise cause of failure. But what else can you do when you're far away from the electronics lab? Well, try some simple checks. Is the fuse really defective — and if there are several, which one? Not easy to determine without testgear. Unless you have a built-in failure indicator...

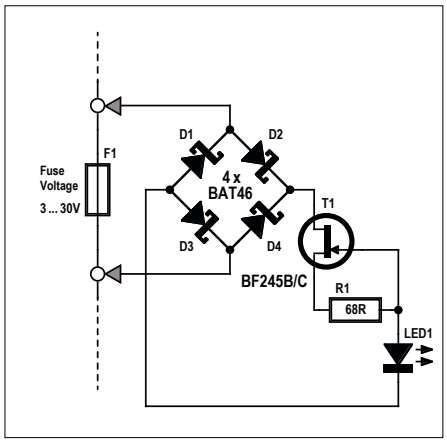



Figure 1. The circuit of the failed fuse indicator uses only seven components.

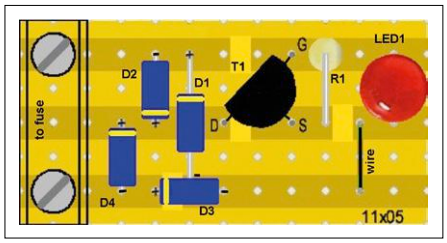



Figure 2. Here's how the circuit in Figure 1 can be built.

If the device went bang, fizzed or simply did nothing when turned on, you could assume that it has cost you a fuse (or several). If, with it still plugged in, the main circuit breaker has not cut out and there is no overall power failure, the other items connected to the same wiring circuit will at least still work.

Some instrument fuses are filled with silica sand, however, in order to extinguish any arcs arising when the fuse element (a length of fine wire) blows and cuts the current flow of any short circuit as rapidly as possible. This is good for the appliance we want to be protected but has the disadvantage that it's hard to see whether the fuse has actually blown or not. An optical indication would be no bad thing, particularly for devices with which this happens more often or not. This is easy at mains (line) voltage: a small bulb can be wired in parallel with the fuse. However, the failure indicator described here uses an LED to light up when the fuse blows and will also work at lower voltages.

## Failure indication

A failed fuse indicator needs to be simple, cheap, reliable and small. Ergo it should also require not many components. The circuit in **Figure 1** fulfills all of these criteria. Its operation is supremely simple: an LED is wired in parallel with the fuse along with a little electronic wizardry. Under normal conditions the LED is shunted by the intact fuse and remains unlit. If the fuse blows, the operating voltage is available via the load connected to the defective fuse. The parallel LED can avail itself of this and light up.

The electronics needed can remain straightforward. The first consideration is that the failure indication should operate regardless of DC polarity and on AC as well. This means having the LED plus current limiting circuitry driven by a bridge rectifier. To avoid too much voltage drop and maintain operation at low supply voltages, Schottky diodes are employed for D1 to D4.

The only other thing we need to ensure now is that the current through the LED is not only limited but also maintained unaltered across a broad voltage range. This works best using a constant current source. The simplest option is an N-channel barrier layer FET with a resistor between the Gate and Source. The

current flowing is dependent mainly on the slope of the FET or its DC voltage at a particular current across the resistor. However, one has to keep in mind the maximum voltage between Drain and Source and the maximum power dissipation. With the BF245 FET you can reach 30 V. The failure indication can therefore be used with fused voltages of between 3 and 30 V.

Performance considerations now: according to the datasheet [1] a BF245 can tolerate ambient temperatures up to 75 ºC 300 mW, getting significantly hot and bothered nevertheless. Better to keep it below half of that. The subtype BF245A produces a current of 4 mA when R1 = 0Ω. This way any LED will light up clearly and total power dissipation for T1 remains in the region of 100 mW even at 30 V. With the BF245B the figure is around 10 mA and finally around 18 mA with the BF245C. This makes things brighter. With R1 you can reduce the current through the LED at will. For the BF245B, a current of approximately 7 mA is obtained with a value of 68Ω for R1.

## Construction

The circuit is so simple that you should have no problem using perf board for the seven components. The placement plan in **Figure 2** will assist you. For convenient connection in parallel with the fuse a 3-way screw terminal connector (5 mm pitch) is provided. One more tip: for especially low voltage a red LED is recommended, as these exhibit the lowest voltage drop. To make this particularly noticeable use a flashing LED for LED1. ⏮

(160449)

## Web Link

[1] www.nxp.com/documents/data_sheet/BF245A-B-C.pdf

**About the Author**

Hans-Norbert Gehrig was employed as a teacher at the Kreuzburg Franciscan High School near Frankfurt am Main, where he also supervised various electronics student groups. Radio and electronics have fascinated him ever since his school days and now in retirement, he can devote himself to his technical hobby with all his heart.



Advertisement

THE BEST VALUE IN ELECTRONIC TEST & MEASUREMENT

SIGLENT®



359€ HT

SDS1202X-E
Oscilloscope Super-Phosphore

- Bande passante 200 MHz, 14 Mpts de profondeur mémoire
- 2 voies analogiques, échantillonnage 1 GSa/s
- **Décodage de bus en standard (IIC, SPI, CAN, LIN, UART)**
- **FFT 1 million de points Taux de capture maximum des signaux 100,000 wfms/s**



soit 1449€ HT

SSA3000X
Spectrum Analyzer

- 9 kHz to 2.1 GHz (SSA3021X), 9 kHz to 3,2 GHz (SSA3032X)
- min. -161 dBm/Hz DANL (plancher de bruit)
- **Opt. EMI/ AMK/ Reflection Measurement Module**
- **Promotion:**
- **min. 1 Hz RBW**
- **Option générateur suiveur gratuit**
- **SSA3032X: 2259€ HT（soit 740€de remise）**

**SIGLENT TECHNOLOGIES EUROPE GmbH**
Liebigstrasse 2-20, Gebaeude 14, 22113 Hamburg Deutschland

Tel.: +49(0)4081995946
E-mail: info-eu@siglent.com

www.siglenteu.com

eLABS PROJECT

# Traffic Signals for Model Railroads

## Programming with ladder diagrams

By **Rob van Hest** (The Netherlands)




It all started with a simple question: could the author make a controller for traffic signals on a model railroad? That led to the rediscovery of ladder diagrams as a simple and effective way to create program logic.

**PROJECT INFO**

- LED traffic signal
- Microcontroller
- Ladder diagram
- ➡ entry level
- intermediate level
- expert level
- 1 hour approx.
- Normal soldering tools, programmer (optional)
- €25 / £20 / $30 approx.

Nowadays when you need to automate something, the solution is obvious: you take an Arduino or Raspberry Pi module or choose one of the countless other single-board computers, put together a shield, a hat or whatever you want to call it, write some code, and you're done.

But using a complete minicomputer just to control a traffic signal is naturally a bit like using a cannon to swat flies. There must be an easier way, with a microcontroller and a little bit of I/O.

### Features

- Traffic signal for model railroads
- Ladder diagram programming
- Free programming environment
- Suitable for 12 V and 24 V
- All components through-hole

### PLC

In the real world, traffic signals are often operated by programmable logic controllers (PLCs), which are also used extensively for industrial control tasks. PLCs are usually programmed using ladder diagrams – and for good reason.

Microcontrollers are normally programmed using a high-level language such as C, which should not present any difficulties for most Elektor readers. However, service technicians and electricians in the industrial sector are not as fluent in these programming languages. On the other hand, they are perfectly at home with relay-based control circuits. That is the main advantage of ladder diagrams: if you can build a

circuit using electromechanical relays, you can program a PLC with the aid of ladder diagrams.
The author actually worked with ladder diagrams many years ago, and with a very convenient ladder compiler that runs perfectly under Windows 7 now available, the choice was easy: build the traffic signal controller as a mini-PLC and program it with ladder diagrams.

**An introduction to ladder diagrams**

As the name suggests, a ladder diagram looks something like a ladder with several rungs. Each rung consists of one or more switch contacts, which are drawn from left to right, and an actuator (the coil of a relay). For this brief introduction, we remain within the realm of electromagnetic relays. The switch contacts are depicted as follows:

```
---] [---
```
Normally open (NO) contact

```
---]\[---
```
Normally closed (NC) contact

As you can see, switch contacts are represented by square brackets in "reverse" order.

Actuators are indicated by round brackets (parentheses):

```
---( )---
```
Normally inactive actuator (unenergized relay coil)

```
---(/)---
```
Normally active actuator (energized relay coil)

As you probably already guessed, we can use these symbols to depict logical relationships.

A logic AND looks like this:

```
---] [----------------] [---------------( )---
   sw1                 sw2               motor
```

Here the motor is only activated (switched on) when contact sw1 and contact sw2 are both closed.

A logic OR takes the following form:

```
---+---] [---+---------------------------( )---
   |   sw1   |                           motor
   |         |
   +---] [---+
       sw2
```

A logic NOT (inversion) is indicated by a slash:

```
---] [----------------]/[----------------( )---
   sw1                 sw2                motor
```

Here the motor is activated when contact sw1 is closed and contact sw2 is not closed.
Using this method, you can construct a logic controller step by step (rung by rung) using switches and relays.

**A mental exercise**

As a sort of mental exercise, let's see if we can use this knowledge to do something useful – without actually drawing a circuit diagram.
Anyone interested in electronics is probably familiar with the "quizmaster" circuit: several contestants hear the quizmaster ask a question, and the one who presses the button first gets the chance to answer the question and score a point.
Now let's try to implement this circuit in the form of a ladder diagram.
We start with a master signal called "enable," which is only active when none of the contestants has pressed their button. When a contestant presses the button, an associated bistable relay is set and a lamp or something similar is switched on. The corresponding ladder rung looks like this:

```
  Yenable      XAplay              YAplay
----] [----+----] [-----------------(S)----
           |
           |   XBplay              YBplay
           +----] [-----------------(S)----
           |
           |   XCplay              YCplay
           +----] [-----------------(S)----
```

The "Yenable" signal is only true (active) when none of the contestant buttons is pressed:

```
   YAplay     YBplay     YCplay   Yenable
----]/[--------]/[--------]/[-------( )----
```

Of course, we also need a way to reset everything for the next round. That could look something like this:

```
  Yenable     Xreset               YAplay
----]/[---------] [----+------------(R)----
                       |
                       |           YBplay
                       +------------(R)----
                       |
                       |           YCplay
                       +------------(R)----
```

Finally, "END" indicates that we are done:

```
----[END]----------------------------------
```

As you can see, it is easy to construct a logic circuit with just a bit of logical thinking, without putting a single mark on a piece of paper.

**LDmicro**

That's all well and good, but we naturally need a programming environment if we want to draw ladder diagrams and compile them to produce hex files that can be loaded into a microcontroller. That is exactly what we want to present here. The program in question is called LDmicro, and it

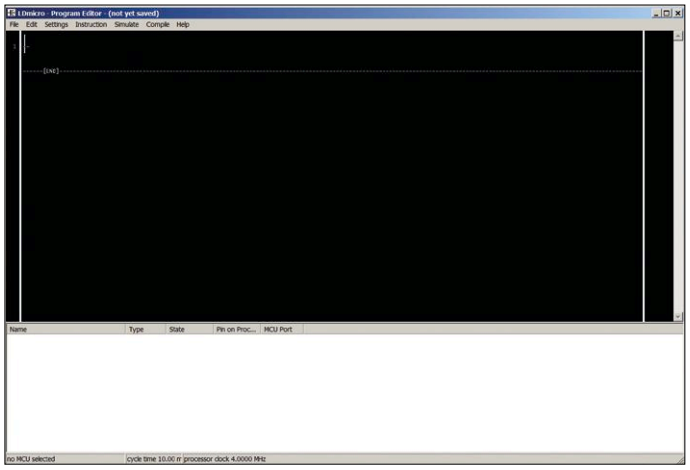


Figure 1. LDmicro starts with a clean sheet.

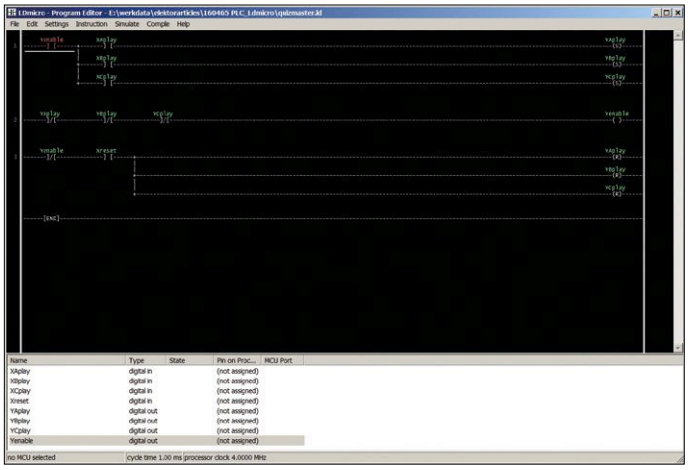


Figure 2. The quizmaster mental exercise in LDmicro.

works well, looks good, and does not cost anything [1]. The program takes the form of an executable file that does not have to be separately installed. **Figure 1** gives an impression of the user interface.
The diagram for our quizmaster experiment is shown in **Figure 2**. As you can see, each rung of the ladder is assigned a number, and the various contacts and actuators are listed in the bottom pane as inputs and outputs (this is done fully automatically).

The program documentation is exemplary in all regards, so it will not take you very long to learn to use the program and there is no need to go into the details of how to enter the various components of the diagram.
In order to translate the ladder diagram into a file that can be used to program a microcontroller (in other words, to compile it), you first have to select the target microcontroller as illustrated in **Figure 3**. As you can see, the program supports a variety of microcontrollers. Here the author selected the PIC16F628A, simply because he had a large number of these devices on hand.

Finally, before compiling the program you have to assign the various signals to the microcontroller pins. To do so, click on an input or output in the bottom pane and select the appropriate pin in the resulting pop-up window, as illustrated in **Figure 4**. Then compile the program and download the resulting hex file to the microcontroller. The author used a Velleman K8048 programmer for this, but the TL866A universal programmer available in the Elektor Store [3] also works well.
The download for this article [2] also includes the quizmaster example in a more elaborate version with an output for a buzzer. We leave it up to you (as a good exercise) to figure out how that works.

**Getting down to business**

Let's get back to where we started with this article: a traffic signal for a model railway. The object here is to safeguard an intersection, so we need four traffic signals in total, operating in pairs.
This could of course be constructed with a ring counter (using a couple of CD4017 ICs, for example) and a diode

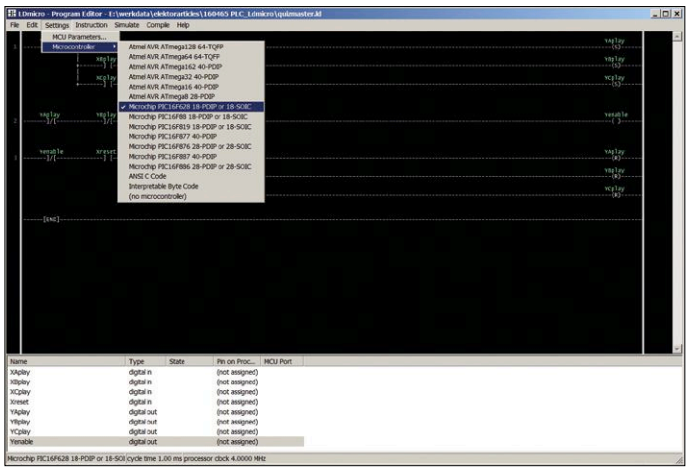


Figure 3. You can select the right microcontroller under Settings. LDmicro supports many different types.

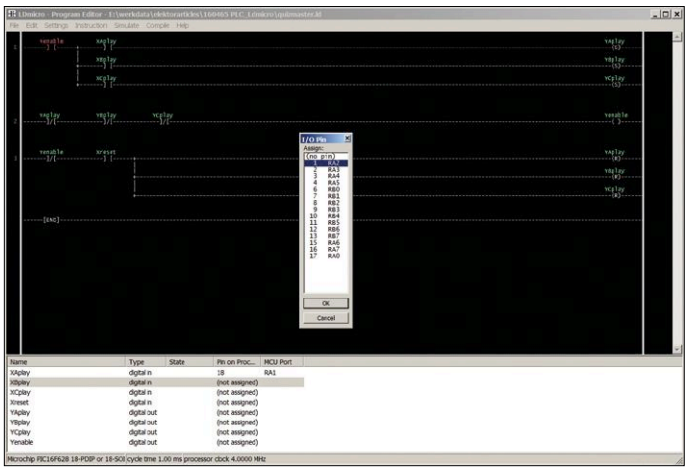


Figure 4. Assigning inputs and outputs is very straightforward.

matrix, but that has two disadvantages: you need a relatively large number of components, and it is difficult to modify the traffic signal behavior. We will therefore build a simple PLC that can be programmed using ladder diagrams.
The core of the program actually consists of a counter which counts 30 steps (from 0 to 29), of which 15 are allocated to one set of signals (which we call north/south) and the other 15 to the other set (east/west).
This counter can be programmed in LDmicro as a single rung.

```
  Rclock        Ton              Toff             Rclock
---]/[----[TOF 500.0 ms]--[TON 500.0 ms]--+-------( )------
                                           |
                                           |     Ccycle1
                                           +---{CTC 0:29}---
```

Here we employ two delays of 500 ms each, resulting in a cycle time of 1 second (1 Hz repeat rate). To keep track of the cycles, we use a counter which counts from 0 to 29 and then resets to 0.
The timing of the signal lights is determined by the counter state. For example, for the north/south pair we have:

| | |
|---|---|
| 1 or less: | red |
| 2–8: | green |
| 9: | yellow |
| 10 or more: | red |

This is fairly easy to translate into a ladder diagram:

```
   [Ccycle1 >=]     [Ccycle <]        Ygreen
---[ 2        ]-----[ 9      ]----------( )---

   [Ccycle1 ==]                       Yyellow
---[ 9        ]-------------------------( )---

   [Ccycle1 >=]                        Yred
---[ 10       ]--+----------------------( )---
                 |
   [Ccycle1 <]   |
---[ 2       ]---+
```

For the east/west pair, we simply add 15 to the previous counter states:

| | |
|---|---|
| 16 or less: | red |
| 17–23: | green |
| 24: | yellow |
| 25 or more: | red |

This is the main part of the traffic signal program. The complete program is included in the download file. There we added a pedestrian signal, an option for selecting either the signal sequence with yellow before green as well as yellow before red (as used in Germany) or the sequence without yellow before green, and a night mode (all signals blinking yellow). **Figure 5** shows part of the complete program in LDmicro, as well as the pin assignments.

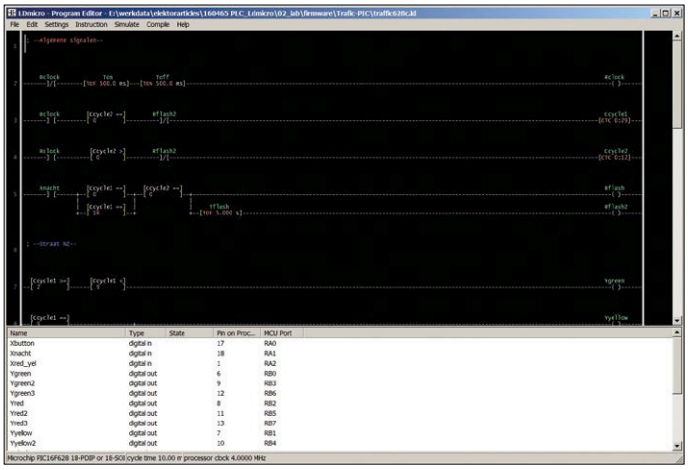


Figure 5. The traffic signal program in LDmicro.

## Pocket-size traffic signals

### The hardware

**Figure 6** shows the schematic diagram of the traffic signal controller. This does not need much explanation. The supply voltage input is on connector K7. The supply voltage stated on the schematic is 12 $V_{DC}$, which is commonly used for model railroads, but the circuit can be used unchanged with supply voltages from 9 V to 24 V. In the latter case, it is a good idea to fit a small finned heat sink on voltage regulator IC2 (type 78L05). The 5 V output from IC2 is used solely to power the microcontroller. Diode D1 provides reverse-polarity protection.

The circuit is built around the microcontroller IC1. No crystal is used here; the clock signal for the processor is generated by the free-running internal oscillator of IC1. It is more that sufficiently accurate for our purposes.
The inputs (connectors K4–K6) are protected by voltage dividers and series resistors, so that voltages up to a maximum of 24 V can be used as input signals without any problem.

The microcontroller outputs are buffered by the driver IC3 (type ULN2803). It should be familiar to most readers. The outputs of this IC switch to ground. The driver can handle approximately 60 mA per output when all outputs are active at the same time. That is more than enough for model railroad traffic signals. If you need more current, you can connect relays to the outputs, and the free-wheeling diodes necessary for this are already integrated in the IC.
Jumper JP1 selects between normal operating mode (2-3) and programming mode (1-2). In programming mode, the microcontroller can be programmed through connector K8 without removing it for the circuit (in-system programming, ISP). Of course, you can always program the microcontroller in a separate programmer.

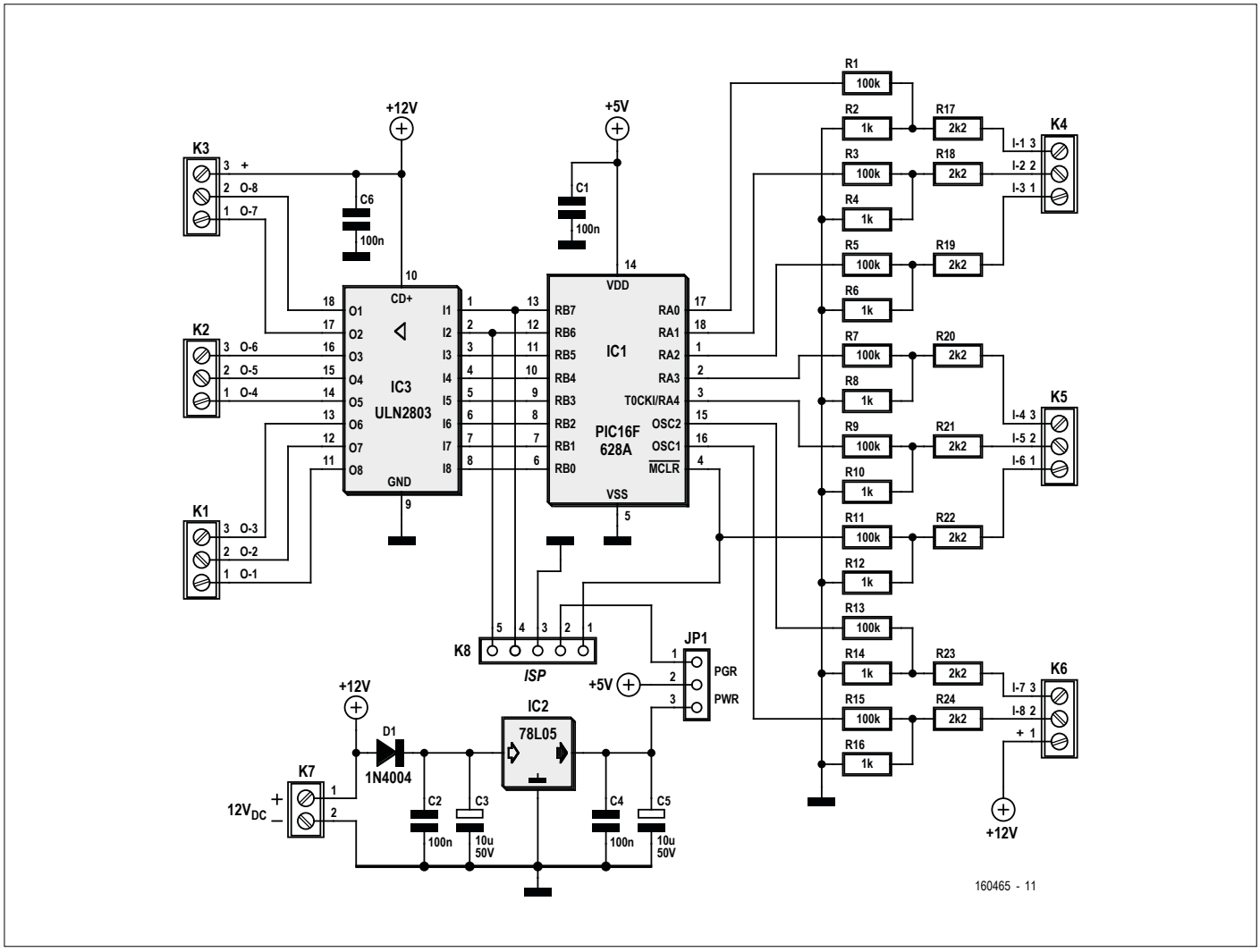



Figure 6. Schematic diagram of the traffic signal controller. This is actually a small general-purpose PLC, which with suitable software can control traffic signals on a model railroad.

## Construction and use

For the traffic signal controller we designed a PCB (single-sided) using exclusively conventional through-hole components (**Figure 7**). Assembling the board should not present any problems for most readers. Start with the sockets for IC1 and IC3, then mount the low-profile components (diodes and resistors), followed by the capacitors and connectors. After a careful visual inspection, you can connect a 12 V power source to connector K7 (pay attention to the right polarity) and check for a clean and stable 5 V supply voltage on pin 3 of JP1. If that test is passed, switch off the power and place a jumper on pins 2 and 3 of JP1 (where it can remain forever if you never need to program the microcontroller in system) and insert the microcontroller (IC1) and the driver (IC3) in their sockets. Next, connect the LEDs of the various traffic signals and the switches as indicated in **Figure 8**. Note that if you want to use a 24 V supply voltage instead of 12 V, it is probably a good idea to increase the series resistor values for the LEDs to 2.2 kΩ. Using the controller is easy: press S1 briefly to activate the pedestrian light (just like real life, where you always have to press a button before you can cross safely).

Switch S2 activates night mode; when it is switched off, the controller sets all lights to red for safety before changing to normal mode. Finally, S3 allows you to select either "NA/UK" or "Germany" signal sequence.

One final tip: If you use a 24 V supply voltage, it is a good idea to increase the value of R17–R24 to 4.7 kΩ. This will prevent the voltage on pin 4 of the microcontroller from rising too high. Although that would not cause any serious damage, it might unintentionally put the microcontroller in programming mode.

## Conclusion

A model railroad traffic signal is of course a perfectly legitimate application for this circuit. However, we hope this project has aroused your interest in programming with ladder diagrams — using the board described here (with its wealth of inputs and outputs), you can implement a wide range of attractive applications without any programming language experience. Be sure to tell us what you use this PLC for — we're keen to know!

(160456-I)

## COMPONENT LIST

**Resistors**

Default: 5%, 0.25W

R1,R3,R5,R7,R9,R11,R13,R15 = 100kΩ
R2,R4,R6,R8,R10,R12,R14,R16 = 1kΩ
R17,R18,R19,R20,R21,R22,R23,R24 = 2.2kΩ

**Capacitors**

C1,C2,C4,C6 = 100nF, 0.2" pitch
C3,C5 = 10µF 50V, 0.1" pitch

**Semiconductors**

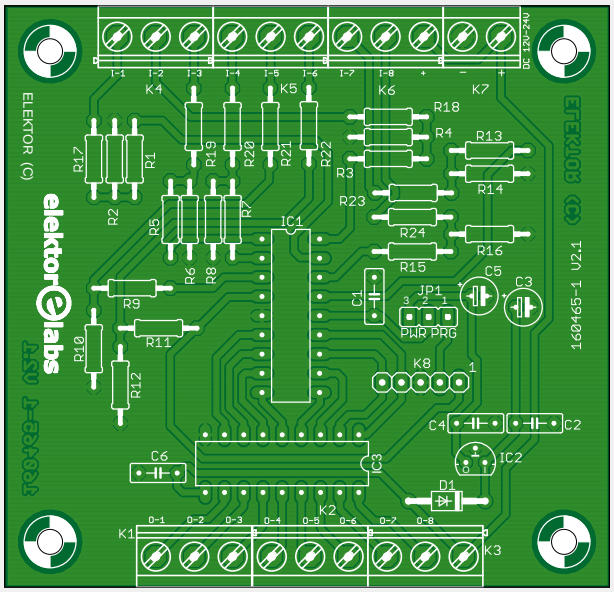
D1 = 1N4004
IC1 = PIC16F628A, programmed, Elektor Store # 160465-41
IC2 = 78L05
IC3 = ULN2803

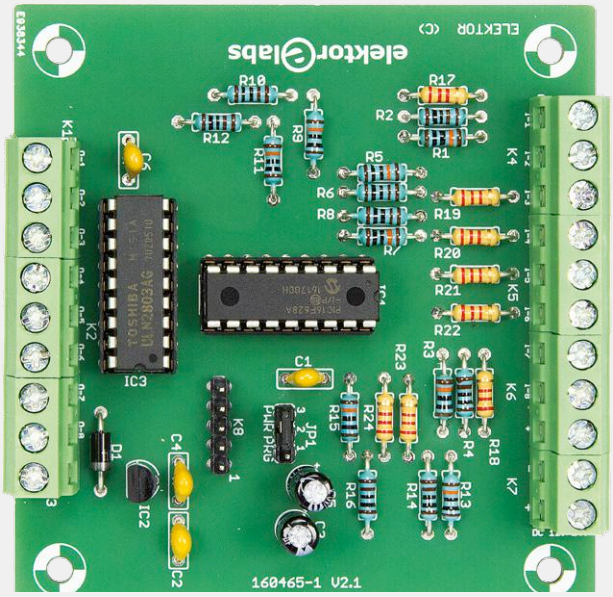
**Miscellaneous**

K1–K6 = 3-way PCB screw terminal block, 0.2" pitch
K7 = 2-way PCB screw terminal block, 0.2" pitch
K8 = 5-pin pinheader, 0.1" pitch
JP1 = 3-pin pinheader, 0.1" pitch, with matching 2-pin jumper
2x 18-pin DIP IC socket

PCB 160465-1

Figure 7. The PCB for the PLC / traffic signal controller.




### Web Links

[1] http://cq.cx/ladder.pl
[2] www.elektormagazine.com/160456
[3] www.elektor.com/tl866a-universal-programmer

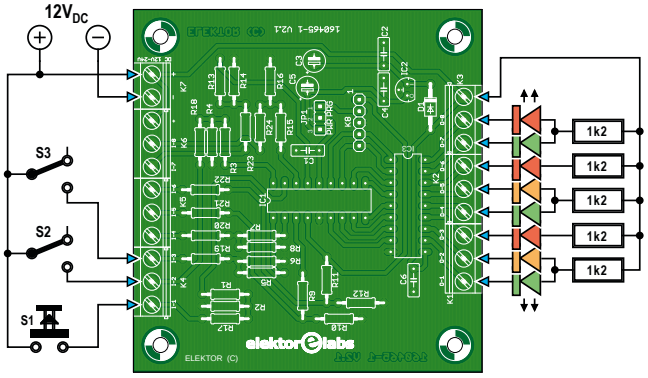



Figure 8. Wiring diagram for the LEDs and switches.

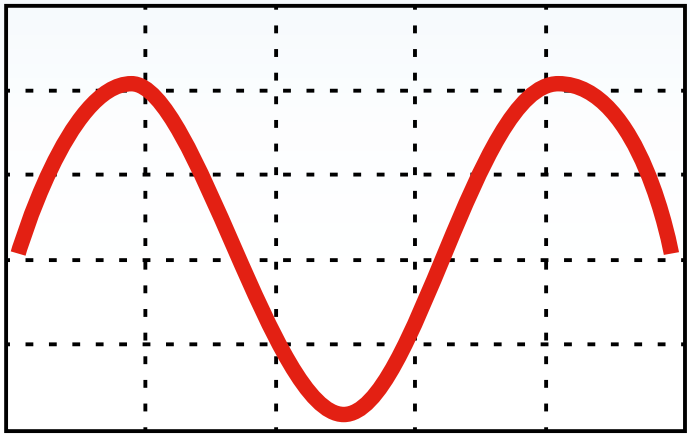
## FROM THE STORE

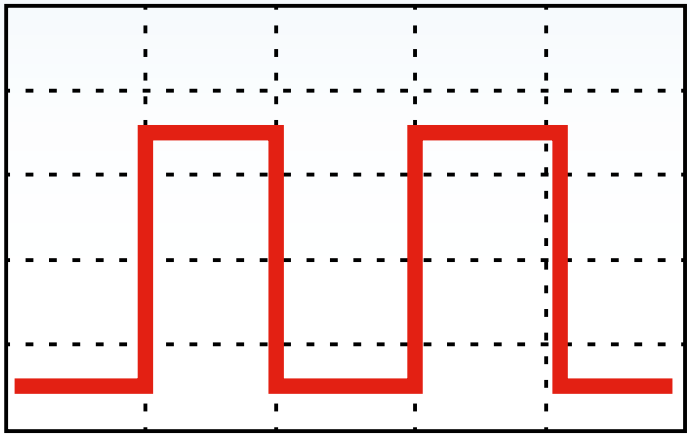
→ 160465-1
PCB

→ 160465-41
Programmed microcontroller

# Function Generator Shootout

## Three pro products compared





By **Alfred Rosenkränzer** (Germany)

The models discussed are the PeakTech 4125, Siglent SDG1035X and the Rigol DG1032z. As a kind of benchmark we'll reference them against the Keysight 33522A, a professional model that the author has available (its correspondingly higher price ticket rules it out from taking part in our comparison).

All of the generators operate on the direct digital synthesis (DDS) principle and operate at 125, 150 or 200 MS/s as appropriate, with 14-bit D-to-A converters and two channels. Only the reference device delivers 250 MS/s and a resolution of 16 bits. The upper frequency of the sine-wave signals produced is limited to between 25 to 30 MHz. From their similar data you might expect the performance of each device to be comparable but is that the case? You will have to read the article to find out!

### Fundamentals

In principle a function generator is the opposite of an oscilloscope. Whereas the latter's task is to make signals visible, the function generator produces signals (of varying form and frequency) that appear on the 'scope. Particularly when evaluating amplifier stages we require an externally produced signal whose treatment and quality can be assessed with an oscilloscope. The consequence of this elemental challenge is that the quality of signals produced by the function generator must be as good as absolutely possible. Just as desirable as the standard sine-wave is a choice of other wave shapes and, where appropriate, more complex waveforms such as frequency sweeps. Modern function generators can handle all of these requirements, even at frequencies up into the MHz region. More specialized versions also exist, such as generators for audio signals of exceptional purity or RF generators that provide radio frequencies and modulation capabilities. But in this article we are looking at 'normal' multi-purpose function generators.

### Operating modes

Whereas until the 1980s special variable frequency circuitry, analog in nature, was still being used, today everything is digital. The keyword is DDS (direct digital synthesis) and how this works is described concisely but adequately in Wikipedia.

The top frequency achievable by this method depends on the clock rate of the DDS chip and according to Nyquist amounts to a maximum of 50% of this clock signal. As a rule, however, multiple grid points are used for improved signal quality, meaning you can reckon on only 10 to 25 %. Frequency resolution is determined purely by the width in bits of the phase register. 48 bits is absolutely standard these days and the current normal resolution leaves nothing to be desired. The quality of the signals is determined largely by the characteristics of the D-to-A converter (its resolution) and other circuitry components, which need to produce the lowest possible jitter level in the clock signal.

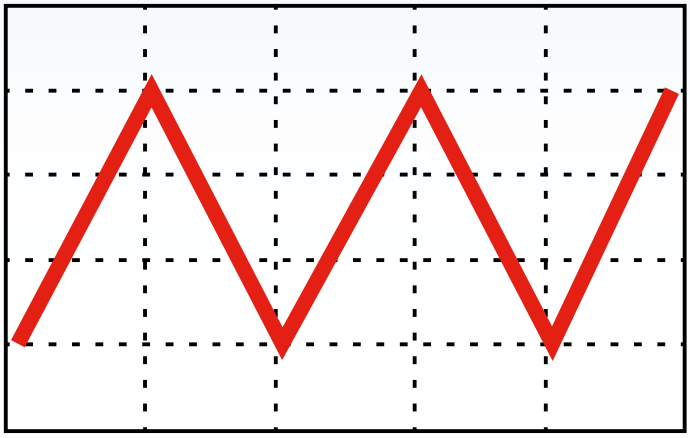
### Signals

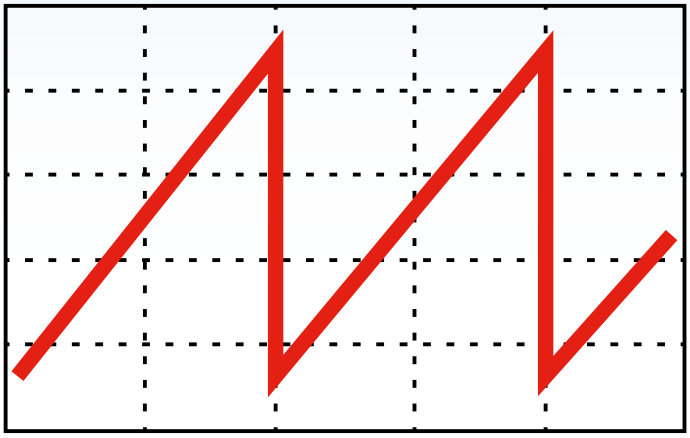
It goes without saying that modern function generators should provide as well as the obligatory sine-wave signal, in which aspects such as harmonic distortion and signal-to-noise ratio primarily dominate, also additional signal shapes such as square-wave, triangle, sawtooth and pulses (of customizable duty cycle). Fast rising and falling edges are vital for pulse and square-wave signals, with minimal overshoot, whilst for sawtooth and triangle waveforms the voltage wave-shape should be as linear as possible. These days the ability to save and repeat custom (so-called arbitrary) signal forms is no longer a 'dream ticket' indulgence either.

### Other considerations

It's self-evident that purchase decisions are determined not merely by technical details but also after considering ergonomic usability, the electromechanical quality of the controls,

Modern function generators are abundantly complex these days and unlike in analog times they offer far more functions. Studying the various data sheets is worthwhile, even though the copious words and numbers in there do not tell the full story. To help you form your own opinion this article showcases three instruments offering a broad range of applications that might suit a small lab at an acceptable price.





a good display and the provision of special functions. Many function generators can be controlled from a PC and are equipped for this with a USB or network connector. Also desirable is the ability to set an offset (superimposed DC voltage) with the widest possible range, along with a normalized output impedance of 50 Ω.

Additional functions, such as the ability to modulate signals, frequency sweeps across the broadest possible (and presettable) range and a capability for generating bursts (a set number of signal periods) are the norm. An amplitude sweep is admittedly not so vital but it's handy to have on occasions nevertheless. A dedicated sync output can help ensure positive triggering of an oscilloscope when signals are distorted. Also worth mentioning are extras like a reference input and output for the internal clock, together with a second output for differential signals. A not entirely necessary bonus is an integrated frequency counter, which thanks to the hardware already installed can be realized very economically, as it costs the manufacturer only a few lines of code.

## PeakTech 4125

The operating manual supplied on CD clearly relates to an older version of the device (**Figure 1**). The front panel of the newer version in fact has a USB connector for an external hard drive. On the other hand the manual still shows the connector on the rear panel, which looks completely different. The reference inputs and outputs are labeled '20 MHz' but customarily these are 10 MHz. It's a shame that the manufacturer does not (yet) offer a newer version of the user manual.

Irritation begins at switch-on: the noise of the fan is loud and intrusive, arising not from any defect but more probably from excessive rotational speed, providing abundant ventilation at the price of peace and quiet. For around €455 / £420 / $540 one might expect something different.

Operation, as on all such devices, uses soft-keys surrounding the LCD together with a rotary encoder switch and a keypad. BNC connectors are used for the signal inputs and outputs. The description that follows mentions the means of performing individual functions only when they are particularly good or terrible solutions.

**Sinewave signals** can be adjusted up to a frequency of 25 MHz, which accounts for an acceptable 20 % of the sampling frequency. There's an informative




Figure 1. PeakTech 4125: A newer version than the model illustrated here is now on the market, conveniently equipped with a USB connector on the front.

discussion on sample rates and ratios at [1]. This provides sufficient reserve for the use of a simple output filter. The maximum amplitude amounts to 20 $V_{pp}$ up to 10 MHz and 10 $V_{pp}$ above this. The offset possible is ±10 V. All specifications relate to Hi-Z (= high impedance, unterminated) outputs.

**Squarewave signals** are limited to 5 MHz and additionally to a duty cycle of 50%. The maximum frequency is too low to clock modern microcontrollers or digital circuitry for test purposes.

**Pulses** are also limited to 5 MHz. Unlike the duty cycle, the <10 ns rise and fall times indicated are not adjustable.

**Ramp signals** can be generated up to 1 MHz.

There is no **Sync-Output** for triggering oscilloscopes externally if you alter the amplitude and offset. The handbook mentions this function but I was unable to find it.
The PeakTech product does indeed provide **two channels** but they do not have the same characteristics. You can use all types of modulation and the frequency sweep facility on Channel 1 only. In reality this is not a serious restriction.

**Differential outputs** for audio or rapid digital signals are certainly possible but are somewhat complex to set up. You can copy the characteristics of one channel over to the other but you cannot link them. For instance, as soon as you change the frequency, you have to transfer this across.

With **modulation** the maximum internal modulation frequency amounts to 20 kHz, which makes it suitable for the audio range.

**Reference frequency**: the reference output (10 MHz in fact) provides a trapezium-shaped signal of 1.6 $V_{pp}$ into 50 Ω.

## Siglent SDG1032X

This time the user instructions do match up with the actual device, thank goodness. Once again the fan noise is definitely audible, even if not nearly as intrusive as on the PeakTech. In price terms, at around €450 / £420 / $540, this generator is in the same ballpark.




Figure 2. Siglent SDG1032X: Straightforward and free of frills. Once again the USB connector is fitted on the front.

**Sine-wave signals** are generated up to 30 MHz with a sample rate of 150 MHz, which results in a ratio of 20 %. Here again the maximum amplitude is 20 $V_{pp}$ up to 10 MHz and 10 $V_{pp}$ above this — both measured into Hi-Z. A remarkable setting option, which I had not observed previously, is the 'Harmonic' function. If you enable this you can add defined harmonic components with adjustable amplitude and phase to a sinewave signal. This not only lets you check how you can avert defined harmonics in circuits but also even eliminate a harmonic produced in an amplifier by adding a signal of equal amplitude with 180° phase.

For **squarewave signals** you have virtually no restrictions. Even at 30 MHz you can still adjust the duty cycle in the range 41 to 59%. At lower frequencies the adjustment range of the duty cycle is extended significantly. Rise and fall times of 4.2 ns are indicated.

When it comes to **pulses**, if you select the 'Pulse' curve shape you can even preset the rise and fall times — an extraordinarily useful feature. Compared to the square-wave function, the minimum times are slightly increased by 16.8 ns.

Using 'Delay' you can shift the signal of the **second channel** in a relative manner, similar to adjusting the phase of the sine-wave signal.

**Ramps** up to 500 kHz are possible.

Using the **DC function** you can generate an adjustable DC voltage in the range of ±10 V.

On the rear panel of the case there is a **Sync output** that you can enable in the Sync menu and link this with one of the channels. However, in comparison with the signal channel this displays jitter at higher frequencies and is therefore of limited use. On the data sheet [3] its maximum frequency is given as 1 MHz, although the function continues to work up to 10 MHz.

On this Siglent device **both channels** appear to work identically — at least I could not detect any restrictions. You can even add the signals from the two channels.

Regarding **modulation**, the maximum internal modulation frequency is 20 kHz, as against 50 kHz for the external input.

The reference output provides a square-wave signal (filtered by an R-C low-pass) of 10 MHz at an amplitude of 1.5 $V_{pp}$ into 50 Ω.

**Differential signals** can be generated very simply. Next to the Copy command there is a Tracking Menu, in which you can set the parameters to be




Figure 3. Rigol DG1032Z: Here not just the USB connector but also the input for the integrated frequency counter are both installed on the front panel.

linked, even by using an offset value.

## Rigol DG1032Z

This comes with a 'Quick Guide' but no CD. The 'proper' operating instructions are of course downloadable from the Rigol website and there is even a German version. Plus point: the fan noise is moderate. At around €560 / £520 / $670 the function generator is a good 20% dearer than the other two devices.

**Sine-wave signals** are generated up to 30 MHz with a sample rate of 200 MHz. With a ratio of only 16.7 % an extra grid point per period is produced. If the impedance is set to 50 Ω (not as Hi-Z) the amplitude can be defined only in dBm. With open-circuit (unterminated) outputs the maximum amplitude is once again 20 $V_{pp}$ up to 10 MHz, dropping to 10 $V_{pp}$ above this frequency. Here again there is a 'Harmonics' function for setting harmonics, albeit restricted to eight harmonics.

**Squarewave signals** are limited to 15 MHz. You can adjust the duty cycle within the range 19 to 81 %, with an even broader range at lower frequencies. The rise and fall times are given as <10 ns; my oscilloscope confirmed this as around 7 ns. There is an error in the data sheet [3], where 25 MHz is given as the maximum frequency.

**Pulses** are again feasible up to 15 MHz. You can additionally make the rise and fall times larger than the minimum values of 10 ns.

**Ramps** can be generated with a frequency up to 500 kHz maximum.

On the rear panel of the Rigol product two independent **Sync signals** are provided for Channels 1 and 2. No jitter is detectable, even at 15 MHz. This works for sine-wave signals up to the maximum frequency of 30 MHz.

Here again **both channels** are apparently identical — I could not detect any restrictions.

**Modulation**: at 1 MHz, the maximum modulation frequency is extraordinarily high.

The **reference output** provides a 10 MHZ square-wave signal with 1.5 $V_{PP}$ into 50 Ω with steep flanks.

Here too **differential signals** can be generated very easily. Beside the Copy command there is a Tracking Menu Next to Copy command there is a Tracking Menu, in which you can set the parameters to be linked, even by using an offset value.

## Keysight 33522A

We have there are no issues of importance with the reference device. This signal generator is what it sets out to be and it functions in the same way too. However, for professional equipment of this rank you should expect to pay a good €3,000 / £2,775 / $3,580.

**Sine-wave signals** are generated up to 30 MHz with a sample rate of 250 MHz. The ratio of 12 % allows more than eight grid points per period. The D-to-A converter provides 16-bit resolution. The maximum amplitude amounts to 20 $V_{pp}$ (Hi-Z) up to 30 MHz without any restriction.

**Squarewave signals** and **pulses** are possible up to 30 MHz. Rise and fall times are fixed for square-wave at 8.4 ns; for pulses you can also select larger values.

**Ramps** are limited to a maximum frequency of 200 kHz here.

Using the **DC function** you can produce an adjustable DC voltage in the range ±10 V.

The **PRBS** (pseudo-random bit stream) function is available exclusively on the Keysight product. A noise signal of this kind is very handy for determining bandwidth.

The Keysight generator is the only device that has the sync output on the front panel. There is no frequency restriction, nor any visible jitter.

As to **modulation**, using the external input signals can be modulated with frequencies up to 100 kHz.

**Frequency counter**

All three products under review have an integrated frequency counter with a bandwidth of up to 200 MHz. I measured the sensitivity of each at three different frequencies. **Table 1** shows the results. The levels in dBm are valid for Hi-Z (= high-impedance). The last line shows additionally the maximum frequency up to which each counter still works.

All three products can switch between AC and DC coupling. It is also feasible to activate a low-pass filter and set the trigger level. On the PeakTech device you can set the sensitivity in three steps, although this requires a higher level than on the other two examples. Furthermore, only six settings were given — a drawback versus the Rigol product with seven and the generator

| Table 1. Sensitivity of the frequency counter. | | | |
|---|---|---|---|
| **Frequency** | **Siglent** | **PeakTech** | **Rigol** |
| 1.97 MHz | -30 dBm | -18 dBm | -20 dBm |
| 19.7 MHz | -32 dBm | -22 dBm | -22 dBm |
| 197 MHz | -25 dBm | 2 dBm | -11 dBm |
| Maximum: | 265 MHz | 310 MHz | 200 MHz |

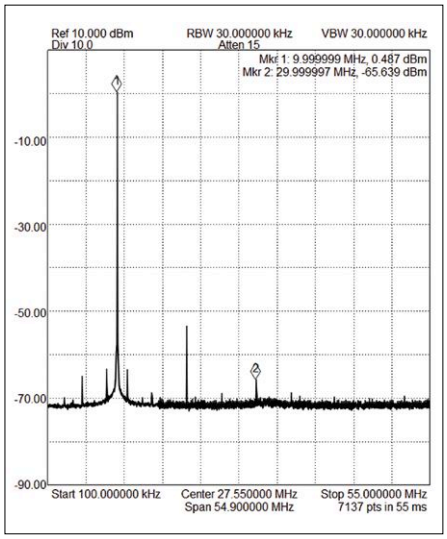



Figure 4. PeakTech 4125: Spectrum of the 10 MHZ sine-wave signal. The level of the third harmonic is –65 dB.

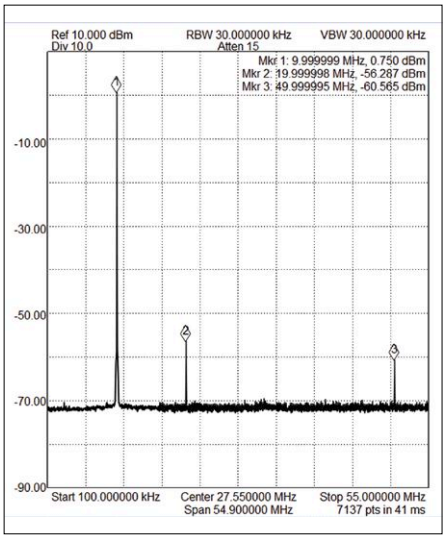



Figure 5. Siglent SDG1032X: Spectrum of the 10 MHZ sine-wave signal. The level of the third harmonic is down in the noise. The fifth harmonic is at –60 dB.

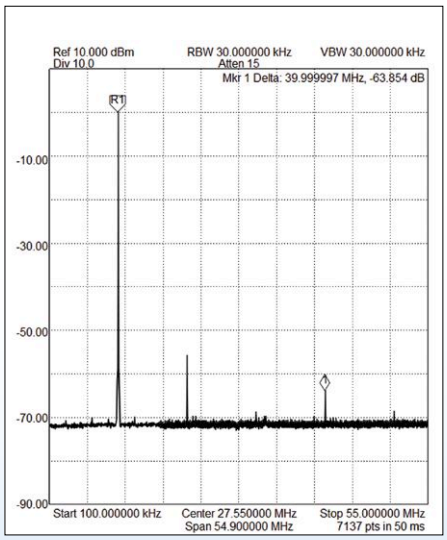



Figure 6. Rigol DG1032Z: Spectrum of the 10 MHz sine-wave signal. The level the third harmonic is almost in the noise. The fourth harmonic is at –64 dB.

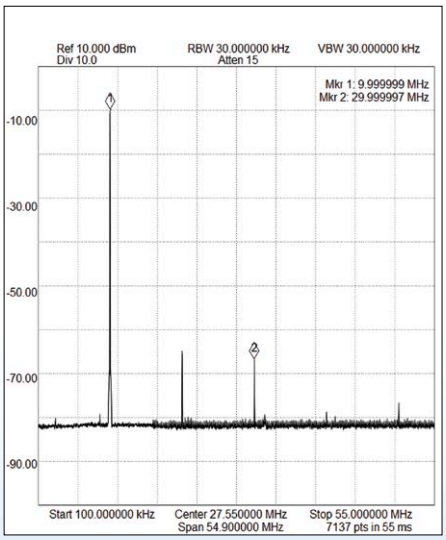



Figure 7. Keysight 33522A: As a comparison here is the spectrum of the 10 MHZ sine-wave signal of the reference device. At -57 dB the level the third harmonic is no better.

from Siglent with yet eight positions. There's another restriction with the frequency counter on the PeakTech device: only the direct frequency is displayed, whereas both of the others can indicate the center, minimum and maximum values. The Rigol function generator shines out as very practical, with its counter input on the front panel.

For me what is missing on all three devices is a switchable 50-Ω terminating resistor. To avoid signal reflections you will need to use a through-pass termination where appropriate.

### Signal quality

For comparing signal quality on each product I set up **sinewave signals** at 1 and 10 kHz with a level of 1 V RMS and measured them with an audio analyzer. **Table 2** indicates the harmonic distortion and **Table 3** the headroom between the wanted signal and the combination of distortion and noise. Although the harmonics differed by more than 10 dB, from my point of view not one of the generators is fit to serve as a low-distortion signal source for audio measurements without additional filtering.

### Spectrum test

The spectra of a 10 MHz signal with a level of 0 dBm were measured using a spectrum analyzer. When you make direct measurements the bulk of the first harmonic is attributable directly to the spectrum analyzer account, so a 20 MHz high-pass filter was connected in series. The latter attenuates the 10 MHz signal, making it possible to raise the sensitivity of the spectrum analyzer.

### Reference signal

**Figures 8**, **9** and **10** show the curve shape and amplitude of the 10 MHZ Signals at the reference output of the three function generators. As you can see, the signal of the PeakTech product resembles a sine-wave with some clipping, whereas the generator from Siglent clearly provides a lowpass-filtered square-wave signal. Only Rigol delivers a square-wave signal with sharp edges, which should mean entirely problem-free triggering.

### Feel and finish

All three generators are housed in

| Table 2. Distortion. | | | |
|---|---|---|---|
| **Frequency** | **PeakTech** | **Siglent** | **Rigol** |
| 1 kHz | 68 dBc | 62 dBc | 73 dBc |
| 10 kHz | 70 dBc | 62 dBc | 74 dBc |

| Table 3. Distortion + Noise. | | | |
|---|---|---|---|
| **Frequency** | **PeakTech** | **Siglent** | **Rigol** |
| 1 kHz | 62 dBc | 62 dBc | 73 dBc |
| 10 kHz | 62 dBc | 63 dBc | 74 dBc |



Figure 8. PeakTech 4125: Curve shape of the 10 MHZ signal at the sync output.

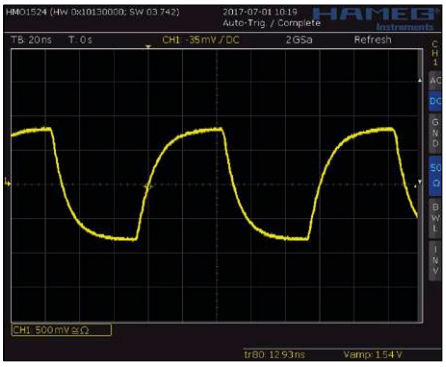


Figure 9. Siglent SDG1032X: Curve shape of the 10 MHZ signal at the sync output.

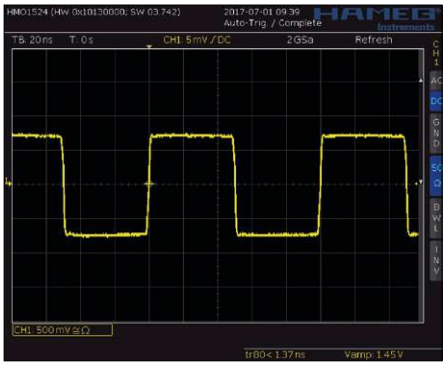


Figure 10. Rigol DG1032Z: Curve shape of the 10 MHZ signal at the sync output.

## Is there a standout winner?

high quality enclosures. The operating controls and connectors give a robust impression. How stable they are in reality becomes apparent only after lengthier use. Also decisive is whether a device is used daily in a professional lab or sees only infrequent use in a hobby workshop. In this respect the Keysight device makes a particularly good impression bit it is also significantly more expensive than the three generators showcased here.

**External control**

The remote controllability of the products was not investigated, as this seemed in part extremely time-consuming. The generators from Siglent and Rigol each provide a USB and a LAN connection. For this the PeakTech product comes with USB and an RS-232 interface.

**Verdict**

In my personal opinion the function generators from Siglent and Rigol share first place. Both have strengths and weaknesses, which more or less balance out. On the Siglent product I like the way that square-wave signals are available up to 30 MHz, which is very useful when working with microcontrollers. It's unfortunate, however, that the sync output has only restricted usability, in contrast to the model from Rigol. With the latter you must content yourself with 15 MHz maximum for square-waves. For me the PeakTech generator has to take third place. The noise from its blower disturbed me a lot.

The function generator by Keysight remains the yardstick for solidity and displays no shortcomings. That said, it comes without a frequency counter and for hobby pursuits its price ticket puts it the wrong side of level-headed budgets.

(160476)


**The Author**

Alfred Rosenkränzer has worked for more than 30 years as a development engineer, initially in the professional television technology field. Since the end of the 1990s he has developed digital high-speed and analog circuitry for IC testers.


**Web Links**

[1] www.dataq.com/data-acquisition/general-education-tutorials/what-you-really-need-to-know-about-sample-rate.html

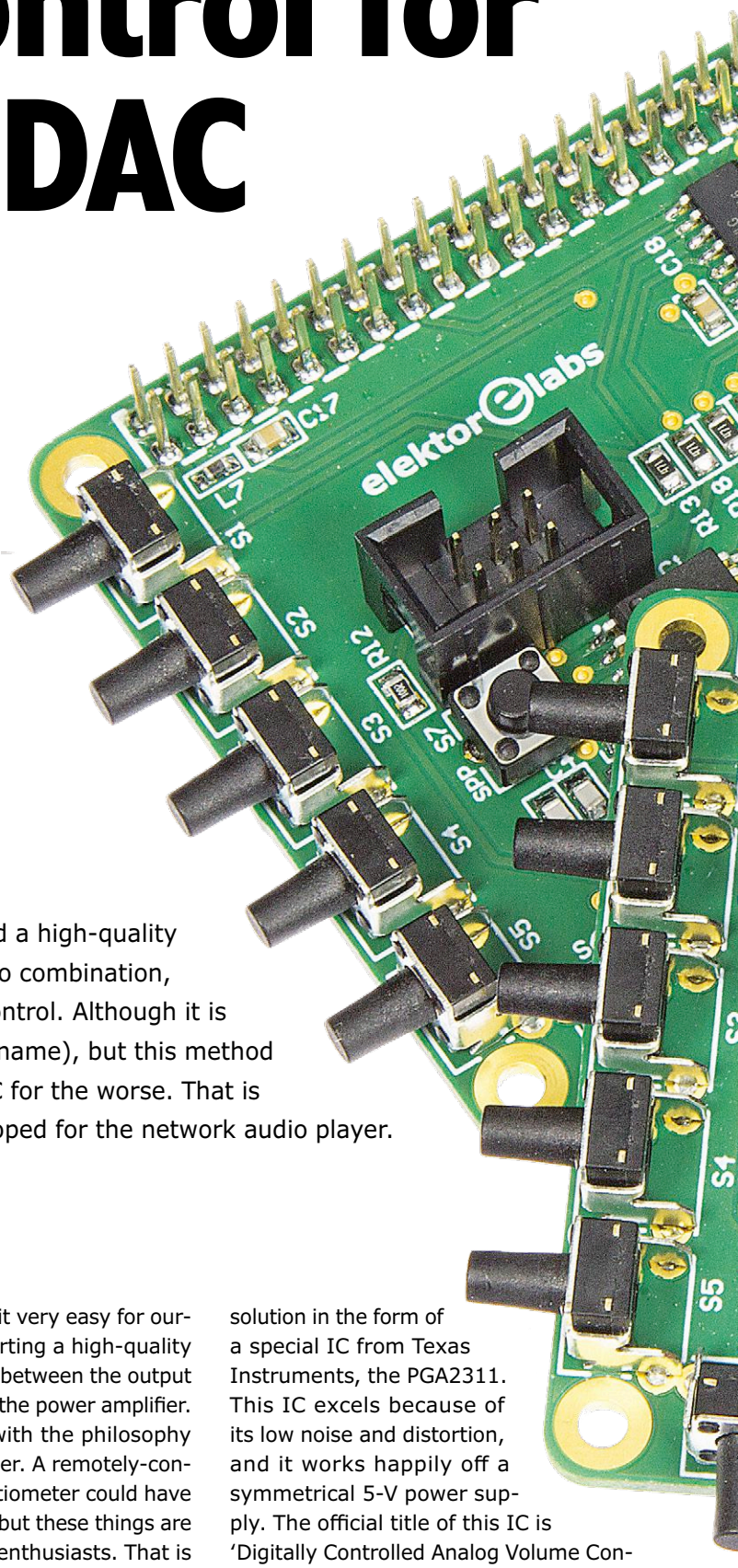
[2] PeakTech 4125: www.peaktech.de/productdetail/kategorie/dds-funktion-arbitrary-generator/produkt/p_4125.html

[3] Siglent SDG1032X: www.siglent.eu/siglent-sdg1032x-function-generator.html

[4] Rigol DG1032Z: www.rigolna.com/products/waveform-generators/dg1000z

# Volume Control for RPi Audio DAC

## loud and quiet in high quality



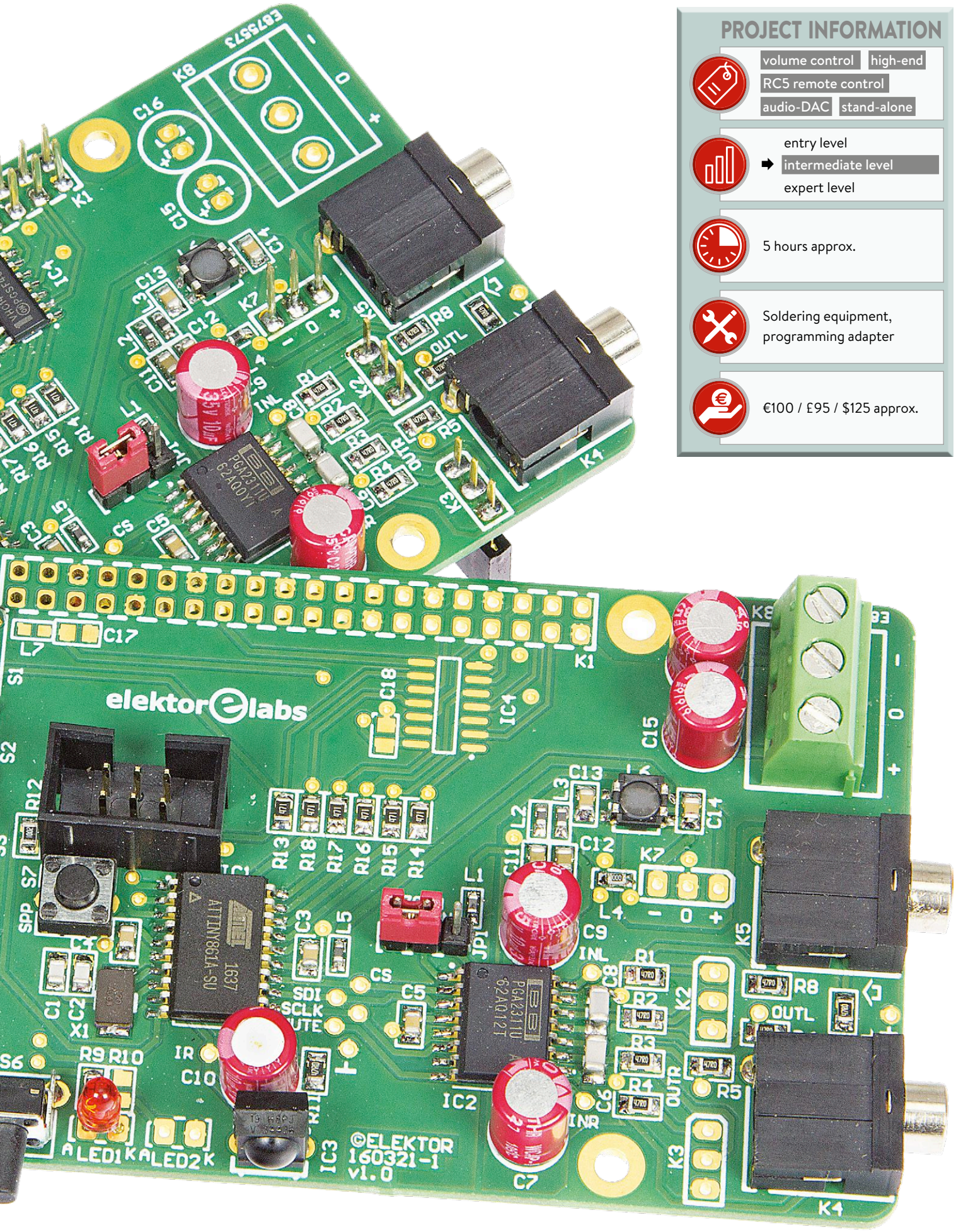

By **Ton Giesberts** (Elektor Labs)

To the delight of many, the previous issue of Elektor had a high-quality network audio player based on the Raspberry Pi/Volumio combination, which however was not provided with a 'real' volume control. Although it is possible to control the volume via Volumio (what's in a name), but this method seriously affects the high-end characteristics of the DAC for the worse. That is why we now describe a volume control especially developed for the network audio player.

### Characteristics

- High-end volume control
- Suitable for Raspberry Pi + audio DAC
- Compact sandwich construction
- Uses an RC5 remote control
- Also suitable as stand-alone volume control

We could have made it very easy for ourselves by simply inserting a high-quality stereo potentiometer between the output of the audio DAC and the power amplifier. But that doesn't fit with the philosophy behind the audio player. A remotely-controllable motor potentiometer could have been a solution here, but these things are scarcely available to enthusiasts. That is why we chose a high-quality, solid-state solution in the form of a special IC from Texas Instruments, the PGA2311. This IC excels because of its low noise and distortion, and it works happily off a symmetrical 5-V power supply. The official title of this IC is 'Digitally Controlled Analog Volume Control'; the control of this IC takes place




## PROJECT INFORMATION

volume control | high-end | RC5 remote control | audio-DAC | stand-alone

entry level
➡ intermediate level
expert level

5 hours approx.

Soldering equipment, programming adapter

€100 / £95 / $125 approx.

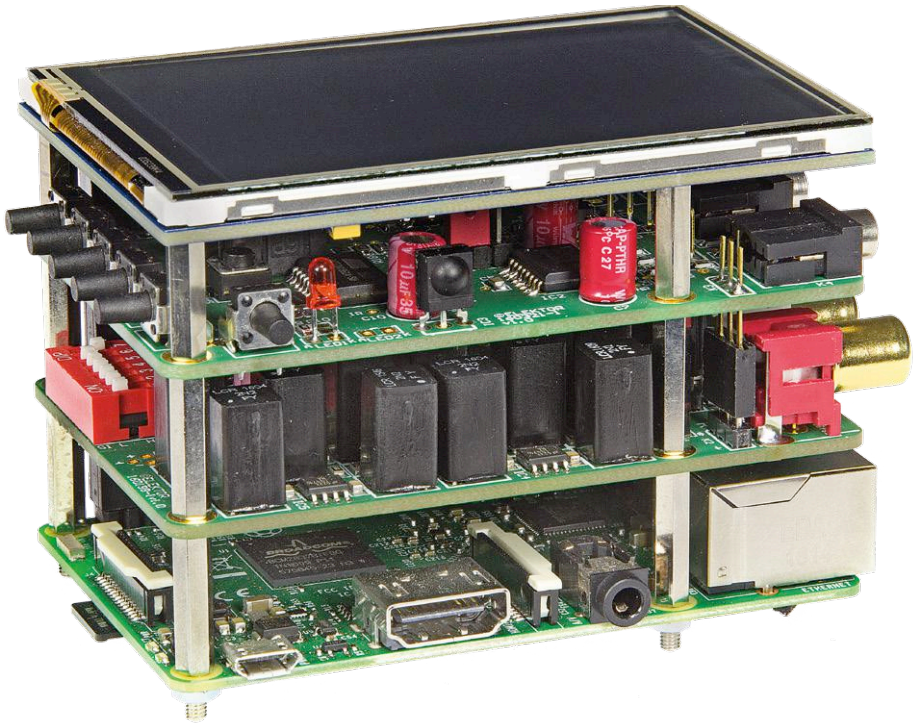


Figure 1. The 'Big DAC' in full glory.

across a serial interface connection. Additional features of this IC include a zero-crossing detector and a mute function. For the control we use a microcontroller from Atmel's ATtiny series.

## General design

Because this volume control was designed, in the first instance, to be used with the audio-DAC from the previous edition, its circuit board has been designed such that it can simply be plugged into the DAC circuit board. This way a thick sandwich of Raspberry Pi, audio-DAC, volume control and display is created (disrespectfully called a 'Big DAC', after a well-known product from the fast-food chain with the Scottish name), as can be clearly seen in **Figure 1**.

With the aid of half a dozen pushbuttons (a seventh button is used to store the default volume setting), Volumio and/or another application that is running on the RPi can be controlled. In addition, there is (obviously) provision for using an IR-remote control that uses the standard RC5 protocol.

And now then the good news for all those readers who are searching for an 'independent' usable volume control (that is, without RPi and audio-DAC): that is perfectly possible with the design described here! You can even omit a few parts...

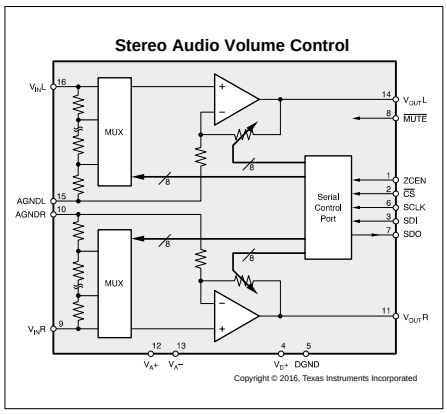



Figure 3. Amplifiers with variable gain are used for the volume control.

## The schematic

We've drawn the complete schematic for the volume control in **Figure 2**. You may have expected us to describe the circuit as a block diagram first, but that is not necessary — to the untrained eye the schematic looks complicated, but appearances can be deceiving.

We attempt to deal with the description systematically, and start with the beginning: the power supply (after all, without a power supply even the most artistically assembled circuit board is useless).

## The power supply

If the volume control is used as an 'add-on' to the audio-DAC then a separate power supply is not necessary. For the volume controller, the required (analog) +4.74 V and −4.81 V (abbreviated in the schematic to +4V7_A and −4V8_A) are supplied by the DAC board; the same is true for the +4.7 V for the microcontroller (+4V7_D in the schematic). These power supply voltages enter via connector K7, which is soldered to the bottom side of the circuit board (refer later on) and connects with K6 on the DAC board. The +3.3 V, which is only required for the inverters (IC4) that drive the GPIO pins of the expansion connector, is generated by the Raspberry Pi (pin 1 of the expansion connector).

The power supply voltage for the PGA2311 is filtered by the common-mode filter coil L6 and inductors L2, L3 and L4. Filter inductor L5 filters the digital power supply voltage for the microcontroller (IC1) while L7 assumes this task for the 3.3-V power supply. Inductor L1 ensures sufficient separation between the digital and analog power supply voltages for the PGA2311. Inductor L4, finally, has a very important job: it prevents the creation of an RF ground-loop, because both the power supply ground (K7) and the ground connections of K2 and K3 are already connected together on the DAC board.

When used as an add-on to the DAC board, connector K8 and capacitors C15 and C16 do not need to be fitted.

On the other hand, when using the volume control for other applications, the components IC4, C17, C18, L7, K2, K3 and K7 do not need to be mounted. Additionally, in this case it is also best to replace L4 with a 0-Ω resistor. Note: for stand-alone use, a separate, symmetric 5-V power supply is required!

## The PGA2311

For those who are interested, in **Figure 3** we have drawn the internals of the PGA2311. It is immediately clear that we have two (because it is stereo) amplifiers, the amplification of which are set by a set of (serial) control inputs. The control signals ($\overline{CS}$, SCLK, SDI and $\overline{MUTE}$) come from microcontroller IC1.

A brief remark about the connection of ZCEN (Zero Crossing Enable). The PGA2311 has a zero-crossing detector; the idea behind this is (when this function is enabled) that the change in volume only takes effect after the next positive-going zero-crossing of the audio

input signal. In this way spurious audio artifacts as a consequence of the change in volume can be minimized. On the circuit board ZCEN can be connected via a jumper to either +4.7 V or ground, so that you have the choice whether to enable or disable this functionality.

For a more detailed description of the inner workings of the PGA2311 we refer you to the datasheet [1]. Between the microcontroller and the PGA2311 we provided test points on the circuit board for the serial control signals (~~CS~~, SDI, SCLK and ~~MUTE~~).

The input and outputs go via resistors

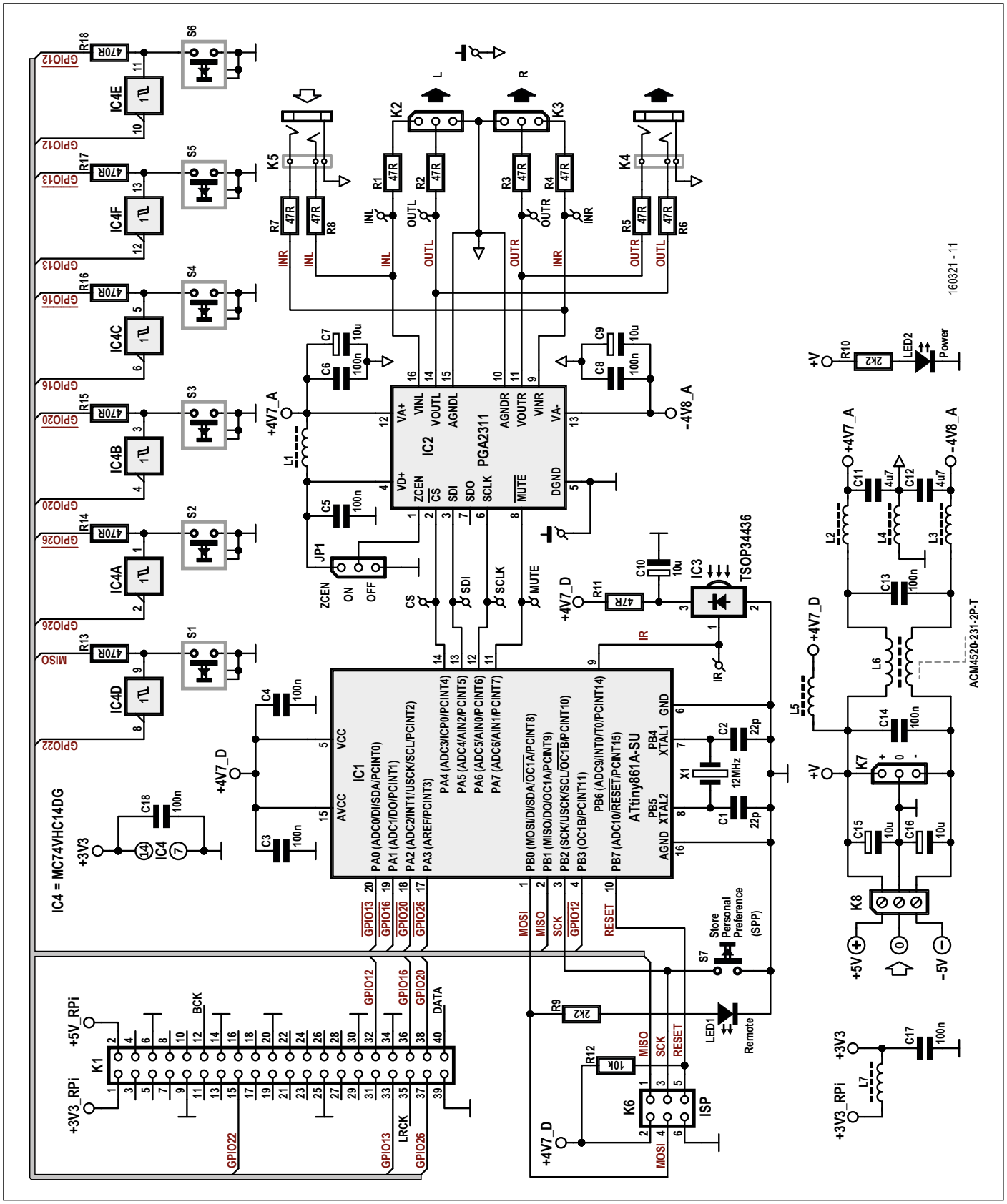



Figure 2. Unsurprisingly, at the heart of the circuit sits a microcontroller.

of 47 Ω (R1 through R8) to two different connectors. A pair of so-called stack-through-connectors (K2 and K3) connect the inputs and outputs of the DAC and the volume control (in the same was as is done for the power supply).

The inputs and outputs of IC2 also go to two 3.5-mm audio connectors (K4 and K5) on the circuit board. This means that the unprocessed DAC signal is available using a 3.5-mm jack (K5) or via the cinch connectors on the DAC board. The output signal from the volume control is available on K4.

When the board is used as an 'independent' volume control, K5 is the input for the PGA231 and K4 is the output. The 47-Ω resistors protect the outputs from

## COMPONENT LIST

### Resistors

R1,R2,R3,R4,R5,R6,R7,R8,R11 = 47Ω, 125mW, 1%, SMD 0805
R9,R10 = 2.2kΩ, 100mW, 5%, SMD 0805
R12 = 10kΩ, 100mW, 5%, SMD 0805
R13,R14,R15,R16,R17,R18 = 470Ω, 100mW, 5%, SMD 0805

### Capacitors

C1,C2 = 22pF, 50V, 5%, C0G/NP0, SMD 0805
C3,C4,C5,C13,C14,C17,C18 = 100nF, 50V, 10%, X7R, SMD 0805
C6,C8 = 100nF, 25V, 5%, C0G/NP0, SMD 1206
C7,C9,C10,C15,C16 = 10µF, 35 V, 0.04Ω, 6.3mm max. diam., pitch 2mm or 2.5mm, Würth Elektronik 870055673001 (WCAP-PTHR Series)
C11,C12 = 4.7µF 25V, 20%, X5R, SMD 0805

### Inductors

L1,L2,L3,L4,L5,L7 = 600Ω @ 100MHz, 0.15Ω, 1.3A, SMD 0603, Murata BLM18KG601SN1D
L6 = ACM4520-231-2P-T (TDK), 2 x 0.05Ω, 230Ω @ 100MHz, 2.6 A, SMD

### Semiconductors

LED1 = red, T-1, 3mm
LED2 = green, low-power, SMD 0805
IC1 = ATtiny861A-SU, SOIC-20, programmed with firmware # 160321-11 (or 160321-12)
IC2 = PGA2311UA, SOIC-16
IC3 = TSOP34436, 3-pin epoxy package
IC4 = M74VHC14DG, SMD SOIC-14 (ON Semiconductor only)

### Miscellaneous

K1 = 40-way (2x20) GPIO stacking socket (female), extra long
K2,K3,K7 = 3-way bus strip, vertical, 0.1" pitch (mounted on the bottom side of the board), with long stack-through pins
K4,K5 = 3-way stereo jack, 3.5mm, PCB mount, e.g. Lumberg KLBR 4
K6 = 6-way (2x3) boxheader, vertical, 0.1" pitch
K8 = 3-way PCB screw terminal block, 0.2" pitch, 630V
JP1 = 3-pin pinheader, vertical, 0.1" pitch
LED1 = 2-pin pinheader, vertical, 0.1" pitch
JP1 = jumper, 0.1" pitch
S1,S2,S3,S4,S5,S6 = pushbutton, right angle, SPST, FSMRA4JH TE Connectivity
S7 = pushbutton, SPST, FSM4JRT TE Connectivity
X1 = 12MHz miniature quartz crystal, 5x3.2mm, SMD, Cload 18pF
4 pcs 17mm M2.5 spacer (05.12.173 Ettinger)
4 pcs 6mm M2.5 bolt
4 pcs M2.5 nut
PCB 160321-1 v1.1

### Optional

To support the LC display (instead of 4 bolts):
4 pcs 14mm M2.5 spacer (Ettinger 05.12.143)

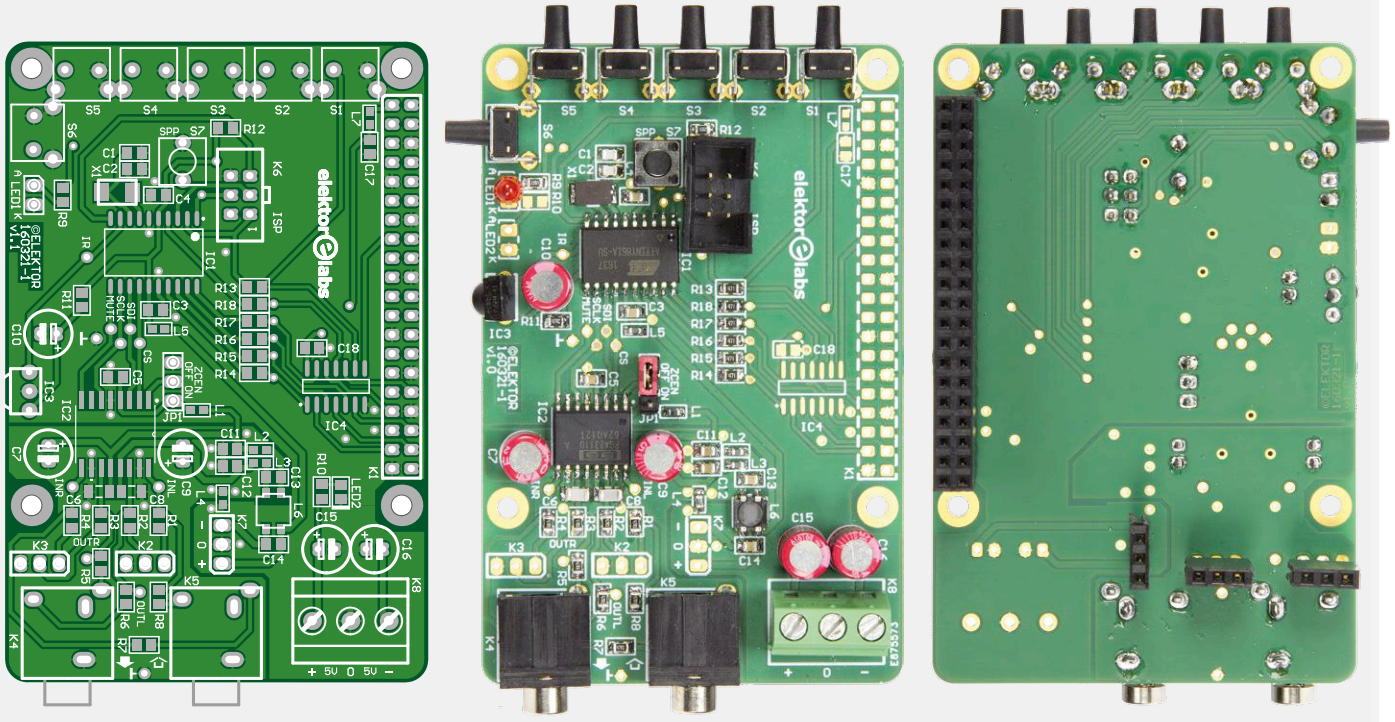


Figure 4. The circuit board for the volume control fits perfectly on the circuit board for the audio DAC.

capacitive loads and create a separation between the various connectors.

### Construction

For the volume control we designed a circuit board with dimensions that are identical to those of the Raspberry Pi and the audio DAC (**Figure 4**). The board can be sandwiched with the RPi and the audio DAC; after that the display can be plugged into the entire assembly. **Figure 5** gives an impression of the complete assembly.

Note: for connectors K2, K3 and K7 (the so called stack-through connectors), parts with extra long pins are used; these have to placed some distance away from the circuit board, otherwise they will not fit into the corresponding headers of the DAC board. And, naturally, you have to make sure that the connector bodies are fitted nice and parallel to the circuit board. Mutatis mutandis this is also true for the expansion connector K1, which is also fitted on the bottom side of the circuit board.
The circuit boards are firmly attached together using male/female standoffs; the same applies to the display.



Figure 5. Here you can clearly see how the sandwich is put together.

### The microcontroller

In order to get the PGA2311 to do what it needs to do, we need a microcontroller that sends the appropriate data to the serial interface of the volume controller. Here we use an ATtiny861A — a 20-pin IC with just enough I/O for the serial interface, seven pushbuttons, the IR receiver (for the remote control) and an LED. The latter (LED1) is a through-hole version that, if desired (using a PCB header on the board and a couple of wires), can potentially be mounted somewhere else. All the ISP I/O lines are also in use (except Reset); when you are programming the microcontroller (via connector K6) you have to make sure that your fingers are away from the pushbuttons! For the IR receiver we used a read-made solution in the form of the TSOP34436 (IC3); the output of which is also available as a test point. The IR protocol is RC5, for which there is an extremely handy command in Bascom (which is what we used when writing the firmware):

```
Getrc5(Address, Command)
```

Two version of the firmware are available, depending on the version of Volumio that you are using. Both versions are part of the free download [3]. We will describe both versions below.

## High-end volume control on a small surface area

### Firmware 160321-12

This version of the firmware was developed with Volumio 2 in mind [2]. It is provided with a plug-in for the GPIO buttons (**Figure 6**). Nearly all GPIOs can be linked to one for the following functions: play/pause, volume+/–, previous, next and shutdown. Because our DAC does not have its own volume control, four functions are used to operate Volumio; however for the benefit of other DACs that have a volume control we have also connected the functions volume+ and volume–. For these functions nearly any of the GPIO can be freely selected – provided that they are not already used for $I^2S$ or the 3.5" touch-sensitive display.

The GPIOs are provided with pull-downs by default. Our Volumio version (2.041 from 12.12.2016) does not (yet?) support the 3.5" LCD. In **Table 1** we summarize the functions of all the pins of the RPi expansion connector for you.

We use the RC5 commands from **Table 2**.

For the GPIOs logic '1' (High) corresponds to 3.3 V. We therefore need to use an MC74VHC14DG (IC4, hex Schmitt-trigger inverter — note: from ON Semiconductor) as a level shifter. The inputs of this IC can withstand voltages up to 7 V — perfect for our purpose. The corresponding I/Os of the microcontroller have to be High in the inactive state. Consequently the two I/Os that are configured as the inputs for the volume control (PA0/PA1), the pullups have to be turned on; the outputs that are configured for the four functions (PA2/PA3/PB1/PB3) have to be High in the inactive state.

| Table 1: Raspberry Pi Expansion Connector Pin Function | | | | |
|---|---|---|---|---|
| **K1** | **RPi (v3)** | **160321-12** | **I²S** | **3,5" LCD** |
| 1 | 3,3 V | 3,3 V | | 3,3 V |
| 2 | 5 V | | | 5 V |
| 3 | GPIO2 (SDA1) | | | NC |
| 4 | 5 V | | | 5 V |
| 5 | GPIO3 (SCL1) | | | NC |
| 6 | GND | GND | GND | GND |
| 7 | GPIO4 (GPIO_GCLK) | | | NC |
| 8 | GPIO14 (TXD0) | | | NC |
| 9 | GND | GND | GND | GND |
| 10 | GPIO15 (RXD0) | | | NC |
| 11 | GPIO17 (GPIO_GEN0) | | | TP_IRQ |
| 12 | GPIO18 (GPIO_GEN1) | | BCK | NC |
| 13 | GPIO27 (GPIO_GEN2) | | | NC |
| 14 | GND | GND | GND | GND |
| 15 | GPIO22 (GPIO_GEN3) | S1/Play/Pause | | NC |
| 16 | GPIO23 (GPIO_GEN4) | | | NC |
| 17 | 3,3 V | | | 3,3 V |
| 18 | GPIO24 (GPIO_GEN5) | | | LCD_RS |
| 19 | GPIO10 (SPI_MOSI) | | | LCD_SI/TP_SCK |
| 20 | GND | GND | GND | GND |
| 21 | GPIO9 (SPI_MISO) | | | TP_SO |
| 22 | GPIO25 (GPIO_GEN6) | | | RST |
| 23 | GPIO11 (SPI_SCLK) | | | LCD_SCK/TP_SCK |
| 24 | GPIO8 (SPI_CE0_N) | | | LCD_CS |
| 25 | GND | GND | GND | GND |
| 26 | GPIO7 (SPI_CE1_N) | | | TP_CS |
| 27 | ID_SD | | | |
| 28 | ID_SC | | | |
| 29 | GPIO5 | | | |
| 30 | GND | GND | GND | |
| 31 | GPIO6 | | | |
| 32 | GPIO12 | S6/Shutdown | | |
| 33 | GPIO13 | S5/Volume up | | |
| 34 | GND | GND | GND | |
| 35 | GPIO19 | | LRCK | |
| 36 | GPIO16 | S4/Volume down | | |
| 37 | GPIO26 | S2/Previous track | | |
| 38 | GPIO20 | S3/Next track | | |
| 39 | GND | GND | GND | |
| 40 | GPIO21 | | DATA | |

| Table 2: RC5 commands | |
|---|---|
| Play/pause | RC5 command 1 (Digit entry) |
| Previous track | RC5 command 2 (Digit entry) |
| Next track | RC5 command 3 (Digit entry) |
| Shutdown | RC5 command 12 (Standby) |
| Mute/de-mute | RC5 command 13 |
| Personal preference settings | RC5 command 14 |
| Increase sound volume | RC5 command 16 |
| Decrease sound volume | RC5 command 17 |
| Shift sound balance to the right | RC5 command 26 |
| Shift sound balance to the left | RC5 command 27 |

The six pushbuttons (S1 through S6) are connected via resistors of 470 Ω to the I/Os of the microcontroller. When one of these buttons is pushed the resistors prevent a short-circuit at the output of the µC, while at the same time the corresponding inverter is activated by making its input Low. The value of the resistors is low enough to make the inputs of the µC logic low.

When pushbutton S7 is pressed the volume level that is currently set will be stored as a personal preference. This button is not with the other six near the edge of the circuit board because it is likely it will be used only sporadically.

LED1 flashes briefly four times when S7 is pushed, to indicate the present setting of the volume level is being written to EEPROM. After switching on this setting is also used as the initial value.

When a new microcontroller is used for the first time, a default volume level of -20 dB is set. This can be 'surprisingly' loud — you are warned!

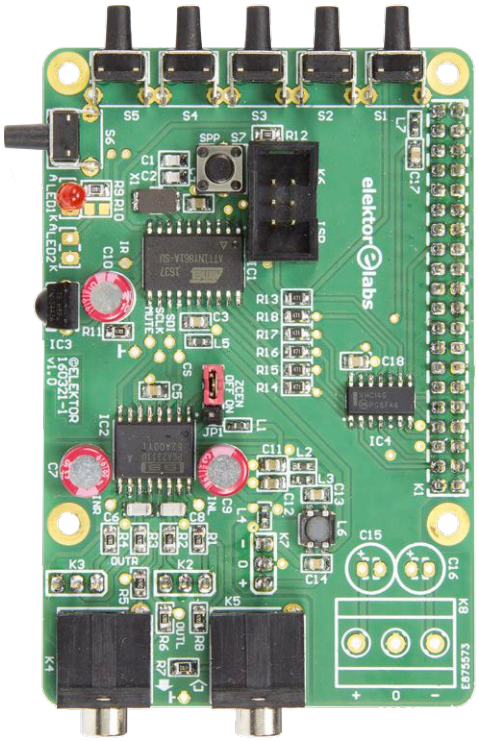
LED1 indicates that an RC5 command has been received — but that does not necessarily have to be one of the commands actually supported by the volume control. The IR receiver (a Vishay type suitable for TC5) is connected to PB6 because the Getrc5 command uses TIMER0 and the TIMER0 interrupt. If you are going to develop your own software for another IR protocol, then do make sure you use the correct IR receiver. In any case, the connections for the receiver are fairly common.

While we were developing the software we noticed something strange: when the 'divide clock by 8' is turned off, the Getrc5 statement doesn't work very well. This means that the 'ckdev8' fuse has to be enabled when programming the fuses in the microcontroller (see **Figure 7**)! For this reason we have increased the crystal frequency for the microcontroller to 12 MHz (X1 in the schematic).

### Firmware 160321-11

This version of the firmware was initially developed for use with Volumio 1.55 and our Raspberry Pi Audio DAC (ref Elektor July & August 2017). However, this version can also be used with Volumio 2 — or for entirely other applications. Note: when you use the volume control as a stand-alone device, you need a separate symmetrical 5-V power supply! In order to prevent ground loops we strongly suggest using an isolated power supply;

additionally L4 has to be replaced with a 0-Ω resistor (0603) or a wire link.
If K4 and K5 are fitted and the appropriate cables are used, the volume control can be inserted into any analog audio signal path ($V_{max}$ amounts to 2 V). In this way you can provide any amplifier with a remote volume control.
It is also possible to use connectors K2 and K3, instead of the 3.5-mm audio connectors K4 and K5. In that case circuit board headers have to fitted on the board (or the connecting wires can be soldered directly to the circuit board).
Pushbuttons S1 through S6 are used only for changing the volume settings; these functions are, together with the RC5 commands you can find summarized in **Table 3**.



Just as with the other firmware version, here S7 is also used to store the current volume setting as a personal preference in EEPROM.

When the mute function (IC2 pin 8) is activated (via software or hardware), the internal preamplifiers in the PGA2311 are disconnected from the outputs and VOUTL and VOUTR are connected to ground via 10-kΩ resistors. This can increase the noise in a connected amplifier. Also when the mute function is activated an internal offset calibration is carried out.

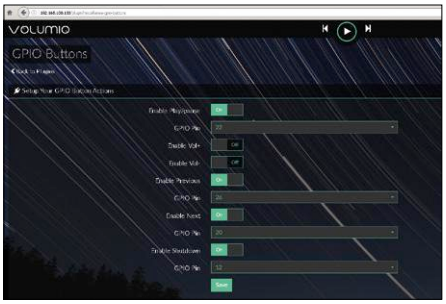



Figure 6. The Volumio GPIO plugin.

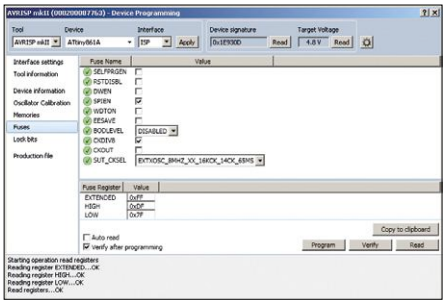


Figure 7. Must-know: the programming of the fuses.

## Volumio Mk. 2

At the time of writing, the small 3.5" touch screen is not (yet) supported in Volumio 2. When the plugin for supporting the original Raspberry Pi display is installed, a standard monitor can be connected to the HDMI connected of the RPi. Another hint related to the installation of this plugin: do not connect a keyboard to the RPi. If you are using a dongle for

| Table 3: Pushbuttons and RC5 commands (firmware 160321-11) | | |
|---|---|---|
| S1 | personal preference | RC5 command 14 |
| S2 | balance to the left | RC5 command 27 |
| S3 | balance to the right | RC5 command 26 |
| S4 | volume down | RC5 command 17 |
| S5 | volume up | RC5 command 16 |
| S6 | mute | RC5 command 13 |

| Table 4: Selected measurement results | |
|---|---|
| **Power supply current** | |
| LED1 off | 11 mA |
| LED1 on | 12 mA |
| **Maximum output voltage (1 kHz, THD = 0.1%)** | |
| 0 dB gain | 2.3 V |
| 20 dB gain | 2.6 V |
| –20 dB gain, (3.9 V in) | 0.39 V |
| **THD+N** | |
| 1 kHz, B = 22 kHz | 0.0003% (0 dB, 2 V out) |
| 1 kHz, B = 80 kHz | 0.00042% (0 dB, 2 V out) |
| 20 kHz, B = 80 kHz | 0.0022% (0 dB, 2 V out) |
| 1 kHz, B = 22 kHz | 0.0016% (20 dB, 2 V out) |
| 1 kHz, B = 80 kHz | 0.0025% (20 dB, 2 V out) |
| 20 kHz, B = 80 kHz | 0.0035% (20 dB, 2 V out) |
| **IMD (50 Hz : 7 kHz = 4:1)** | |
| 0 dB gain, 2 V in | 0.0007% |
| 20 dB gain, 200 mV in | 0.0038% |
| **DIM (3.15 kHz square wave + 15 kHz sine)** | |
| 0 dB gain, 2 V in | 0.0009% |
| 20 dB gain, 200 mV in | 0.0008% |
| **Bandwidth (–3 dB)** | |
| 0 dB, output triangle | 1.1 MHz |
| 20 dB | 690 kHz |
| **Crosstalk** | |
| 1 kHz | < –100 dB |
| 20 kHz | < –90 dB |

the mouse/keyboard combo it is better to remove that too. It is recommended to start the installation from within the WebUI.

### Measurements

Just as with the audio DAC for the RPi we have performed a few measurements on our volume control; the most important results are summarized in **Table 4**.

We have also made a few plots with our Audio Precision Analyzer, both for the audio DAC as well as the volume control. These are further explained in the sidebar. ⏮

(160321)

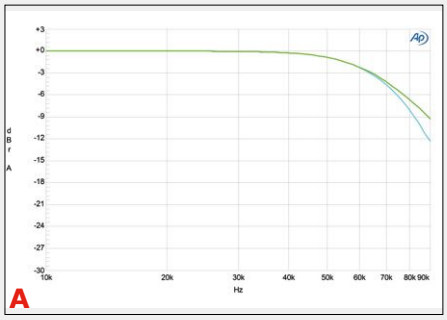
### Web Links

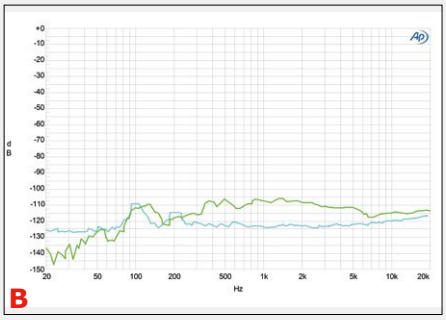

[1] Datasheet PGA2311: www.ti.com/lit/ds/symlink/pga2311.pdf

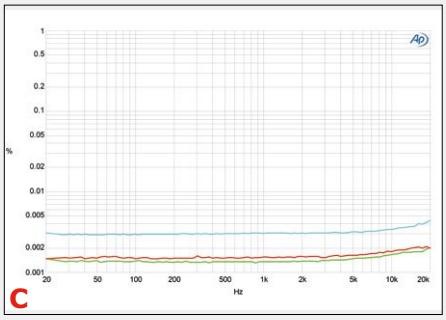
[2] Volumio: https://volumio.org

[3] Download: ww.elektormagazine.com/160321


## Measurements on the Big DAC




One of the showpieces of our former audio lab in Holland is a full blown Audio Precision Analyzer — which fortunately has survived the recent move to Aachen intact. Naturally we have tested the 'Big DAC' using our analyzer; below you can see some of the results.

### Audio-DAC

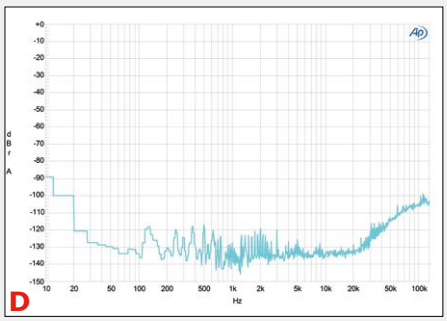
**Plot A** shows the amplitude as a function of the frequency with a stepped sine signal (70 frequencies, full scale) at a sampling frequency of 192 kHz. The top (green) curve was measured with a steep output filter; the bottom (blue) curve with is less steep filter. The corner frequency amounts to 63.5 kHz (64.5 kHz in theory). At 90 kHz the amplitude is -9.2 dB (green) and -12.3 dB (blue) respectively. At 45 kHz and 51 kHz both curves are equal (-0.5 dB and -1 dB respectively).

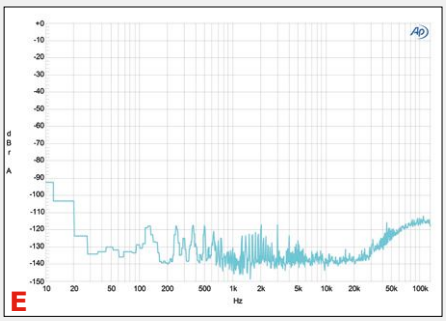
**Plot B** gives an impression of the channel separation as a function of frequency. The top (green) curve is the crosstalk from the left channel to the right channel. In this direction the crosstalk is slightly larger than the other way around. This is because the length of the signal traces on the circuit board to the output connectors are not the same. The trace for the right channel is longer, with the consequence of more crosstalk from left to right. Nevertheless, this is a good result for such a compact design.

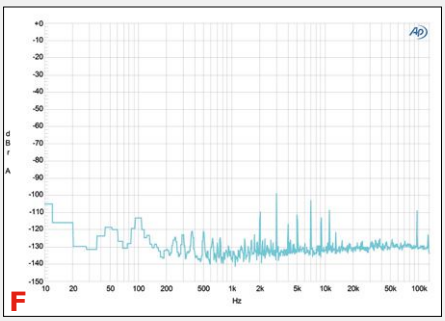
**Plot C** shows the total harmonic distortion plus noise (THD+N) as a function of frequency at sample frequencies of 48, 96 and 192 kHz. In all cases the bandwidth is 80 kHz. Incidentally, at a bandwidth of 22 kHz the THD+N is much lower when the sampling frequency is 48 kHz.

**Plot D** shows the FFT analysis of a 1 kHz full-scale sine at a sampling frequency of 32 kHz. Remarkable is the increase of the noise just above the audio bandwidth. This could be noise shaping. At a bandwidth of 22 kHz the THD+N amounts to 0.0007% (at a bandwidth of 80 kHz that is 0.012%). A large number of noise spikes are visible, mainly caused by the RPi, which is just below the DAC. The second and third harmonics of the sample frequency are just visible.

**Plot E** shows the same as Plot D, but then at a sampling frequency of 44.1 kHz. The high-frequency noise is here less when compared to the previous plot. The second and third harmonics of the sample frequency are a little bit more pronounced here; nevertheless the harmonic distor-



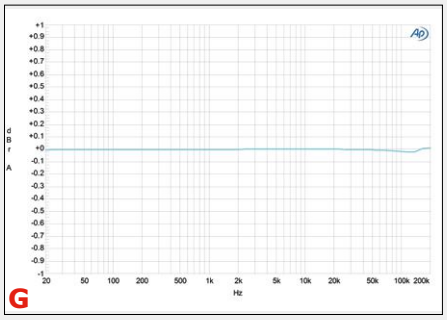
## FROM THE STORE

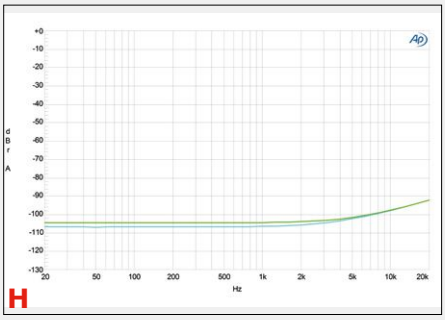
→ **160321-1**
Bare printed circuit board

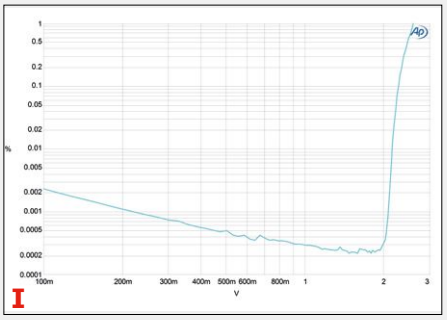
→ **160321-91**
Assembled module
(headers supplied separately — DIY soldering)

→ **160198-91**
Assembled Audio DAC module
(headers supplied separately — DIY soldering)

→ **SKU 17631**
Raspberry Pi 3 (model B)




tion (without noise) amounts here to only 0.00019%.

**Plot F** finally shows the FFT analysis of a 16-bit full-scale sine of 1 kHz at a sampling frequency of 32 kHz, 'on the fly' converted to 24-bit/96-kHz (as is possible in Volumio 1.55). Remarkable is the absence of the high-frequency noise that is visible in Plot D.

### Volume control

**Plot G** again shows the amplitude as a function of frequency at a gain of 0 dB and an output level of 2 V. The maximum frequency of the generator in the Audio Precision Analyzer goes up to 200 kHz. The small 'bump' around 100 kHz is not significant (note the vertical scale!).

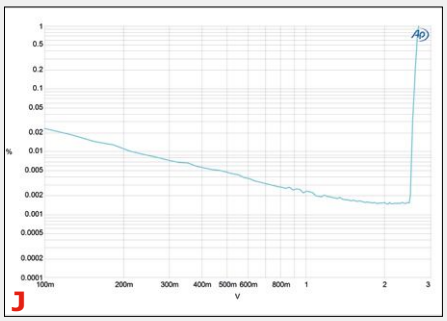
**Plot H** shows the crosstalk as a function of frequency. The difference between the two curves amounts to about 2 dB — the circuit therefore behaves almost completely symmetrically.

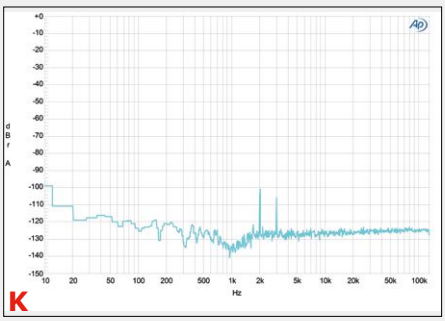
**Plot I** makes the THD+N visible as a function of output level at a gain of 0 dB. The bandwidth is here reduced to 22 kHz to better show where the circuit begins to clip. At this low power supply voltage the maximum undistorted output voltage is about 1.9 V. Above this level the THD increases quickly (0.1% at 2.3 V and 1% at 2.6 V).

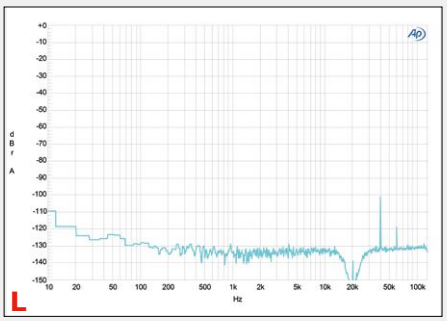
**Plot J** is the same is Plot I, but now at a gain of 20 dB. The maximum undistorted output voltage is now surprisingly much higher, at 2.48 V (THD = 0.0015%) the circuit begins to clip. Apparently at lower gain this occurs earlier in the input stage because the input voltage is then higher.

**Plot K** shows the FFT of a 1-kHz signal at a gain of 20 dB and an output voltage of 2 V. Here only the second and third harmonics are visible, corresponding to a THD (without noise) of 0.001%. (THD+N at 22 kHz bandwidth is 0.0016%, and 0.0025% at 80 kHz bandwidth).

**Plot L** finally is the same as Plot K, but now for a 20-kHz signal. Here too the second and third harmonics are visible. THD = 0.00086%; THD+N at 80 kHz bandwidth is 0.0035%.



# Temi

## A meteorological marionette

By **Lars Lotzenburger** (Germany)

Electronics is everywhere in daily life, and even the young cannot be spared the effects of the march of progress. This project illustrates how the latest technology can be used to update the traditional jumping jack toy: instead of pulling a string, 'Temi' is operated by pressing a button, whereupon he will display the current temperature or humidity.

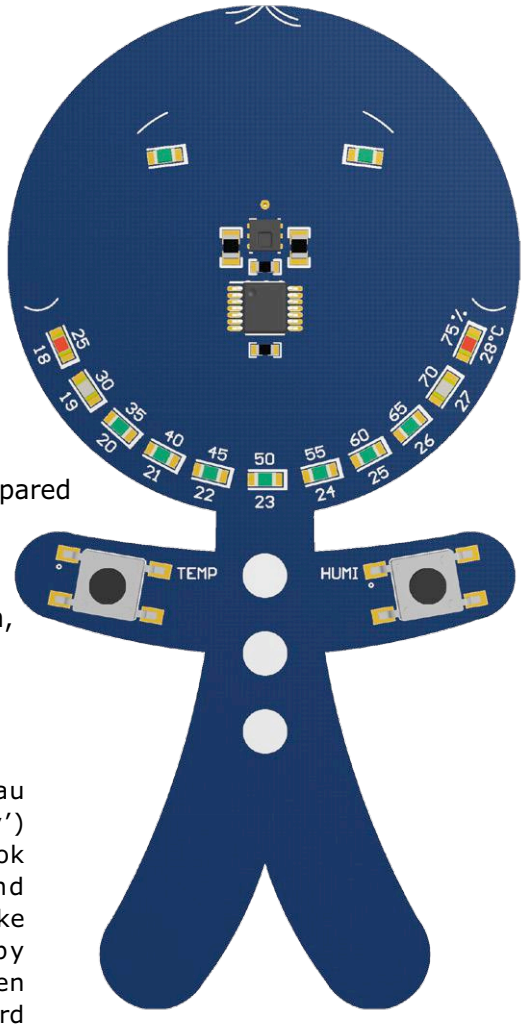



Figure 1. Rendering of Temi's printed circuit board.

It is a pity that an ever diminishing number of people are aware of the capabilities of modern electronic technology, and in particular microcontrollers, and the opportunities it can offer. Perhaps one reason behind this is that ICs and other special-purpose components have become increasingly tiny, and at the same time so complex that it is hard to understand their internal workings. But it is never too early to try to pique a young person's interest in technology, and the aim of Temi is to attract inquisitive but shy fingers.

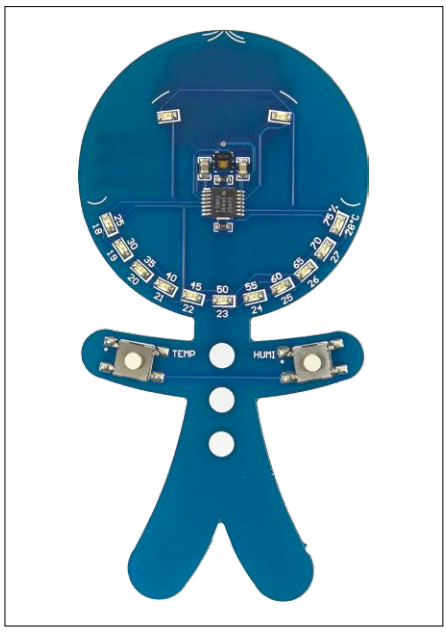


Figure 2. The assembled prototype.

### Temi

Temi (the name is a portmanteau of '**te**mperature' and 'hu**mi**dity') is therefore designed both to look attractive (see the images and printed circuit board) and to make the possibilities opened up by modern electronics accessible even to the very young. The circuit board is made to resemble a traditional jumping jack toy [1], and carries a microcontroller, an environment sensor and a sprinkling of LEDs. Replacing the pull-string there are two pushbuttons. Pressing one of the buttons will display either the current temperature or the current humidity using the LEDs: both these functions are provided by the TI type HDC1080 sensor chip [2]. Processing the outputs of this device and driving the display is the job of an MSP430G2402 low-power microcontroller [3], also from TI. The whole circuit is powered by a tiny lithium coin cell, which should last for years.

A look at the rendering of the printed circuit board in **Figure 1** shows the child-friendly component layout: two green LEDs form Temi's eyes, the sensor and the microcontroller make up the nose, and an arc of eleven LEDs forms his broad grin. The buttons for temperature, on the left, and humidity, on the right, are on the (stylized) hands. We think our assembled prototype in **Figure 2** looks rather dapper.

### A special circuit

All in all there are 13 LEDs in the circuit. Without some form of multiplexing, this would require 13 I/O lines on the microcontroller. Adding on the requirements for the buttons and for communication with the sensor, we soon come to the point where we will need to resort to a microcontroller with rather a lot of pins. Using a conventional multiplexing configuration for the 13 LEDs will not save us enough pins. For $n$ LEDs at least $2\sqrt{n}$ pins are needed, where $\sqrt{n}$ should be rounded up to the next integer. In our case we therefore need eight pins, because $\sqrt{13} \approx 3.6$, which rounds up to 4. It is possible, however, to use rather fewer pins, using a particularly clever trick called 'charlieplexing'. How this works is explained in the **text box**. Some readers may raise a skeptical eyebrow, but we assure you that we are not pulling your leg: there is even a Wikipedia article on the subject [4]

and, over ten years ago, the technique featured in its own Elektor article [5]. Looking at the circuit in **Figure 3** we can see that we manage to drive twelve LEDs using just four I/O pins. LED2 is the LED left over, and that is driven directly from its own port pin. Another saving in comparison to the conventional arrangement will also be apparent: the LEDs have no series resistors! In practice this works reasonably well as the output drivers of the microcontroller do not have a particularly low impedance. Bearing in mind the LED's threshold voltage and that the supply voltage is only 3 V, we find that the maximum current through an LED is only 20 mA, which it can handle comfortably.

It is also important to note that for the LEDs to be bright enough they must be high-efficiency types: the average current through one of the charlieplexed LEDs is only about 1.6 mA. In our prototype we used LEDs from Würth Elektronik: type 150080GS75000 for the green LEDs, and with the 'G' replaced by 'Y' or 'R' for the yellow and red LEDs respectively.

Unfortunately the microcontroller does not have a built-in $I^2C$ interface to communicate with the sensor. For this reason we therefore need to implement the interface 'manually' in software using bit-banging, taking advantage (as we also do in the case of the pushbuttons) of the selectable pull-up resistors on the microcontroller's inputs. The sensor IC has a very low current consumption, and, for the sake of simplicity, is powered via R1 from an I/O pin on the microcontroller. Together with C1 this resistor forms a lowpass filter that helps suppress interference and ensures a clean supply to the analog circuitry. The arrangement also allows the sensor to be turned on only when a reading is wanted and to be powered down completely in the quiescent state.

### Display and operation

Temi's mouth consists of an arc of eleven LEDs. The two LEDs at the extremities of his smile are red, the next two LEDs towards the center are yellow, and the remaining seven LEDs in the middle are green. This arrangement enables us to display relative humidity from 25 % to 75 % in steps of 2.5 %, and temperature from 18 °C to 28 °C in steps of 0.5 °C, which should be adequate for normal indoor use in temperate climes. We use a little trick to get double the resolution that you might expect using the eleven LEDs: a half-step is indicated by lighting two adjacent LEDs rather than one. So, for example, we indicate a temperature of 21.5 °C by lighting the LEDs corresponding to

## Charlieplexing

The basic idea behind charlieplexing is to connect two LEDs in antiparallel between each pair of I/O port pins. It must therefore be possible to switch each output to either ground or the positive supply: open-drain or open-collector outputs are not suitable. Using two pins we can drive one pair, or two LEDs; with three pins we can connect three pairs, or six LEDs; with four pins we can use six pairs, or 12 LEDs, and so on.

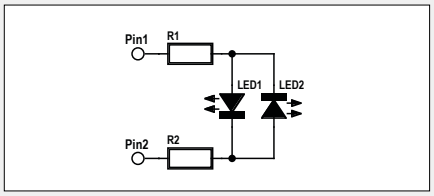
In the two-pin case there is no point in using charlieplexing as we can already drive two LEDs directly. For three or more LEDs the trick only works if the microcontroller has three-state outputs, that is, if the outputs can be set to a high-impedance state. This ensures that the unused outputs do not interfere with the two outputs that are being used to drive the LED in question.

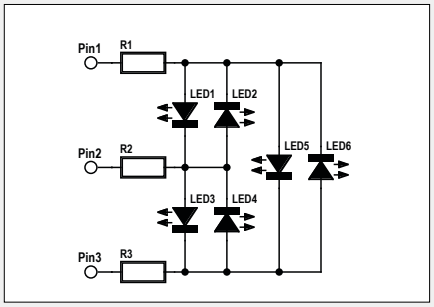
A couple of other points are worth noting. Using direct drive we can light not just one LED, but any combination of them simultaneously. Using normal multiplexing we can light up to $\sqrt{n}$ LEDs at a time, where n is the total number of LEDs: for example, using seven pins and an LED array with four rows of three columns each we can drive 12 LEDs with up to four (all in the same column) lit simultaneously. In this example the LEDs in the three columns would be lit in turn in consecutive time slices; to avoid flickering the column multiplex rate would have to be at least 3 x 30 = 90 Hz.

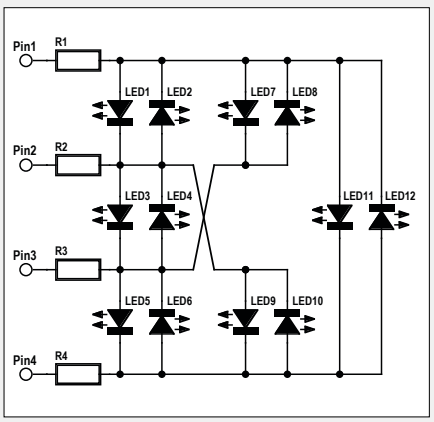
By contrast, in a charlieplexing arrangement, we can only light one LED at a time. This means that the multiplex rate has to be higher: for 12 LEDs being lit simultaneously we must run at a minimum of 12 x 30 = 360 Hz to ensure that flicker is not apparent to the human eye.

But that is not the end of the story: with direct drive each LED can receive its full rated current and so light brightly, but time-slicing the LEDs reduces their brightness as they are only turned on for a fraction of the time. In a normal multiplexing arrangement with, for example, three columns, the LEDs only receive on average one third of their maximum current. In the charlieplexing example that fraction reduces to one twelfth.

A small advantage of charlieplexing is that, whereas implementing normal multiplexing using a microcontroller can involve quite a bit of coding, charlieplexing can be implemented by setting up a table of the output bit patterns required for each LED and simply cycling through them.



both 21 °C and 22 °C.
The LED eyes normally light continuously when Temi is active: only when the battery voltage eventually starts to run low will they start to flash instead. To save power Temi only enters active mode when one of his buttons is pressed. A reading will then be taken (temperature if the left button was pressed, humidity if the right button was pressed) and the result displayed for five seconds. After that Temi will return to his quiescent state.

**Battery life**

Although in principle it would be possible to have Temi's coin cell provide continuous power for the microcontroller and its LEDs, he will not survive long in such a configuration. Instead Temi is designed to spend most of his life hanging on a wall asleep, in which case the battery will last practically forever. A typical CR2032 lithium cell made by Varta has a claimed usable capacity of around 170 mAh, and Varta guarantees a shelf life of ten years with a self-discharge of 1% per year. We can now make a rough calculation to see how long Temi's battery will last in actual use.
First let us consider the quiescent current consumption of the electronics. The sensor is completely powered down when the processor is asleep, and the processor itself goes into deep sleep mode after five seconds of inactivity. In this mode almost all of the processor's internal modules and clocks are deactivated and it can only be woken up as a result of a reset or a dedicated interrupt (here, when pressing a button causes the level on a specified input pin to change). The result is an extremely low quiescent consumption of only 100 nA. The theoretical battery life in this mode is therefore 170 mAh / 0.1 µA = 1.7 x $10^6$ hours, or about 194 years. Even if we take into account leakage currents on the circuit board and through capacitors this result will not change much, and so as long as Temi is not disturbed we will be certain to achieve 10 years of battery life.
When a button is pressed Temi springs into life and the microcontroller, running at 1 MHz, will draw about 320 µA. To this we should add the 1.5 µA drawn by the sensor and up to 20 mA for the LEDs. In total, then, we should reckon on a current draw of about 20.3 mA for the next five seconds. A button press therefore discharges the battery by 5 s x 20.3 mA = 101.5 mAs = 28.2 µAh. So your little darlings can activate Temi 6030 times before you will need to change the battery. That corresponds to four presses a day for a good four years (by which time they will probably be able to change the battery for themselves!).
The microcontroller also measures the battery voltage and flashes Temi's eyes to warn when this falls below 2.8 V: the sensor is only guaranteed to work reliably down to 2.7 V.

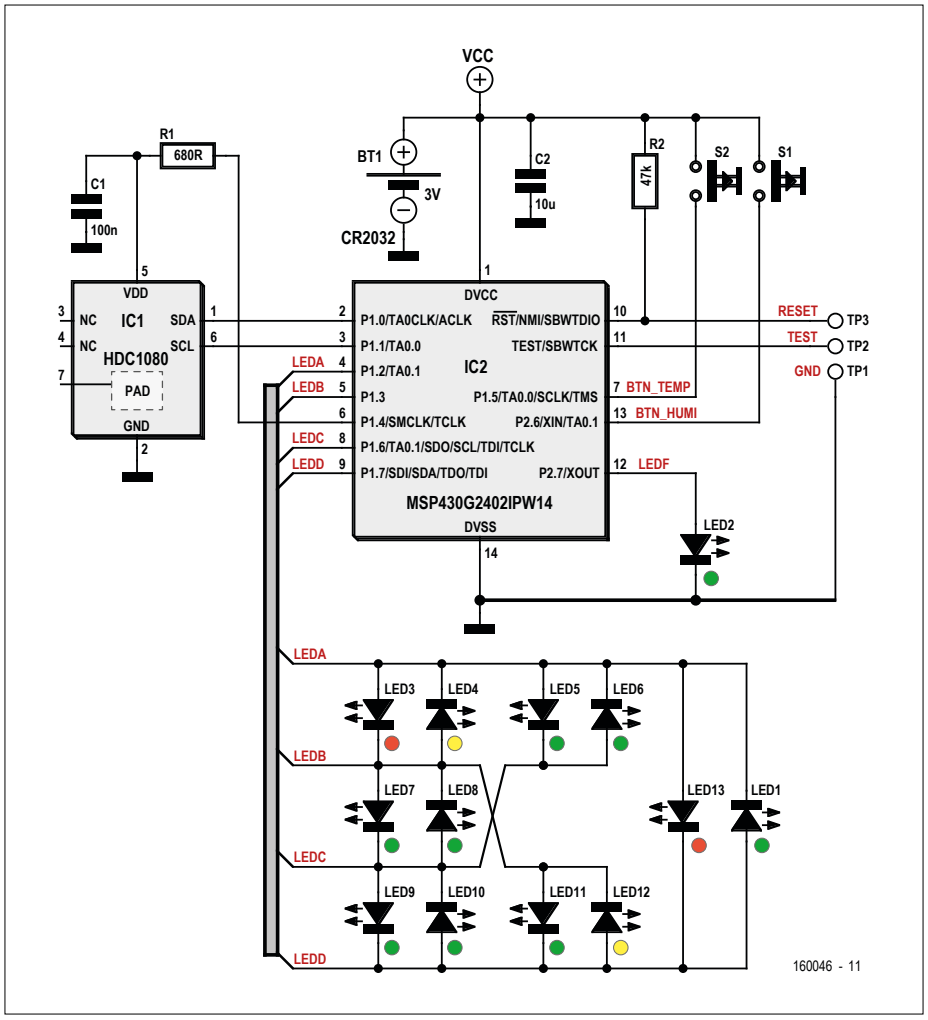



Figure 3. The circuit is reasonably straightforward: the sensor, the microcontroller, and 13 LEDs driven in a charlieplexed arrangement.

**Software and all that jazz**

The design files for the printed circuit board, the parts list and the software (both as source code and as a hex file that can be programmed directly into the processor) are available for free download from the Elektor web page accompanying this article [6].
The software is fairly straightforward and so only a couple of remarks are necessary. First, a macro is defined to light each LED, to isolate the complexities of the charlieplexing arrangement. Then a table is generated to convert sensor values to LED patterns. Pressing a button triggers a state machine with states STATE_STANDBY (deep sleep mode), STATE_TRIGGER (read battery status and start a measurement) and STATE_DISPLAY (show results on LEDs). The LEDs are driven in the main loop, with the measurement initiated from within the watchdog timer interrupt, which is called every 30 ms. This makes it simple to respect the timing constraints of the sensor, and the state machine can be advanced to its next state on each call. Between interrupts the processor enters sleep mode. The software was written using IAR Embedded Workbench [7].
The author used a standard low-cost

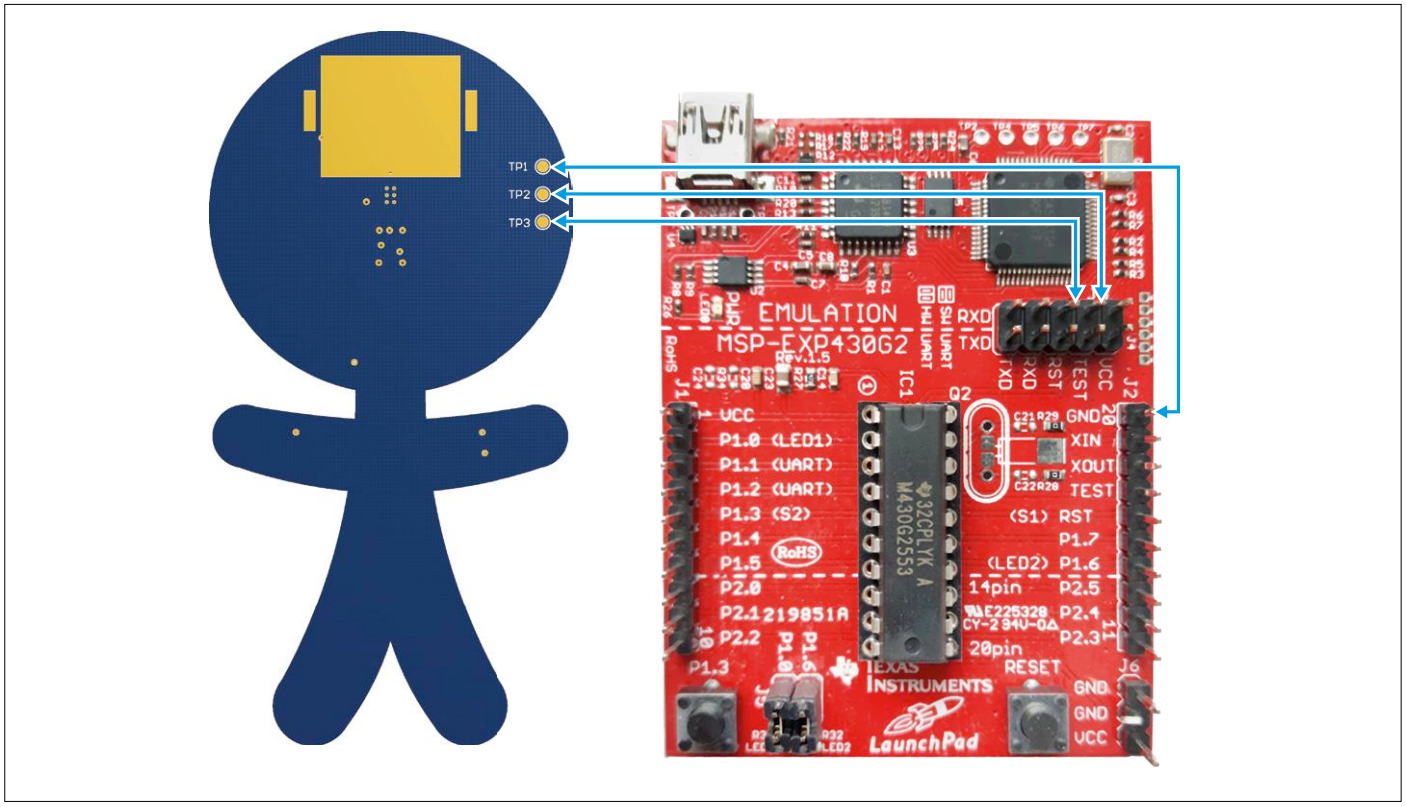



Figure 4. The connections between the Launchpad board and Temi (rear side) to allow his microcontroller to be programmed with the firmware.

MSP430 Launchpad [8] to program the software into the device, removing jumper J3 and connecting the programming signals directly to Temi (see **Figure 4**). This allows Temi's MSP430 to be programmed from the Launchpad using the 'Spy-Bi-Wire' protocol.

(160046)

**Web Links**

[1] https://en.wikipedia.org/wiki/Jumping_jack_(toy)
[2] Sensor: www.ti.com/product/HDC1080
[3] Microcontroller: www.ti.com/product/MSP430G2402
[4] https://en.wikipedia.org/wiki/Charlieplexing
[5] Elektor article on charlieplexing: www.elektormagazine.com/magazine/elektor-200607/18310
[6] www.elektormagazine.com/160046
[7] www.iar.com/iar-embedded-workbench
[8] www.ti.com/tool/MSP-EXP430G2

Advertisement

HAMMOND MANUFACTURING®

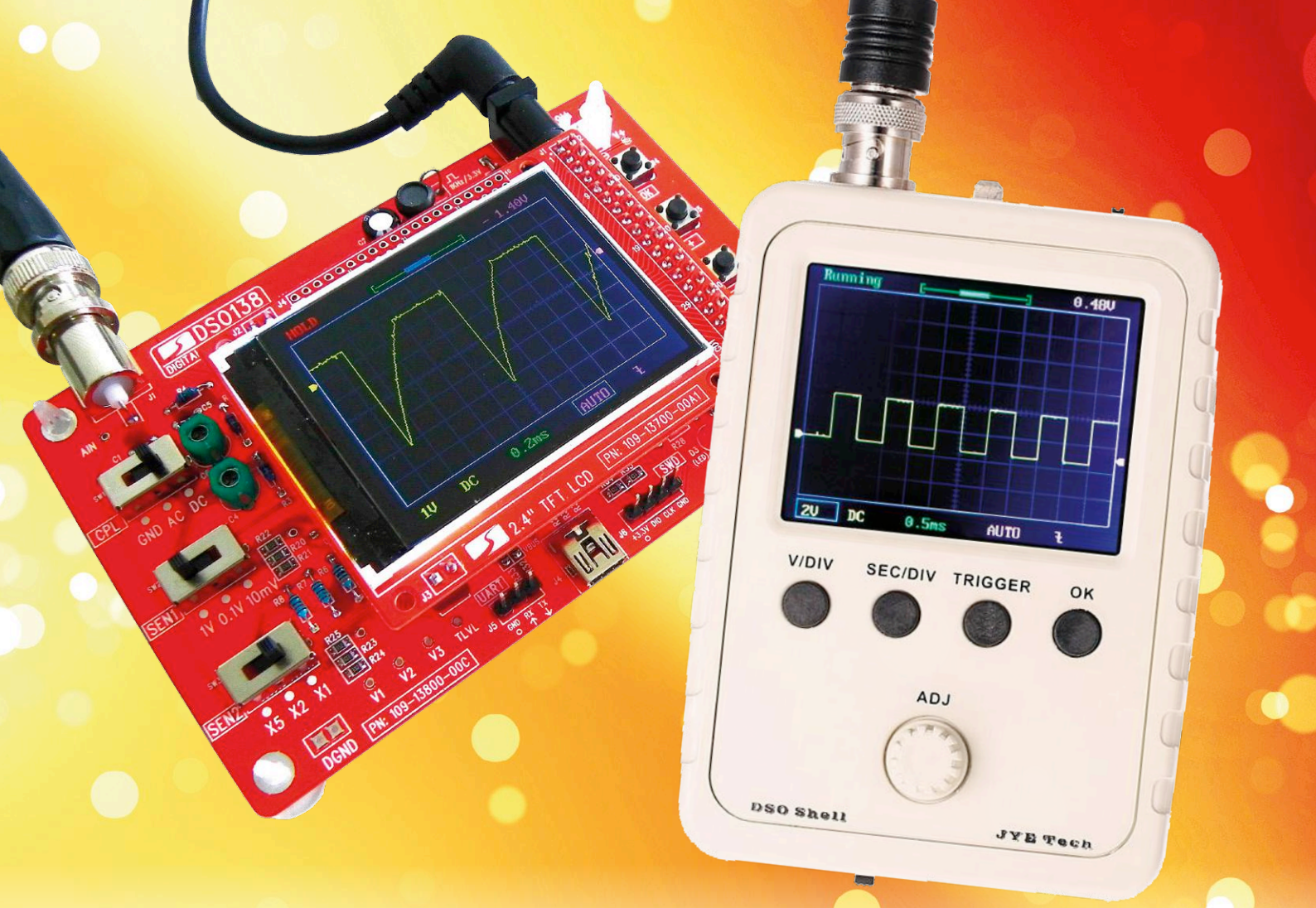
Plastic enclosures standard and miniature

www.hammondmfg.com/dwg8.htm
www.hammondmfg.com/1551USB.htm

01256 812812
sales@hammondmfg.eu








# Mini Oscilloscope

## Display your signals without breaking the bank

By **Alfred Rosenkränzer** (Germany)

Sometimes you want to equip a device with a display that can do a bit more than just show basic status information or a couple of numeric values. One option is the small oscilloscope modules available from suppliers in the far East, which are feature-packed and economical. In some cases a matching enclosure is available to make a complete portable mini-oscilloscope. We took a look at a couple of products made by JYE Tech.

### Features (common to both modules)

- Bandwidth: 0 Hz to 200 kHz
- Maximum sample rate: 1 MSa/s
- Input sensitivity: 10 mV/div to 5 V/div
- Vertical accuracy: <5 %
- Vertical resolution: 12 bit
- Timebase: 10 µs/div to 500 s/div
- Record length: 1024 samples
- Test signal: 1 kHz 3.3 V squarewave
- Hold function (freezes waveform)
- Store and recall waveform

A few weeks ago I was looking for a small oscilloscope module to build into a device to measure and display the current draw of an injector nozzle over time. A quick surf of the web turned up the DSO138 from JYE Tech. Its bandwidth of 200 kHz and sample rate of 1 MHz were enough for my application, and so I ordered one. The module was very cheap: only around £20 (US$25) shipped direct from China. Out of pure curiosity I therefore also splashed out on the DSO150, which is an oscilloscope kit with an enclosure. I will now relate my experiences with the two units.

# An oscilloscope module for pocket money

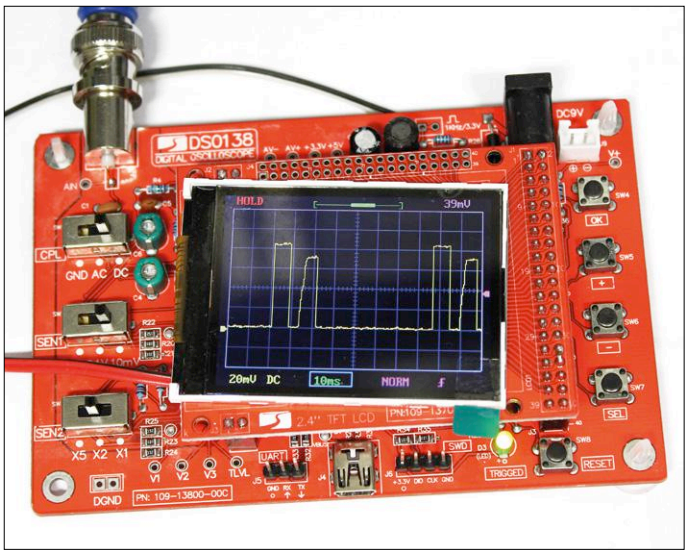



Figure 1. The ready-built DSO138 module. The screen shows a current measurement trace from an injector nozzle.

## DSO138

As **Figure 1** shows, the DSO138 module consists of a circuit board with a color LCD attached. Soldered to the board are all the user controls (slide switches and pushbuttons) and sockets for signal input and power. On the left-hand side you can see that the selection of input coupling and vertical scale is done in the simplest possible way using slide switches. Unfortunately the quality of these switches (at least in the unit I received) is so abysmal that changing the gain setting often results in the complete disappearance of the signal. The buttons on the right control other functions, and the user interface is generally well-designed and intuitive.

The color LCD panel has a resolution of 320x240 (QVGA) and the visible area has a diagonal of 2.4 inches (6.1 cm). Some low-end desktop scopes, much more expensive than this device, have a display with the same resolution, and so the device should be good enough at least for my application. Happily the device comes with a circuit diagram, which is more than you get with many items of test equipment these days. As a result it was easy to see how to add an external TTL trigger input.

The module operates from a 9 V power supply, drawing around 100 mA. A simple inverter circuit generates a negative rail, which, like the positive rail, is stabilized using a linear regulator. The microcontroller, an STMicroelectronics STM32F103C8 based on an ARM Cortex-M3 core, is powered at 3.3 V via another regulator.

The design uses the fast ADC built in to the microcontroller. It offers a resolution of 12 bits and an input voltage range from

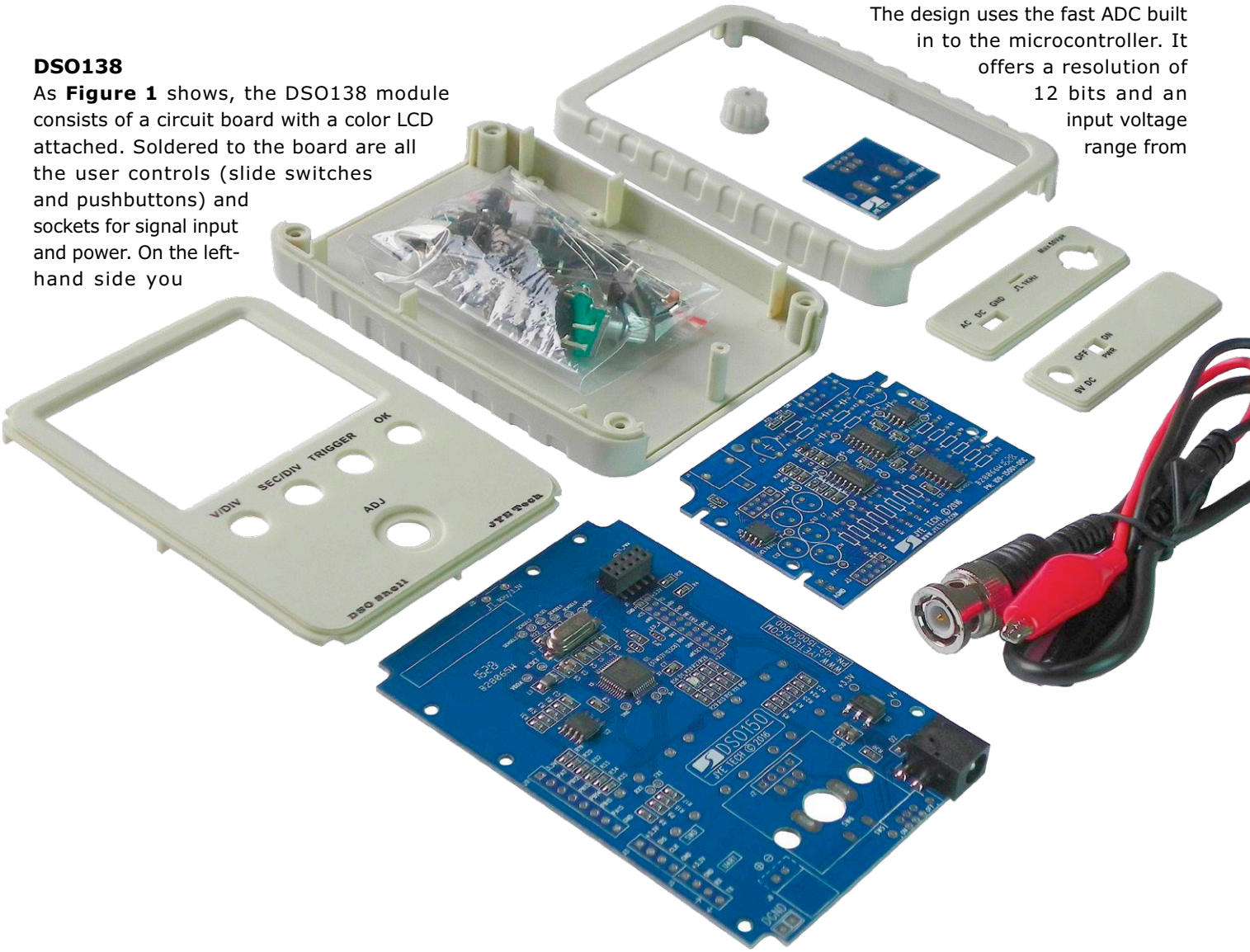





Figure 2. An almost full-scale sinewave at 1 kHz displayed on the assembled DSO150.

0 V to 3.3 V; the middle of this range, 1.65 V, corresponds to the zero position on the display. However, the display is already clipped when the unit is fed with a sine wave of much less than the maximum possible amplitude. This, in conjunction with the high resolution of the converter, allows the vertical offset to be adjusted. In other words, the vertical offset is not implemented in the analog preamplifier circuit, but instead is subtracted from the sample values in software.

The fact that only a relatively small part of the ADC's input range is used obviously makes the device more susceptible to interference and spikes, and these sometimes also affected the horizontal deflection. I tried to fix this by improving the decoupling of the circuit at various points, but without success. 'You get what you pay for' as the cliché has it, but in this case you do get quite a lot of hardware for your money, and its minor deficiencies can be tolerated. The DSO138 is not just available as a ready-made module: if you fancy doing some soldering you can also get it slightly more cheaply in kit form, with the SMD components already fitted.

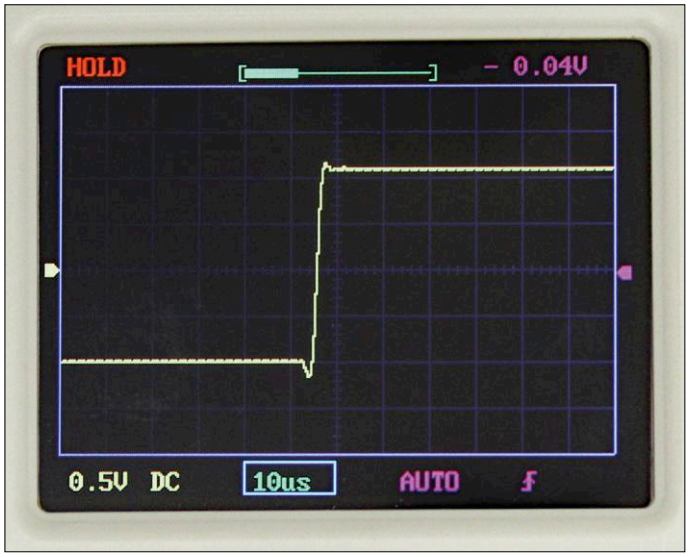



Figure 3. Edge of the 1 kHz test signal. Not as steep as on a 'real' oscilloscope, but nevertheless entirely usable.

**DSO150**

As mentioned above, as I was rootling around on the manufacturer's website [1] I noticed the DSO150 oscilloscope kit (see Figure 2). This device is a development of the DSO138, using the same display and microcontroller. However, this model is only available in kit form with an enclosure. The whole user interface apart from the switch to select the input coupling is driven using four buttons and a rotary encoder. The interface is straightforward and it is hard to see how it could be improved. This kit also comes with the SMD components already mounted. There are just a few leaded components that need to be soldered to the board and then the unit can be put together and tested. The instructions are of a high standard and none of this will present any difficulty to a hobbyist with a little experience. Because the device is supplied in kit form it is not subject to the various standards and regulations that apply when ready-assembled units are sold. The lack of a CE mark is not really a problem, as a manufacturer can always add it later to a finished unit.

I spotted the biggest shortcoming in the unit immediately after assembling it, when I first turned it on: the zero level shifts with changes in the power supply voltage. Since once again a circuit diagram is supplied it was easy to find the cause: unfortunately in the DSO150 the linear regulator to stabilize the supply for the input amplifier has been omitted! The offset for the ADC input is set using just a Zener diode, which does not provide a very well-regulated voltage. As before, you get what you pay for. Fortunately, however, there are leaded series resistors in the positive and negative supply lines, and it is a relatively easy job to replace these with 78L05 and 79L05 voltage regulators. The supply for the Zener diode is now moved to the regulated 5 V rail, and the zero level is stable.

As in the DSO138 the vertical offset is done in software in the microcontroller. The full-scale drive amplitude is 640 $mV_{PP}$, and so only about 20 % of the ADC's available input range is used. Compared to the DSO138 interference is much less visible on the DSO150, and occurs less frequently. It is also possible to add an external trigger input to the DSO150.

The unit is an excellent gadget for experimenting with and could make a good secondary oscilloscope for a hobbyist. After a bit of modification to the power supplies it works perfectly acceptably (see Figure 3), especially when you bear in mind that it only costs about £25 (US$30) including shipping from China. There is a chance that you will end up paying VAT and other import charges on top of this, as its cost may exceed the duty-free threshold, at least in Europe. ⏮

160466)

## Internet Link

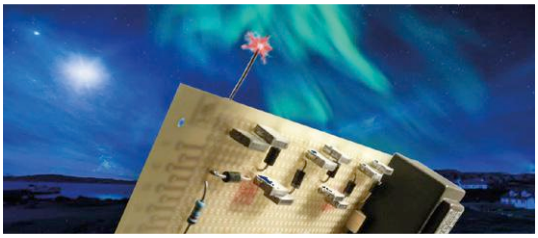
[1] www.jyetech.com

# Elektor Labs Pipeline

Do we ever get enough from projects showing how to measure data and generally control stuff? We don't, so here are a few of them for you to enjoy.

### Feeling depressed? Build an air ionizer!

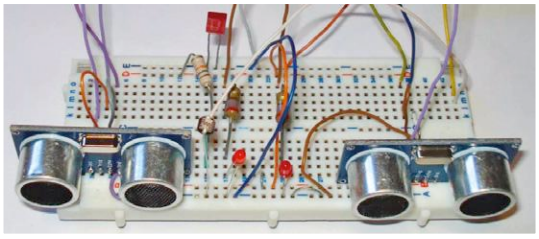
The number of negative oxygen ions in ambient air is reported to affect the psychological and physical state of many people. Both mountain and seaside air contain relatively high numbers of negative ions compared to elsewhere and that's one of the reasons you "feel better" in this type of environment. You can improve the air quality at home too by using the ionizer described in this project.



https://goo.gl/ECws3u

### Build an ultrasound car parking aid

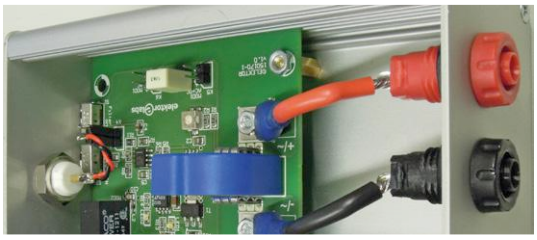
To avoid wasting precious real estate parking spaces in buildings are as minute as possible. Furthermore, such parking spaces are often surrounded by concrete pillars or delimited by walls, making it difficult to move a car into such a tiny reserved space without damaging its precious metallic paintwork. Worse, cars only get bigger and suddenly you understand the need for an electronic parking aid. Here is a simple solution based on cheap ultrasonic transceiver modules.



https://goo.gl/exdN5b

### Potential-free current measurement probe for oscilloscopes

Measuring current with an oscilloscope is difficult because the common of the oscilloscope's input is internally connected to the protective earth of the mains voltage. Current clamps can measure the current without potential, but are slow and usually do not have an output to connect to an oscilloscope. Current clamps for oscilloscopes exist, of course, but are expensive. The circuit presented here shows that a potential-free current measurement can be realized with relatively few but high-quality components.

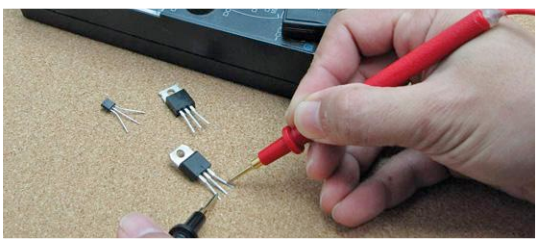


https://goo.gl/hphAFp

### Measure the gain of a transistor with a microcontroller

Of course you can use your super-duper multimeter to measure the gain ($h_{fe}$) of a transistor, but isn't it much more fun to program a microcontroller to do that? Or an Arduino? In the process you might actually learn something from it. Here is a simple project that lets you do just that. Although the software is written in BASIC and intended for an ATmega8 or -88, it should also work on the 328 of an Arduino. Read the source code to figure out how to wire everything together. ⏮

(160530)



https://goo.gl/zKMzdR

eLABS PROJECT

# Spiral MicroTesla

## sparks from a USB charger

By **Daniël Eindhoven** (The Netherlands)



### Properties

- High voltage Tesla coil
- Uses printed PCB coils
- Powered by USB charger or adapter
- Safe sparks
- Complete DIY kit available



## PROJECT INFORMATION

High voltage
USB Tesla
HF

entry level
➡ intermediate level
expert level

4 hours approx.

SMD soldering equipment

£50 / € 55 / $66 approx.
(parts only, without the PCB)

**It seems to be an unwritten rule in Hollywood: there must be an assortment of devices that crackle and spark in the lab of a mad scientist. It doesn't matter that all those noises and flashes have no apparent purpose, as long as the effects look spectacular!**

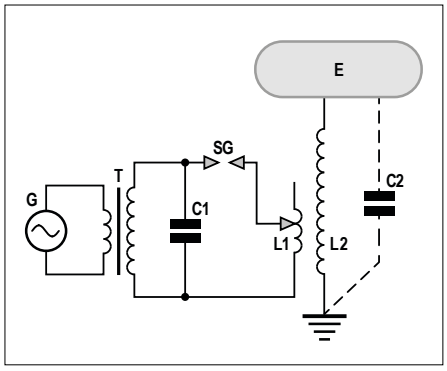



Figure 1. Basic circuit of the original Tesla coil.

To generate these spectacular sparks they used devices known as Tesla coils — named after the physicist and engineer Nikola Tesla (see inset). Most of the Tesla coils have in common that they're quite large and potentially dangerous. There is a lot of energy involved, which makes the chances of a serious shock and/or burns very possible. When we want to make a (relatively) safe Tesla coil we must therefore limit the amount of energy used. In contrast to what the free-energy advocates want us to believe, it is never possible to extract more energy from a system than what was first put into it.

**Power supply via USB**

The main advantage of the Tesla coil described in this article is that it is powered by a USB charger. This means that the supply voltage is only 5 volts. The maximum current consumption is in the region of 1 ampère. The maximum power consumed is therefore:

$P = U \times I = 5\ V \times 1\ A = 5\ W$

which can't cause much damage. You can touch the sparks produced by our Tesla coil with your bare hands, despite the fact that the voltage has been increased to a staggering 30 kV. You would expect to feel a heavy shock, but because the current is so weak you will only notice a slight tingling at most. Nevertheless, we want to emphasize that Tesla coils aren't kids' toys! You can find many projects for Tesla coils on the Internet that are powered directly from the AC line. Please don't even think about building them: they can be absolutely deadly!

**Theory behind the Tesla coil**

The special transformer at the heart of the Tesla coil (L1 and L2 in the basic circuit of **Figure 1**) is called a resonant or oscillation transformer, which clearly operates in a different way to the normal transformers found in AC power supplies. A normal transformer has been designed to transfer energy from the primary winding to the secondary winding as efficiently as possible; an oscillation transformer also (temporarily) stores electrical energy. There is a capacitor in parallel with each winding, which effectively results in two tuned circuits. L1 consists of one or more turns of thick copper wire and it can be tuned; this coil is connected to capacitor C1 via a spark gap (SG). The secondary coil consists of a (very) large number of turns of thin copper wire, within the primary coil. Both coils are air-cored. There is no 'real' capacitor in parallel with the secondary, although there is a capacitance: capacitor C2, shown in dashes in the basic circuit, is the sum of the parasitic capacitances between the individual windings in the secondary and the capacitance between the end electrode (E) and Earth. Both circuits (L1-C1 and L2-C2) have been tuned to the same frequency.

The operation of the Tesla coil (shown in Figure 1 in its original configuration) can be explained (very briefly) as follows:

1. Capacitor C1 is charged up to a high voltage by the supply transformer (T).
2. When the voltage across the capacitor exceeds the breakdown voltage of the spark gap, a spark forms and the resistance across the SG is reduced to a very low value. This closes the primary circuit and a current flows from the capacitor through coil L1. This current oscillates backwards and forwards through L1 at the resonant frequency of the circuit.
3. This causes an oscillating current to be induced in the secondary winding (L2). The energy from the primary circuit is transferred to the secondary circuit during the course of several cycles. The total amount of energy (minus any losses) stays the same as the amount of energy initially stored in C1, so that the oscillating voltage in the primary circuit decreases, while the voltage in the secondary increases. The voltage at the end electrode of the secondary coil will be many times higher than the primary voltage. The very strong electric field at the end electrode ionizes the surrounding air, causing spectacular discharges.
4. The secondary current then generates a magnetic field that induces a voltage in the primary winding; the energy is transferred back to the primary again over a number of cycles. In this way the energy moves backwards and forwards at a high rate between the tuned circuits. The oscillating currents in the primary and secondary windings will gradually die out due to the losses that are always present.
5. When the current through the spark gap becomes too small to keep the air between its electrodes ionized, the spark disappears and the current flow in the primary circuit stops.
6. Capacitor C1 is then charged once more via transformer T and the whole process starts again.

The whole cycle takes place in a very short space of time — the oscillations die out within about a millisecond. Each spark across the spark gap produces a diminishing pulse train of a high-voltage sine wave at the end electrode of coil L2. This pulse train dies out well before the next one starts. The Tesla coil therefore produces a series of damped sine waves, rather than a continuous high-voltage sinewave.
Note that the end electrode of the secondary coil can either be a disc or torus, or have a needle-like shape. In the first two cases the Tesla coil looks a bit like a Van de Graaff generator, although the inner workings are completely different!
Regarding the spark gap: semiconductors were completely unknown when Nikola Tesla invented his coil, and mechanical switches were much too slow to be used in this circuit. The use of a spark gap was therefore a clever way to switch the current on and off very quickly.

**Two types**

There are many designs for the Tesla coil, although we can, broadly speaking, divide them into two groups. The first group consists of Tesla coils that follow the original design, in other words, with a spark gap. The acro-

nym SGTC (*Spark Gap Tesla Coil*) is used for these.
Fast, modern power semiconductors (thyristors, transistors, MOSFETs etc.) are an ideal replacement for the spark gap, without its disadvantages (noisy, gets warm, bad efficiency). These variants are known as SSTC (*Solid State Tesla Coil*).
It won't come as a surprise to the Elektor Magazine reader that we chose the semiconductor version for this project, although several colleagues in the lab have a distinct preference for 'noisy' circuits...

**The circuit diagram**

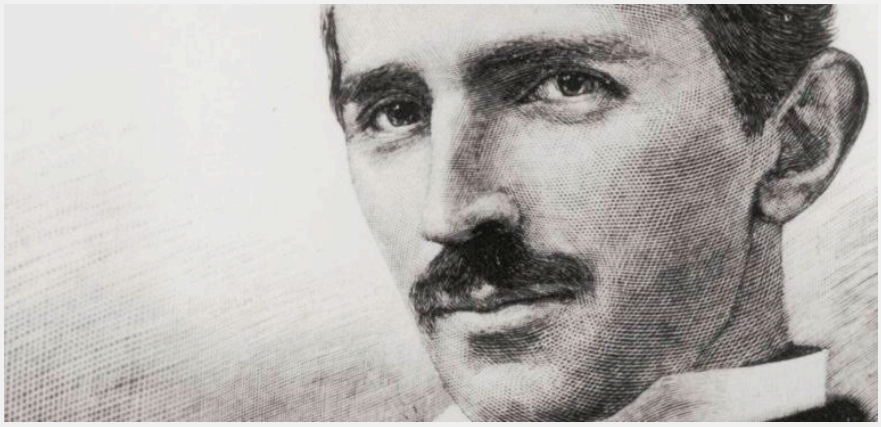
We have shown the circuit for the MicroTesla in **Figure 2**. This is, to be exact, a DRSSTC — *Dual Resonant Solid State Tesla Coil*. It's quite a mouthful, just to indicate that both the primary and secondary circuits use LC series-resonance in order to boost the voltage as much as possible.
The most important components of every Tesla coil are — you've guessed it — the primary and secondary windings. It's well known that electronics hobbyists have an intense dislike of coils if they cannot be bought ready-made. If you give the same instructions for making a coil to ten electronics hobbyists you'll probably end up with ten different coils... To avoid these kinds of problems, we implemented the primary and secondary coils as planar PCB coils. In other words, they've both been made using tracks on the PCB.
The primary coil consists of a single turn on the underside of the board; the secondary consists of 160 turns (in a spiral) on the topside of the board. This secondary coil has a total length of about 25 meters (75 feet).
As we mentioned earlier, the primary coil is not driven by a spark gap. Instead, four fast bipolar power transistors configured as an H-bridge (T4–T7 in the circuit) have been used. These in turn are driven by a couple of high power MOSFET drivers, IC2 and IC3.

**Feedback loop**

We will first concentrate on the driver circuit for the primary coil, since this is at the heart of the circuit. For optimum results, the resonance frequencies of the primary and secondary windings have to be identical (in our case that frequency was in the region of 4 MHz).

## Nikola Tesla — a tragic genius

Nikola Tesla was born on 10 July 1856 in Smiljan (which is now in Croatia). He studied physics, engineering and philosophy at the universities in Graz and Prague. After having worked for some time in France and Germany, he emigrated to the United States in 1884, where he was naturalized as an American citizen in 1891.



In the United States, Tesla initially worked for Thomas Edison, although they fell out after a short period of time. Tesla was then supported by Edison's rival, George Westinghouse, and started his own company, where he either invented or improved upon the most important parts of our contemporary electricity network. Tesla obtained many patents for his inventions, but he wasn't a businessman, in contrast to Edison. He handed over lucrative patent rights to help Westinghouse, who became very wealthy (the Westinghouse Electric Company still exists, although it's now part of Toshiba). This ensured that they broke the monopoly of Edison's General Electric.
Around 1900 Tesla started experimenting with high-frequency alternating currents (one of the results was the Tesla coil) and wireless transfer of electrical energy. However, this turned out to become a (financial) fiasco.
The general public mainly remembers Tesla as an eccentric scientist and inventor. In the latter years of his life (which were mostly spent in New York) he lost contact with reality and he started looking at a 'death ray', anti-gravity, alien life forms and free energy from the 'aether'.
Tesla died in a hotel room in New York on 7 January 1943. However, he wasn't forgotten by his colleagues — they attended his funeral in great numbers.
The SI unit for magnetic flux density (the tesla, *T*) was named after him in his honor in 1960.

[Source: *Wikipedia*]

The H-bridge has to driven at exactly this frequency, but this creates a problem. If we were to use an oscillator with a fixed frequency, it would be very difficult to get its frequency just right. To start with, the inescapable component tolerances would immediately throw a spanner in the works.
It is much simpler (and better) to make use of feedback. When we consider that the current and voltage are in phase at resonance, the solution is easy to see: we're going to use the secondary voltage to switch the primary current.
Because only a relatively low current flows in the secondary circuit, we can use the voltage across a resistor (R28) as the switching signal, without the need to use a current transformer. Inverter IC4 inverts this voltage and turns it into a perfect square-wave (and hence a good switching signal). To prevent the voltage at the input to the inverter from rising too high, we've clamped the coil voltage with a two-stage clamp (D3 and D4).
The output signal of inverter IC4 (*PULSE* in the circuit diagram) is used to switch the H-bridge at exactly the

right time to ensure that the current keeps increasing in strength. This is known as a self-resonating circuit; a simulation of this process can be seen in **Figure 3**.

There is also another reason for choosing to use feedback: the secondary resonance frequency is not constant. Since the plasma channel where the spark appears has a certain capacitance, and because the circuit isn't really ideal due to the use of planar coils, the resonance frequency changes considerably during normal use. Feedback from the secondary circuit to the primary circuit therefore produces the best possible result.

For the same reason we've made the crossover point of the primary circuit much lower than the resonance frequency. It therefore takes several cycles before the resonance gets going.

## Speed considerations

In order to achieve the best possible performance, the delay between the measurement of the secondary voltage and the switching of the primary current has to be as small as possible. This means that we need to use very fast components. For this reason we've chosen fast FZTx51-type transistors (T4–T7) for the switches (actually just emitter followers), and UCC2753x high-power MOSFET drivers (IC2, IC3) to control them.

## Power supply considerations

The maximum supply voltage for the MOSFET drivers in the H-bridge (IC2, IC3) is 35 V; when we add a safety

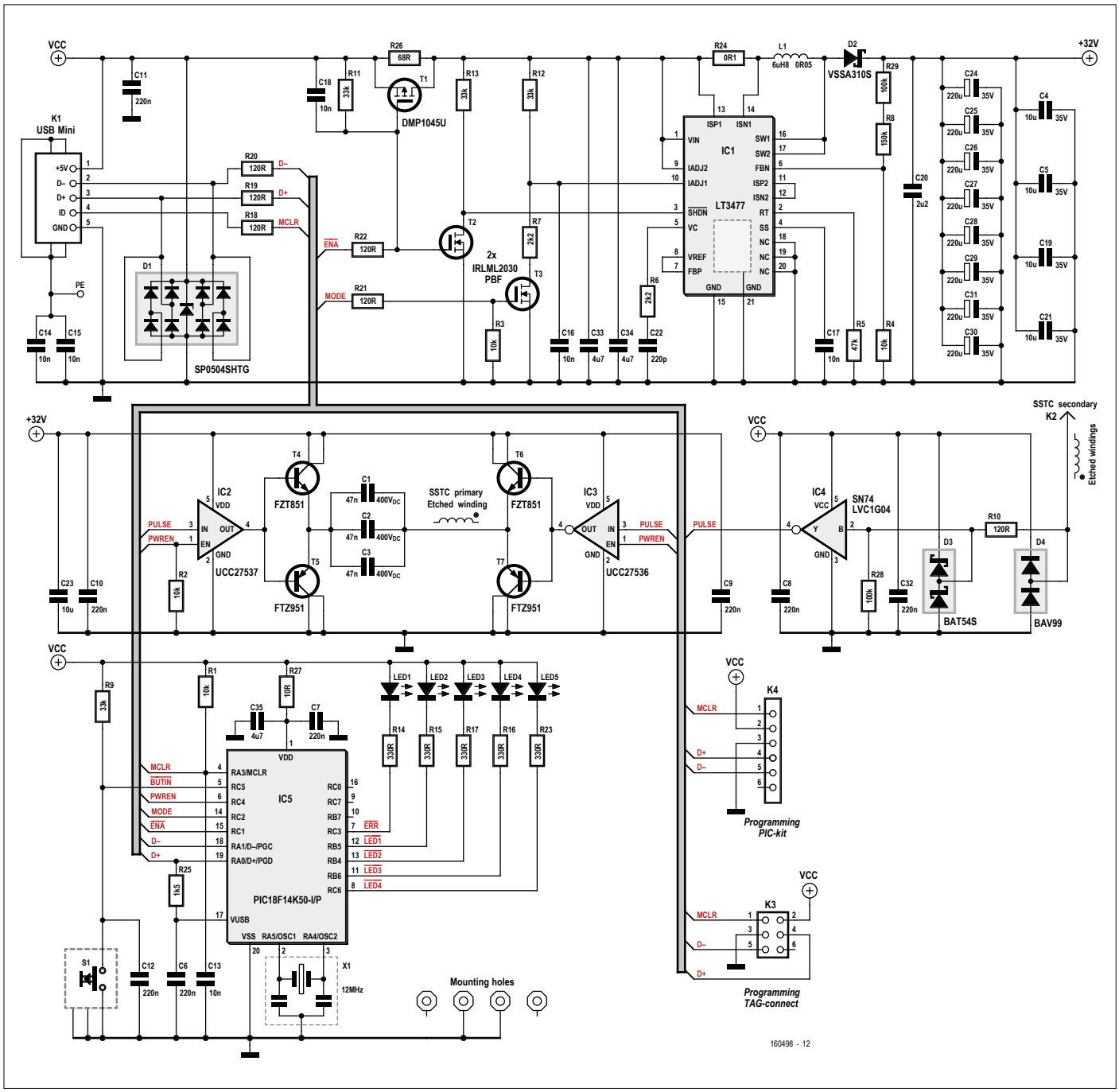



Figure 2. The complete circuit diagram for the Spiral MicroTesla.

margin of 3 V, the resulting supply voltage for the Tesla section becomes 32 V. This has to be derived from the 5 V USB voltage in some way. The first circuit that comes to mind is a DC/DC step-up converter, a circuit that has become a standard building block for the electronics hobbyist.

We decided to use an LT3477 made by Linear Technology, which has rail-to-rail current sense. This feature is perfect for the prevention of overloading the USB adapter or USB charger. The current limit is configured via the voltages on the Iadj pins. When T3 conducts, the input current is limited to a maximum of about 0.5 A (a voltage drop of 50 mV across R24). When the voltage at $I_{adj1}$ is higher than 0.625 V (T3 blocks), the current is limited to a maximum of about 1 A (100 mV across R24). The *MODE*-signal is used to control T3.

The output voltage of the converter is heavily buffered using eight large electrolytic capacitors (C24–C31), with another five ceramic capacitors of 10 µF in 1210 packaging (C4, C5, C19, C21 and C23) connected in parallel. These capacitors ensure that we can obtain the large peak currents of 20 A (!) at the high switching frequencies required by this circuit. A precharge resistor of 68 Ω (R26) is used to keep the initial charging current at switch-on within a reasonable level (remember that we're dealing with a total capacitance of over 1800 µF). After one second this resistor is shorted by T1, which is driven by the */ENA* signal.

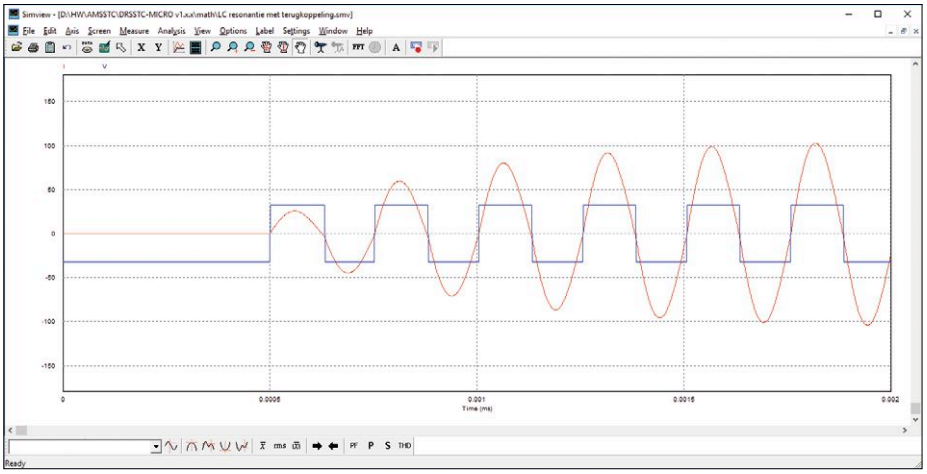


Figure 3. Simulation of the secondary voltage with feedback.

## Lightning in the living room

**Duty cycle**

It is quite a challenge for a designer to create a good Tesla coil. On the one hand, we want to have the largest and most spectacular sparks possible, on the other hand, the power consumption must remain fairly small. One good way to implement these seemingly contrasting requirements is to turn on the circuit for only very short periods of time — the power consumption and the duty cycle are directly proportional.

With a 5 V/1 A power supply we're scraping the bottom of the barrel with a duty cycle of 1.5%. If we choose to have the Tesla coil on continuously, the circuit would consume over 300 watts (!), which it wouldn't be able to cope with: the heat generated would be far too much.

The duty cycle of our Tesla coil can be varied via the software: stronger pulses less often (<10 Hz), or weaker pulses more often (>20 Hz). It is even possible to play some scales by varying the duty cycle (see later).

Virtually all of the power consumed is converted into heat, which has to be dissipated. The power stage uses up most of the power, which shouldn't come as a surprise. The printed circuit board needs to be designed so that all this heat can be dissipated and the components won't overheat. We've added a large copper area underneath the power components to aid with cooling. This is connected using thermal vias to the underside of the board, where more cooling copper has been added.

The infrared photo in **Figure 4** shows the MicroTesla after it has been in operation for two hours. A tempera-

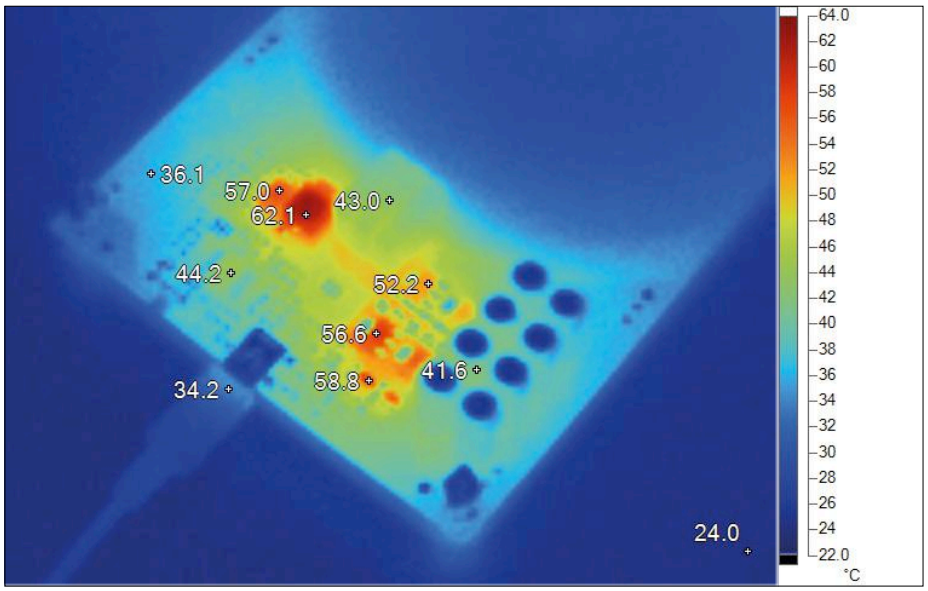



Figure 4. The temperature of the board increases during use, but stays within acceptable limits.

## COMPONENT LIST

**Resistors**

Default: 1%, 0.125W, SMD 0805
R1,R2,R3,R4 = 10kΩ
R5 = 47kΩ
R6,R7 = 2.2kΩ
R8 = 150kΩ
R9, R11,R12,R13 = 33kΩ
R10,R18,R19,R20,R21,R22 = 120Ω
R14,R15,R16,R17,R23 = 330Ω
R24 = 0.1Ω, 250mW, 1%, SMD 1206
R25 = 1.5kΩ
R26 = 68Ω, 250mW, 5%, SMD 1206
R27 = 10Ω
R28,R29 = 100kΩ

**Capacitors**

C1,C2,C3 = 47nF, 400VDC, 5%, PP, 15mm pitch, 5mm width
C4,C5,C19,C21,C23 = 10µF, 35V, 10%, SMD 1210, X7R
C6,C7,C8,C9,C10,C11,C12,C32 = 220nF, 50V, 20%, SMD 0805, Y5V
C13,C14,C15,C16,C17,C18 = 10nF, 50V,10 %, SMD 0805, X7R
C20 = 2.2µF, 50V, 10%, SMD 0805, X5R
C22 = 220pF, 50V, 5%, SMD 0805, C0G/NP0
C24,C25,C26,C27,C28,C29,C30,C31 = 220µF, 35V, 20%, 8mm diam., 3.5mm pitch, EEUFC1V221L (Panasonic)
C33,C34,C35 = 4.7µF, 16V, 10%, SMD 0805, X7R

**Inductor**

L1 = 6.8µH, 3.04A, 0.0498Ω, SMD

**Semiconductors**

D1 = SP0504SHTG, SMD SOT-23-6
D2 = VSSA310S-M3/61T, 100V/3A, SMD SMA
D3 = BAT54S, SMD SOT-23
D4 = BAV99, SMD SOT-23
LED1 = red, low power, SMD, KPTL-3216EC (Kingbright)
LED2,LED3,LED4,LED5 = blue, low power, SMD, KPTL-3216QBC-D (Kingbright)
T1 = DMP1045U, SMD SOT-23
T2,T3 = IRLML2030TRPBF, SMD SOT-23
T4,T6 = FZT851, SMD SOT-223
T5,T7 = FZT951, SMD SOT-223
IC1 = LT3477EFE#PBF, SMD, TSSOP-20
IC2 = UCC27537DBVT, SMD SOT-23-5
IC3 = UCC27536DBVT, SMD SOT-23-5
IC4 = SN74LVC1G04DBVR, SMD SOT-23-5
IC5 = PIC18F14K50-I/SS, SMD SSOP-20

**Miscellaneous**

X1 = 12.0MHz resonator, 0.5%, 5pF with internal capacitors, AWSCR-12.00CV-T (Abracon)
K1 = USB-B mini-connector, horizontal PCB-mounting, SMD
K2 = turned-pin IC socket (only one contact

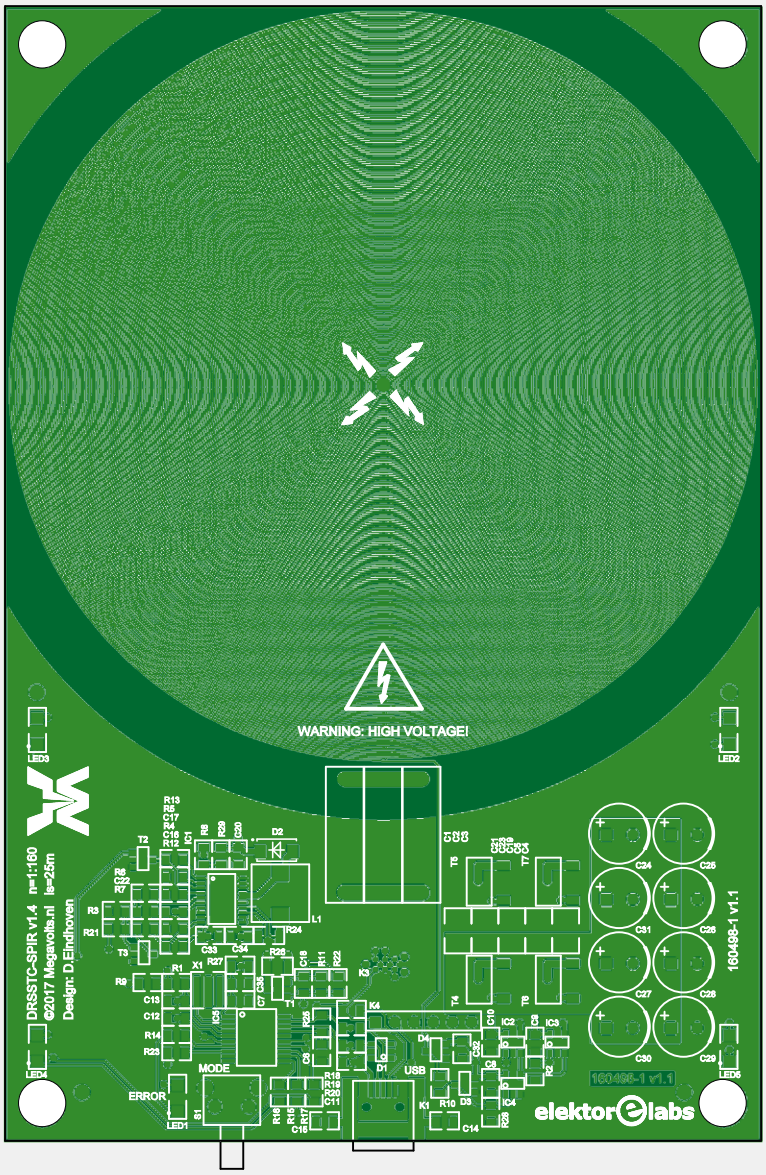


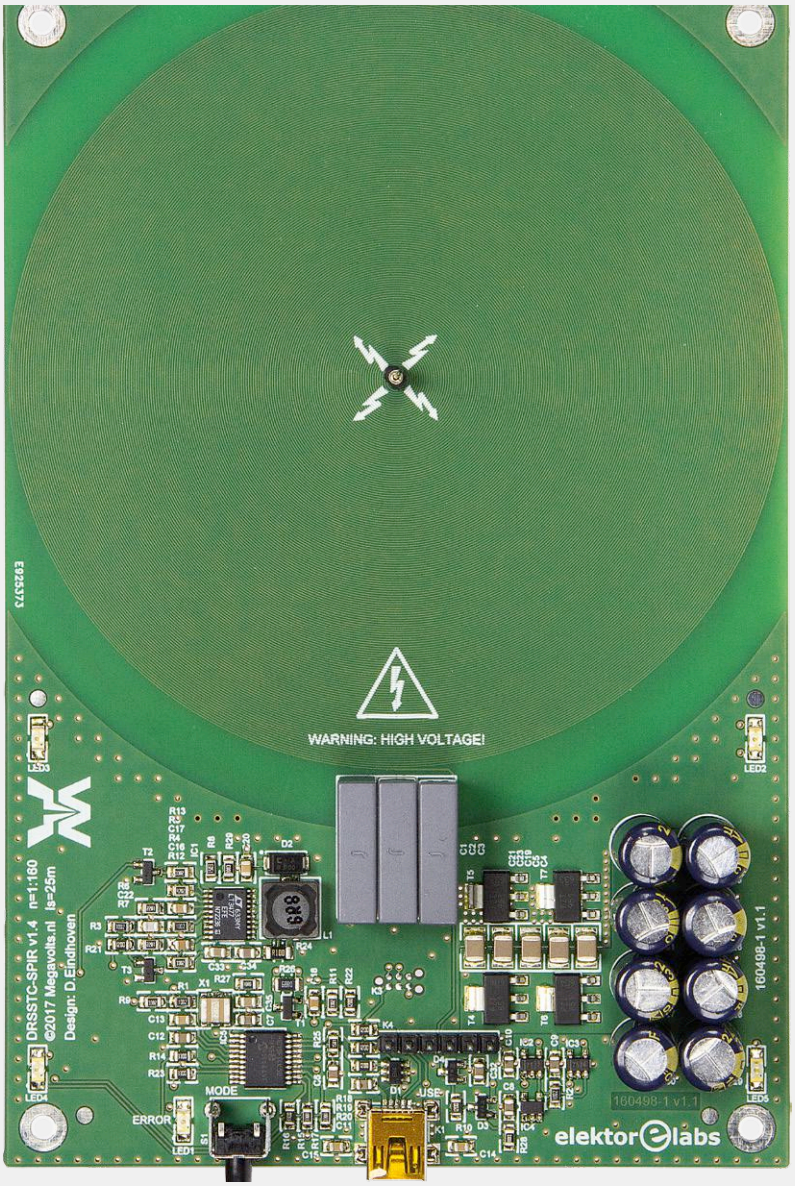

Figure 5. PCB and component layout.

ture of 62.1 °C (which is 38 degrees above the ambient temperature of 24 °C) is perfectly acceptable for power electronics. The board settled at this temperature after about half an hour.

### Firmware

A small PIC microcontroller (PIC18F14K50, IC5) takes care of all the control functions. It controls the precharge transistor, the setting for the current (500 mA or 1 A), and the duty cycle. We decided not to include USB communications in this version of the MicroTesla. The microcontroller also drives an Error LED (LED1). Since it only requires a bit of software and hardly any extra hardware, we've added four blue LEDs (LED2–LED5) to enhance the board's visual aspect. Switch S1 is used to select the required operating mode (see next paragraph).

The firmware can be downloaded from the project page [1], and programmed into the microcontroller via connector K4.




required)
K4 = 6-pin SIL pinheader, 0.1" pitch
S1 = pushbutton, horizontal, SKHHLQA010 (Alps)

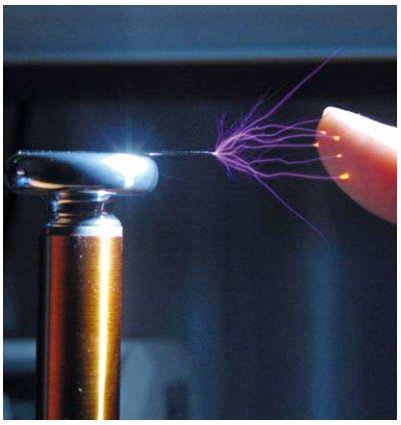
**Mechanical components**

8 spacers 15 mm
4 bolts M3 x 40 mm plus nuts
Acrylic sheet top, 100x150x2 mm
Acrylic sheet bottom, 100x150x2 mm

PCB 160498-1 v1.4

**Sneak preview**

The circuit for the Spiral MicroTesla can also be used as the basis for a 'traditional' Tesla coil with a vertical air-cored coil and a toroidal terminal — the ultimate high-voltage gadget. We hope to make this version available as a kit of parts in the near future.



A kit of parts for a 'real' Tesla coil will be made available in the near future.

**Construction and operation**

A double-sided PCB has been designed for our Tesla coil (or to give it its official name: Spiral MicroTesla), which is shown in **Figure 5**. You can try to etch this yourself, but we wouldn't recommend it — the smallest short between any of the windings in the secondary coil will make the circuit inoperative!

You can buy a ready-made board from the Elektor Store, which is guaranteed to work.

Most of the components are of the SMD variety, although this shouldn't be an insurmountable problem for an experienced hobbyist with a steady hand (and the right soldering equipment, of course).

There is good news for those of you who don't like to work with SMDs: a complete kit for the Spiral MicroTesla is also available, with a board that has all of the SMDs already mounted (you still have to solder a few through-hole components). You can't really go wrong then.

The end electrode is mounted using

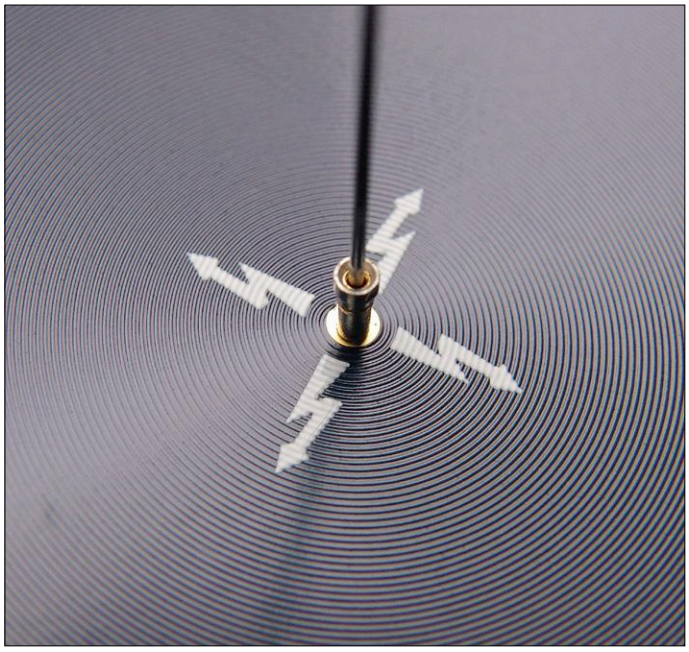


Figure 6. The end electrode is mounted in a sprung socket from an IC socket.

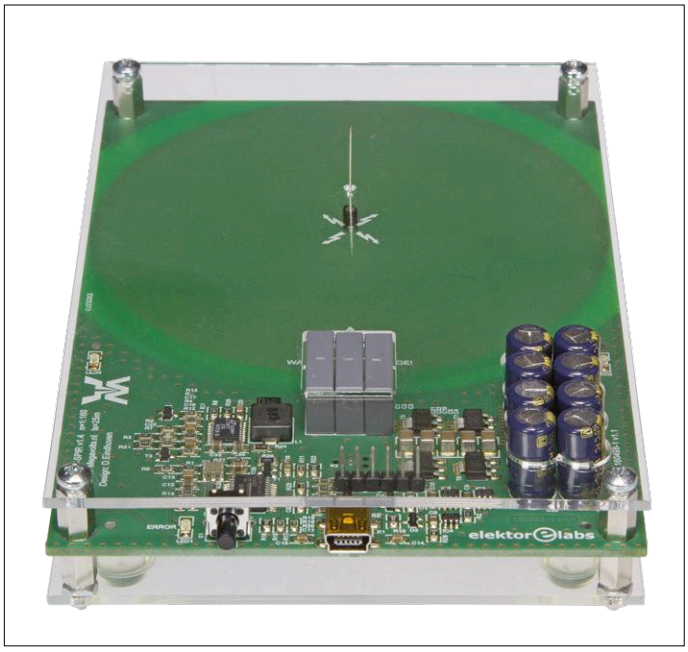



Figure 7. The MicroTesla board is mounted between two acrylic sheets (Perspex®) to prevent any parts from being touched.

**About the author**

Daniël Eindhoven (30) became fascinated by Tesla coils after playing the game Command & Conquer: Red Alert as a child. When he found out that you could make them yourself, there was no stopping him: Daniël has since built a large number of working devices of various designs and implementations. In 2010 he completed his studies in Electronic Engineering at the Haagse Hogeschool in Delft, The Netherlands. Since then he's been working as a hardware design engineer in power electronics. Daniël is interested in just about anything to do with technology. His website at www.megavolts.nl/en is certainly worth a visit.



a single sprung socket that has been carefully removed from an IC socket; the electrode is simply pushed into it (see **Figure 6**). Note that the board is mounted on spacers between two acrylic sheets to prevent any of the components from being touched. The end electrode sticks through a hole in the top acrylic sheet (as well as the large capacitors). **Figure 7** should give you a good idea what we have in mind. Note that these acrylic sheets are also included with the complete kit.
The Tesla coil is operated using a single push button (S1), which works as follows:

- A short press of S1 (less than 1 s) switches between the various operating modes:
  Turn on, 5-Hz pulses → 10-Hz pulses → 20-Hz pulses → musical scale → turn off.
- Press S1 for 1 s to return to the initial mode (5 Hz).
- Press S1 for 3 s to select the high-power (1 A) mode (red LED LED1 flashes); another 3 s press returns to the normal (0.5 A) mode.
- Press S1 for 8 s to turn off the blue LEDs (so that you can take impressive photos of the sparks in the dark).

**Disclaimer**

The use of the Spiral MicroTesla is completely at your own risk; neither the author nor the Editor nor the Publisher accepts any responsibility for any damage that may be caused by its use. You must connect the Spiral MicroTesla to a reliable USB charger or USB adapter, but you must never connect it to a USB port of a computer! Keep sensitive electronics away from the discharges! ⏮

(160498)

## Web Link

[1] www.elektormagazine.com/160498

FROM THE STORE

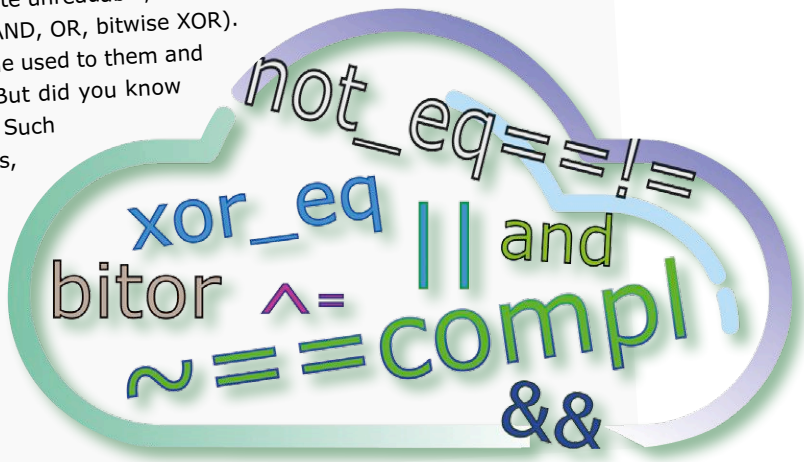
→160498-71
Complete kit (through-hole components need to be soldered) incl. acrylic sheets.

# HomeLab Helicopter

Samengesteld door **Clemens Valens** (Elektor Labs)

## C/C++'s best kept secret

One of the most frequently heard complaints about programming languages is their lack of readability. Programing requires a lot of typing and not surprisingly programmers invent ways to do more with less typing. Resulting in unreadable source code. C and C++ are considered quite unreadable, which is true, just look at the Boolean operators like '&&', '||', '^' (AND, OR, bitwise XOR). After many years of practice programmers have become used to them and they show up in other programming languages too. But did you know that C++ also features so-called 'alternative tokens'? Such tokens provide an alternative spelling for other tokens, e.g. 'and' for '&&', 'bitand' for '&' or 'xor_eq' for '^='. The reason alternative tokens exist is not to improve readability — that they do is pure coincidence — but to support character sets that do not include '{} []#\^|~' (non-ISO 646 characters). If your C-compiler knows about a file 'iso646.h' then it supports these alternative tokens too. More information here: http://en.cppreference.com/w/cpp/language/operator_alternative

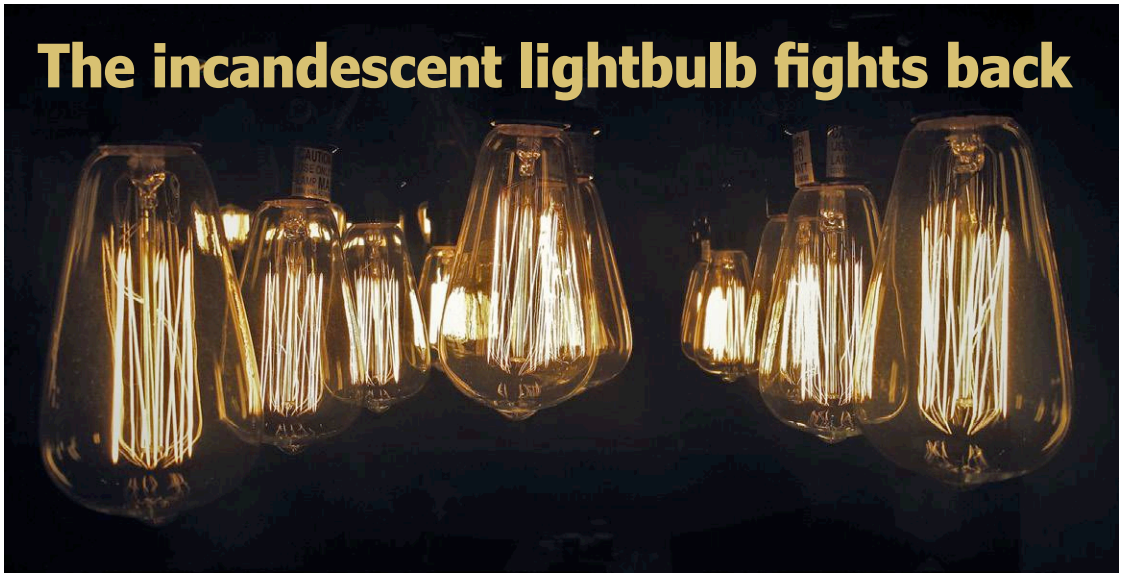



**HiSilicon is China's largest fabless semiconductor and IC design company with a 2016 revenue of €3.3 billion ($3.8 billion).**




**You just can't have too many hot glue guns.**
(Photo by Gregory Ester)

# The incandescent lightbulb fights back



Just a few years ago the authorities decided that incandescent lamps were wasteful, not ecological and not green. And so they were banned, starting with the 100-watt types. Soon the filaments in all other incandescent lamps were replaced by halogen bulbs, then everybody switched to LEDs and incandescent lamps went into oblivion. But not for long.

This summer when browsing a tourist market during my holidays I discovered vintage lighting. A nice shiny wooden box with a large toggle switch and an incandescent lamp mounted on top of it produces a warm glow instead of a lot of light, similar to the valves in modern tube audio amplifiers. On the Internet such designer lightbulbs made of smoky glass with a long meandering filament inside are plentiful. They come in many shapes, and are often supported by fake brass sockets and vintage electrical wires (remember, those fabric- or rayon-covered power cords). The incandescent lightbulb is back, revamped and relooked. Once a life's necessity, today it resurfaces as a decorative object and will probably do so for another century or more.

More-information Internet click bait: antique vintage lightbulb




**They are starting to pop up everywhere, it seems.**
Here is the MakeShop in Dublin, spotted, visited and denounced by Gregory Ester.

## Lissajous in the Sky with Diamonds



Do these two functions mean anything to you?

$x(t) = A \sin(\alpha t+\delta)$

$y(t) = B \sin(\beta t)$

Indeed, they both describe a sine wave. The first has a frequency α, an amplitude $A$ and a phase δ; the second has a frequency β, an amplitude $B$, and zero phase angle. With two function generators these signals can be created easily. When connected to the inputs of an oscilloscope set to XY-mode, a two-dimensional curve will be displayed. If $\alpha = \beta$, $A = B$ and $\delta = \pi/2$ the curve will be a circle; if both $A$ and $B$ are zero you should see a dot. Playing with the ratios $\alpha/\beta$ and $A/B$, and the value of δ will produce all sorts of complex curves commonly known as Lissajous curves (sometimes also referred to as Bowditch curves). But you can do better, much, much better, as is demonstrated by Jerobeam Fenderson who creates Lissajous curves from his music in such a way that they become incredible animations, not just curves (awesome on an old Tek, *Ed.*) Inspired by Jerobeam, Hansi Raber wrote the tool OsciStudio that lets you design your own musical animations. Check it out and send us your videos!

www.elektormagazine.com/news/how-to-draw-mushrooms-on-an-oscilloscope-with-sound

http://oscilloscopemusic.com/index.php

# Life hacks



The Internet is rich in videos showing tips and tricks employing common objects and tools. Some tricks are more creative than others, but they are inspiring without exception. Popular items used in these so-called Life Hacks videos are plastic bottles, clothespins, cigarette lighters and hot glue guns, the best hacks using all of them, proving that laser cutters and 3D printers really are for sissies. Here is a small selection of hot glue gun videos. And if you are not impressed by the hacks, you can always enjoy the music.

https://youtu.be/uoMLCV-qhjQ

https://youtu.be/9W5N8_dYBPI

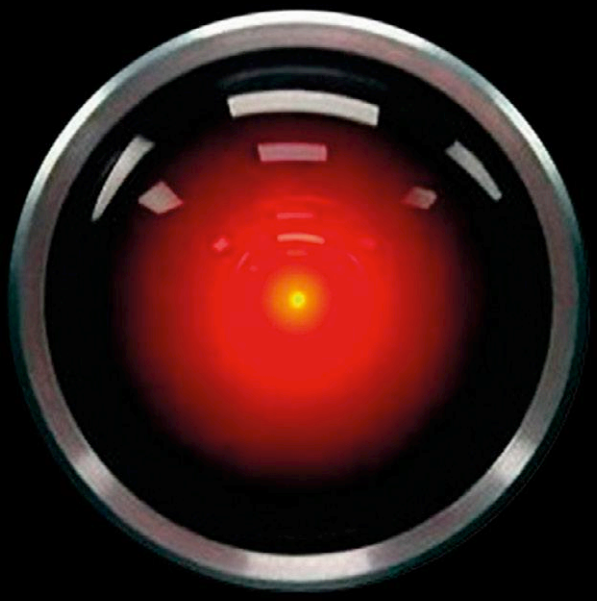
https://youtu.be/MgjotTL5Qp4



# Must-have homelab tool

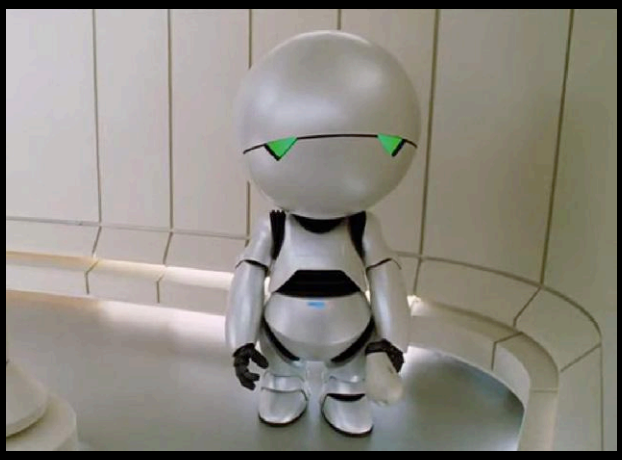
Trending on YouTube, Facebook, Amazon and other gadget websites where it totals hundreds of thousands of views, here is the Angle-izer Template Tool. This clever construction of four sliding rulers is intended for copying and reproducing angles and complicated shapes in difficult tiling, laminating and other cutting situations. Not to be confounded with the death metal band Angelizer, this plastic tool is available almost for free all over the Internet. If you decide to buy one, you'd better buy two, because this is one of those tools that once you have it, you can't live without.

# I am sorry, Dave



AI HAL9000 from 2001: a Space Odyssey

While Artificial Intelligence or AI is slowly and silently invading our daily lives, classic questions like "Will machines become smarter than humans?" or "Will AI destroy the human race?" and, more down-to-earth, "Will AI make me lose my job?" keep more and more of us awake at night. But what about the feelings of an AI? Or as Marvin put it in the Hitchhiker's guide to the galaxy: "I didn't ask to be made; no one consulted me or considered my feelings in the matter." Do AIs have conscience? Do they have moral personhood? Do we have moral obligations to AIs? Are we allowed to turn a possibly sentient computer system off? These and other unsolved ethical AI problems keep philosophers awake, some of them even at night. Klaatu barada nikto!

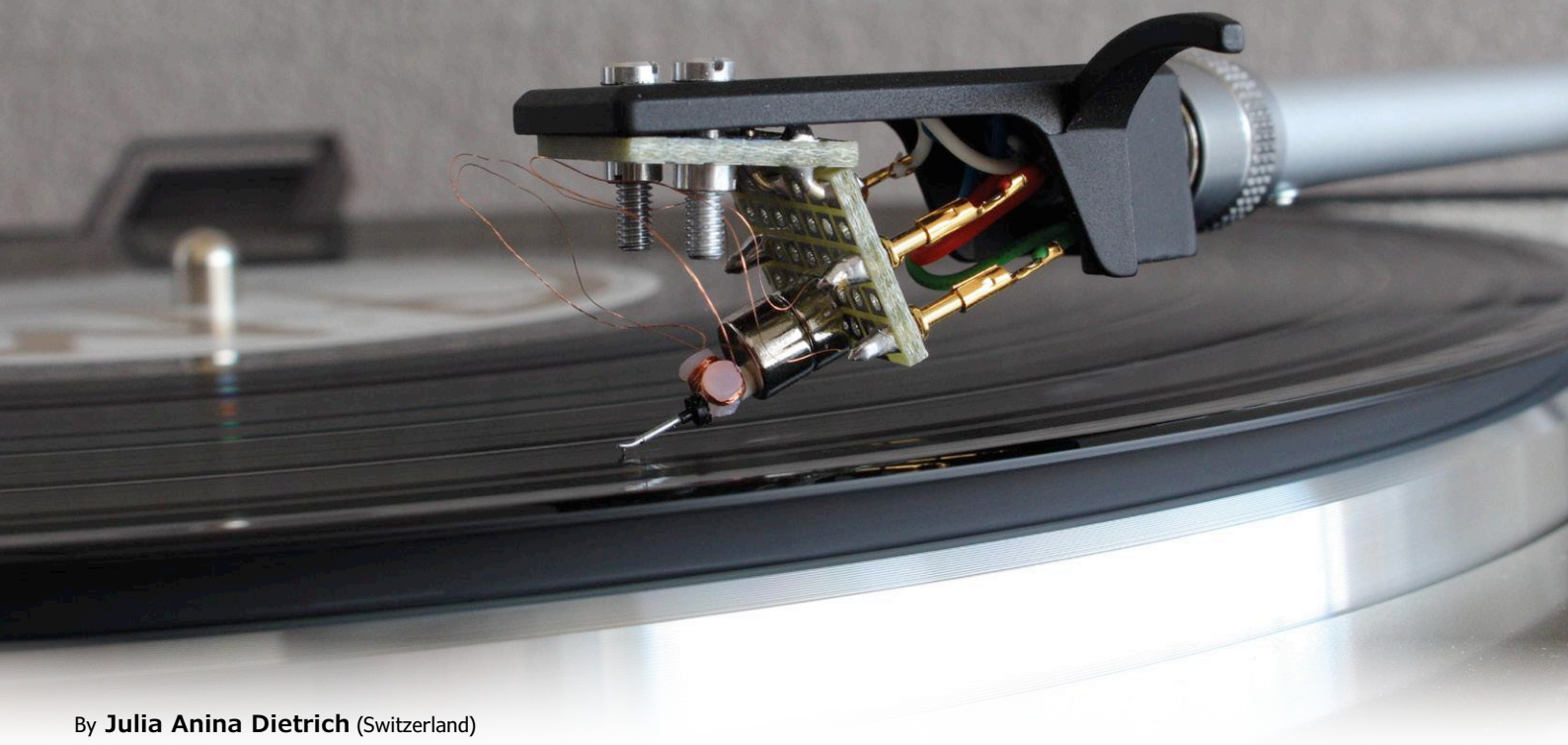


Marvin, the depressed robot
from the Hitchhiker's Guide to the Galaxy.

**Want to participate? Please send your comments, suggestions, tips and tricks to labs@elektor.com**

# DIY MC Element

## Impossible? Of course not!



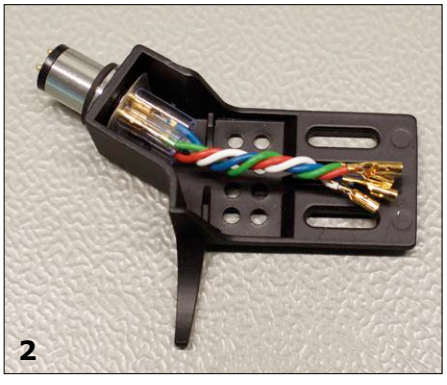
By **Julia Anina Dietrich** (Switzerland)

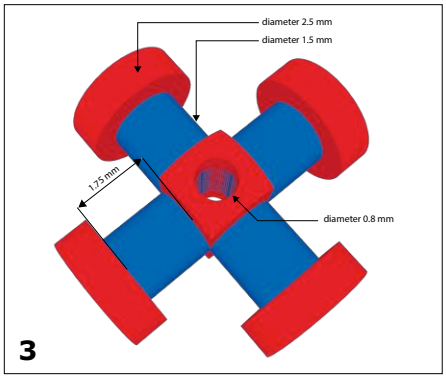
As the topic of her physics final exam essay, student Julia A. Dietrich examined whether it is possible to construct an MC pick-up element yourself. It turns out that with some skill and patience and even without any special tools this is quite possible.

The vinyl disc is entirely back in vogue after having disappeared — come to think of it, actually it was never completely gone! Despite a number of irrefutable advantages of the CD (such as freedom from scratches and noise) many audiophiles swear by the black disc, which is said to sound 'more musical' and 'warmer'. In any case, a record player is relatively low-tech compared to a CD player. So much in fact, that it is possible to fabricate a moving-coil element



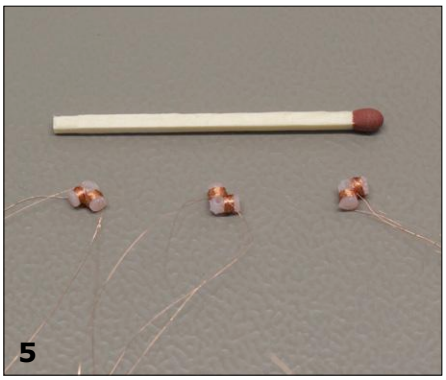
1

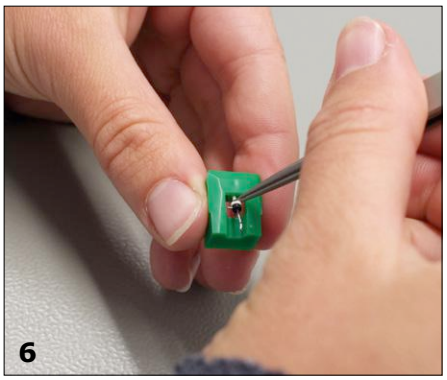

2



3







yourself that — at least to the ear — performs just as well as a bought, ready-made version, as we will demonstrate with this contribution.

**Preparations**

To carry out this project I required remarkably few tools and materials: side-cutters, scissors, fast-curing glue and a few meters of winding wire with a diameter of 0.05 mm (42 AWG) (**Figure 1**). In addition, of course, an empty head shell (element holder) into which the DIY element will be built (**Figure 2**) and a couple of bobbins to wind the coils on. I wanted to build an MC element. With these, the coils are attached to the needle and move in a stationary magnetic field, so that a voltage is induced. That is why I needed a bobbin that was very light yet offered space for two mutually perpendicular coils. The latter is required in order to obtain a stereo signal. Earlier experiments (not described here) indicated that each coil needed to have about 150 turns.

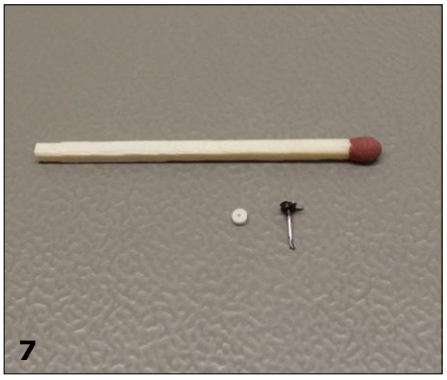
My fellow student Nikola Mastelic made a 3D drawing of the bobbin (**Figure 3**). Subsequently 20 samples of this bobbin were made using a 3D printer. During the first winding attempt it turned out that the four-legged 'cross' was still too large. Because the needle is small and short, the bobbin would touch the record and that is of course not the intention. That is why I cut off two of the four 'legs'.

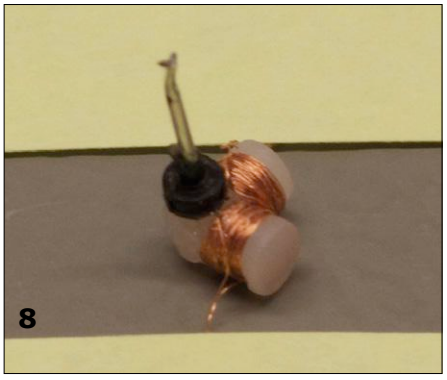
**The winding**

Each coil requires a length of winding wire about 2 m long (6.5 ft.) to hand wind the 150 turns (**Figure 4**). Because the bobbin is very tiny and the winding wire very thin, this is a fiddly job of the first order. Nevertheless, after a bit of practice this was reasonably easy to do. I made four bobbins this way (that is eight coils in total). The first attempt, after a closer inspection, the result was found to be unusable and disappeared into the bin. I continued with the three remaining ones (**Figure 5**). After the coils were wound I fixed the windings in place with a small drop of instant glue.

**The needle**

Now it was the needle's turn. This I (obviously) didn't cut myself, but ordered ready-made — this is not difficult at all and they are not all that expensive either. Ex-factory these needles are mounted in a plastic holder, so that they can be inserted into a conventional MM-element, so small magnets were already attached to the needle. I removed these with tweezers. For all that, at my first attempt the needle jumped from the tweezers. To this day I have not managed to find it again, so it was fortunate that I had bought a few spare ones (**Figure 6**). A small disc of foam allows movement of the needle. Because this disc is required later, I removed it carefully so that it would not be damaged (**Figure 7**).



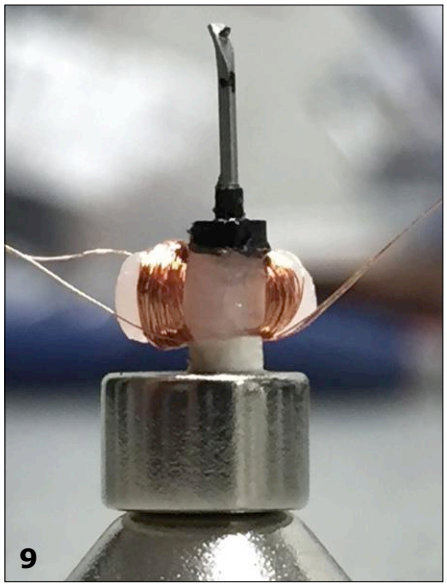


**Mounting**

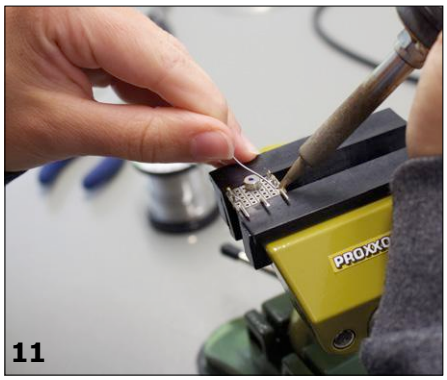
Now the time has come to glue the needle to the bobbin. Since this will become an MC element, the needle needs to be attached to the bobbin. The flexible part of the element will be behind the coils. When gluing, I had to make sure that the needle was at the correct position with respect to the coils, to ensure good channel separation later. I therefore first held the bobbin in place using yellow Post-it notes, and then picked up the needle with tweezers, dipped the back end into a drop of glue and inserted it carefully into the bobbin. This worked really well for all three of the elements that I built (**Figure 8**).

For the magnetic field in which the coils will move, I ordered a number of round magnets of various sizes. The type with a diameter of 6 mm and a thickness of 4 mm turned out to fit best. Between this magnet and the bobbin comes the foam disc from **Figure 7** that acts as the flexible element. This small disc therefore has to be glued directly to the magnet… the next challenge.

On my first attempt I held the disc with tweezers, allowed it to soak up a little bit of glue and then placed it on the magnet. However, the foam absorbs too much glue and after drying is much too hard and therefore no longer usable as a flexible element. Therefore, on my second attempt I used a toothpick to apply a very small amount of glue to the magnet and used that to glue the disc. Now the foam plastic disc remained flexible.


9


10



11

## Low-tech versus high-tech

Compared to a CD player, a record player is, in principle, a low-tech device. To be able to play a CD an entire infrastructure has to be in place, which all has to operate flawlessly. For starters, without power nothing can be done with the silver-colored disc. The laser for scanning, the detector, the microcontroller for decoding the signal and of course the control and positioning of the reading assembly all have to operate faultlessly. If even only a small part doesn't operate correctly then it becomes impossible to play the contents of the CD or even retrieve any of it.

Compare that to the gramophone record. The original vibrations in the air


14a


14b

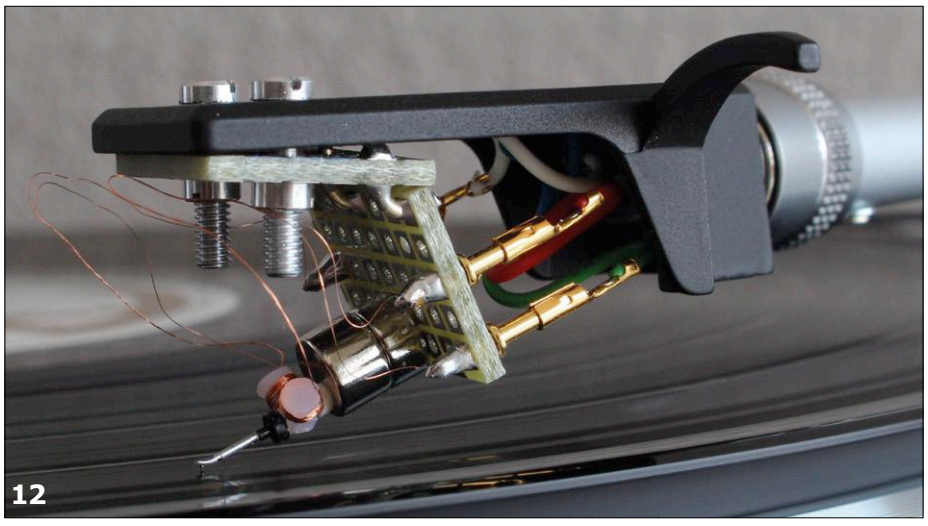
Subsequently I attached the magnet, now with the foam disc, to a conical magnet salvaged from an old loudspeaker. I then glued the bobbin with the needle to the disc using a very small amount of glue. With this I had to make very sure (by eye) that the coils would not be outside the field of the round magnet. The result of all this fiddly work can be seen in **Figure 9**.

### Final mounting

In order to be able to mount the assembly of **Figure 9** in the headshell (and then in such a way that the needle will rest at the correct angle in the groove of the record), for each of my three elements I used two small pieces of prototyping board, which I first cut to size (**Figure 10**).

In one of these pair of boards I drilled holes that allowed the element to be screwed to the headshell. In the other board, four suitable circuit board pins were soldered for the connecting wires. Between these pins I glued a magnet that has the same diameter as the magnet to which the needle is attached. Both magnets are later (carefully) brought together. In this way it remains possible to adjust the angle of needle a little (**Figure 11**).

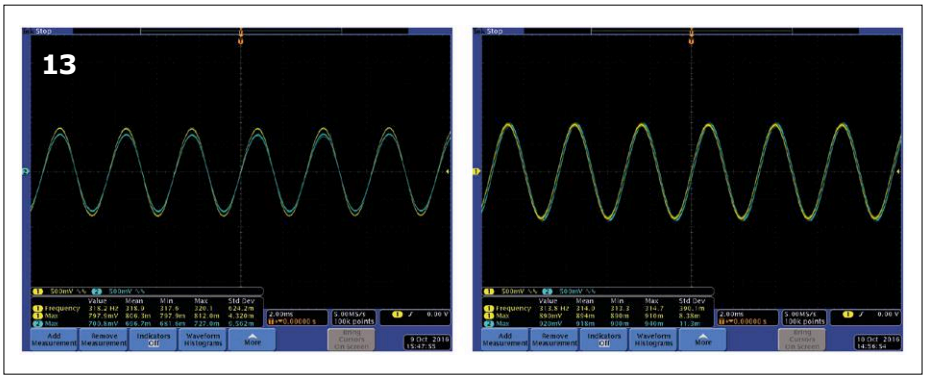
Both circuit boards are joined together using three small pieces of stiff wire. Finally the wires from the coils have to be attached to the circuit board pins and then the DIY element can be mounted in the holder (**Figure 12**).


12

### Assessment

In order to be able to assess the characteristics of the element in practice, I naturally needed a good preamp (since many modern amplifiers don't have a phono input any longer). That is why I decided the build the **Supra 2.0 Preamp** from the Elektor July & August 2016 issue. This is an MM amplifier, i.e. for elements with a moving magnet. Because MC elements generate a much smaller voltage, a small change was required to this preamp. This is described in the

are, without fuss, cut as mechanical vibrations into a carrier; no coding, no complicated signal processing. Well yes, the principle was already patented by Edison in 1877. And because a gramophone record is such a low-tech sound carrier, it can (in a moment of need) be played without any infrastructure (meaning that, in any case, with a bit of effort, the contents can be retrieved). The only things that we need for that is a pencil or something similar that can serve as a spindle, a piece of paper rolled into a horn and a pin to sense the groove (**Figures 14a, 14b**)!
Try it yourself — but don't use an irreplaceable unique vinyl copy for this...



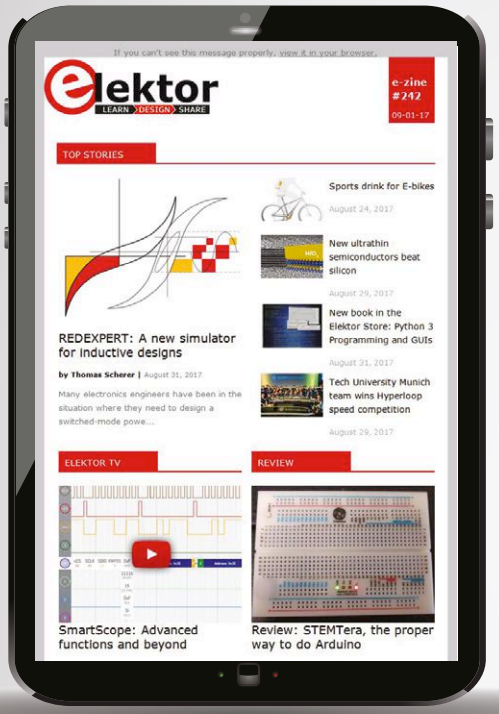

January & February 2017 issue of Elektor Magazine.
With the aid of a special test record, I made several measurements in order to compare my DIY element with a ready-made, commercial version. **Figure 13** shows an example. On the left is the signal from my own element at a needle deflection of 50 µm, on the right the signal from a bought element.

It turns out that with the DIY version both channels do not generate the same amplitude signal, probably because the needle is not resting perfectly square in the groove. However, when judged by ear, no difference is detectable. Certainly not a bad result for an experiment in do-it-yourself!

(160329)

Advertisement

Register now for Elektor's Free Weekly Edition




Elektor's e-zine is jam-packed with interesting news, projects, videos, offers, tips and trends from the world of electronics.

In your inbox each Friday!

FREE PDF inspiration bundle for a limited time only!

Register today at www.elektormagazine.com/ezine

# Tips and Tricks

## From readers for readers

Here is another neat solution sure to make the life of any electronics enthusiast that much easier.

## Repairing damage from leaking alkaline batteries

**By Peter Krueger**

*You know the problem; you go to the drawer to get out a flashlight, radio or electric travel toothbrush you haven't used for a while and the batteries are dead. Worse still, they've started leaking and left a real mess. If you want to restore the battery contacts to (almost) new again, read on. Cursing might help relieve the stress in your head but won't help with the battery situation. Expired alkaline-manganese primary cells are primarily a real annoyance. As they become exhausted the internal pressure rises as a result of electrochemical processes and bursts the 'safety valve' diaphragm. The electrolyte material (a caustic potash) then empties through two degassing holes at the negative pole [1]. Battery manufacturers don't seem able or willing to make their batteries gas tight! If this happens to you and it ruins the equipment, then you will probably agree with me: a product that can potentially cause so much damage really shouldn't be on the market! If the situation is not so bad help is at hand!*

### *Take it from here*

*In* **Figure 1** you can see how my cycle rear-light looked after being abandoned for two years in a drawer. If you are interested in the process, the chemical reaction in the cell producing electrical energy causes the liquid electrolyte (potassium hydroxide) to break down and generate hydrogen gas. The gas pressure eventually ruptures the battery seals and escapes, allowing the electrolyte to leak out. In contact with $CO_2$ in the air it forms potassium carbonate ($K_2CO_3$) which is the white (hygroscopic) crystalline crust that you can see. To be safe you should handle this wearing rubber gloves and try not to breathe in any dust created. This alkaline compound corrodes anything metallic it comes into contact with, usually the contact springs in the battery holder, any wiring and also PCB tracks. To make matters worse, the dry salt deposits make very bad conductors. Just swapping the old batteries for new ones and blowing away the loose salts is not going to do the job. While any damage to wires or tracks can be remedied with a soldering iron and copper wire, unless the gunk is cleaned off, the spring will always be a source of flaky contact and give unreliable device operation. I've seen recommendations to use acetic or some other acid to clean chromium-plated, nickel-plated or gold-plated steel spring contacts but this can give rise to more corrosion. Don't be tempted to try it; there is a better solution…

### *Damage limitation*

*While watching a TV science show about bases and acids, I had this idea to try to dissolve these alkaline deposits from an alkaline*

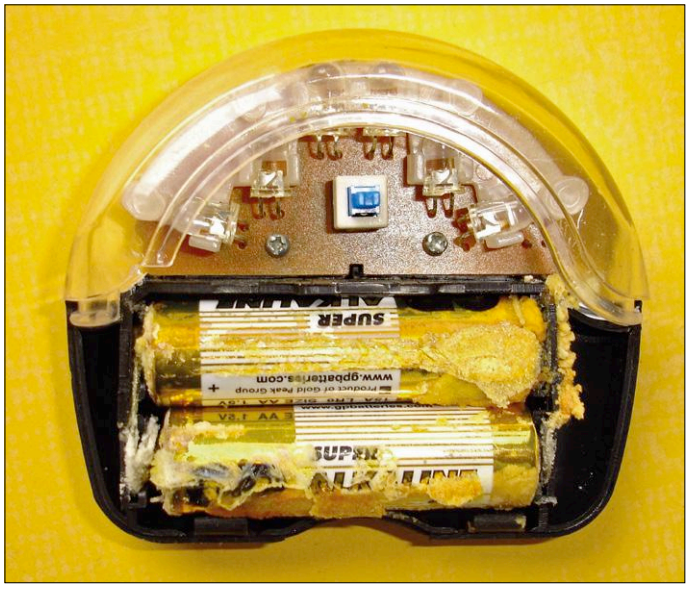



Fig. 1. After two years of neglect the alkaline batteries in this cycle rear light are exhausted and leaking.

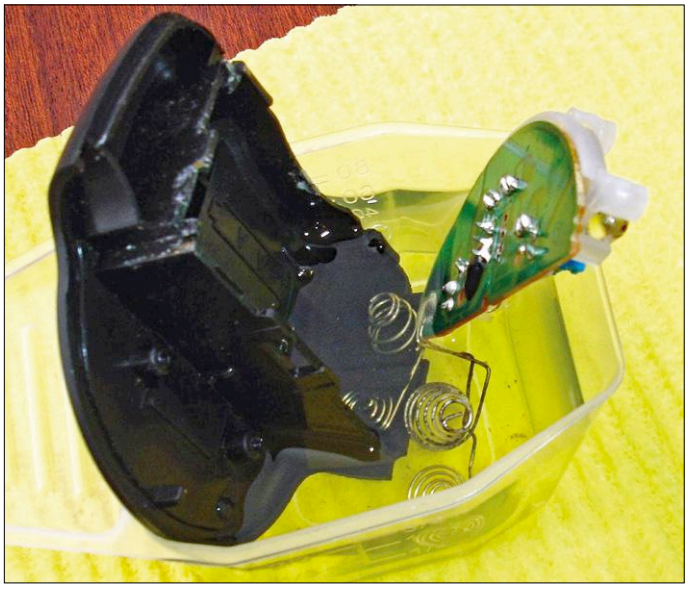


Fig. 2. The battery contacts and affected PCB area are soaked in the bath containing dish washer detergent.

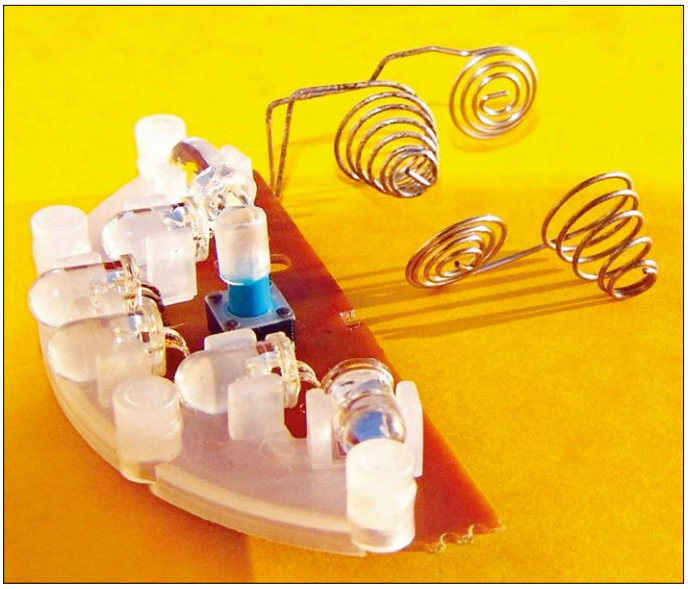


Fig. 3. Sparkling results: The contacts look almost like new!

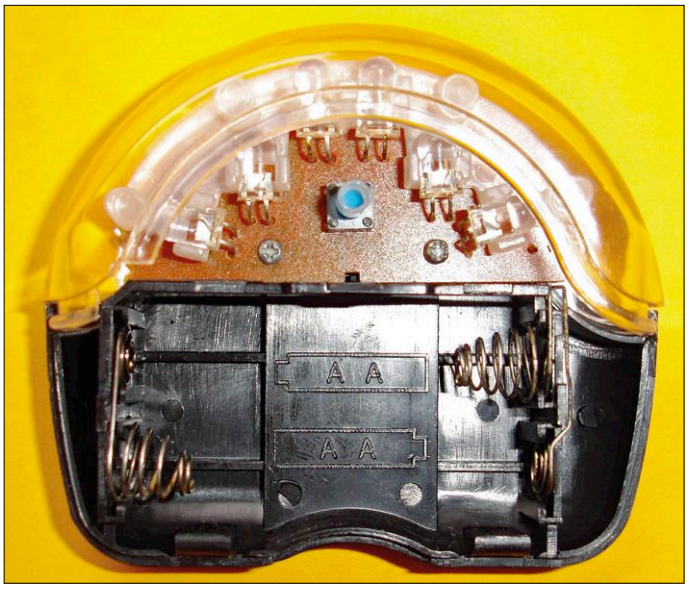



Fig. 4. The rescued rear light just waiting for a new set of batteries. Rechargeables will be a better option.

*battery by using an alkaline solution, such as ordinary dishwashing liquid. I was really amazed after my first test; the recovery of damaged devices is so simple, the idea using dishwashing liquid works a treat. If you don't believe me, just try it for yourself. It's important that the cleaning liquid contents include a few percent anionic surfactants (most do). I got good results using a cheap brand of hand washing liquid soap called 'Aloe Vera' sold at a local store. To avoid any further corrosion damage, the detergent should have a low pH. Check using the simple taste test; it should taste neutral rather than sour. The encrusted battery contacts or complete battery holder (***Figure 2***) can then be placed in a bowl filled with detergent. The solution should be used undiluted, it already contains enough water. After approximately one day soaking in this cleaning solution, the deposits are dissolved and the solution has become thicker during the process. Now rinse with clear water, leave to dry and voilà: almost like new. Check out* **Figure 3** *for proof! Just reassemble and the rear light is ready to go (***Figure** *), all we need now is a new set of batteries.*

***What else?***

*Prevention, they say, is better than cure so I tend to use NiMH rechargeable batteries instead of primary cells. My preference is for the 'ready to use' type which feature low self-discharge (also known as 'Eneloop' from Panasonic). They keep their charge better and can put up with being ignored for a year. Even after this time, it will only be the battery pack that is not working. Standard NiMH rechargeables also succumb to this fate but more quickly and therefore it happens more often. Sometimes the batteries will be totally encrusted and difficult to prize out without causing damage. The solution in this case is to drop the battery holder together with the batteries into a bowl of the cleaning solution. To make things easier it may be possible to desolder the battery-holder leads from the main PCB. Don't worry if this is not possible, I have successfully dunked the whole PCB including batteries and components into the cleaning solution without doing any apparent damage. The PCB is really shiny when it emerges from the process. Make sure that there are no potentiometers, switches or pushbuttons that can come in contact with the cleaning solution; it can be difficult to flush the solution out of all their nooks and crannies so some residue and moisture may remain. Why not try it out? No risk, no fun!* ⏮

(160429)

**Author's bio**

Peter Krueger is an electronics development engineer with 62 years of professional experience in the field of digital and analog electronics including hardware and software development.

[1] https://en.wikipedia.org/wiki/Alkaline_battery

**Have you come up with an inspired way of solving a really challenging problem? Or found an ingenious but 'alternative' way of using some component or tool? Maybe you've invented a better or simpler way of tackling a task? Do write in – for every tip that we publish, we'll reward you with UKP40 (or local equivalent)!**



# Comparing Raspberries and Oranges

## A good-value allrounder

By **Tam Hanna** (Slovakia)

uilding a project around a Raspberry Pi can work out relatively expensive. The made-in-China Orange Pi family comprises over a dozen different boards which are more appealing in terms of both price and computing power. In this article we take the Orange Pi Lite for a quick spin.

Eben Upton's Raspberry Pi Foundation has for a long time had the hobbyist market in single-board computers sewn up. The processor, which was cheap to obtain as it was being mass produced for use in smartphones, allowed the then manager at Broadcom to develop an embedded computer with which at least initially it was hard to compete on price.

It's often said that Chinese manufacturers always come up with a response to any product sooner or later. Allwinner, a processor manufacturer based in China, was not going to let their competitor Broadcom enjoy this success without a fight, and one answer that employed their devices was the Orange Pi, which came out in 2015. Since then the family has grown, and more than a dozen different models can be found at AliExpress.

An engineer is always keen to try to find ways to reduce costs and increase processing power. The Orange Pi family [1] from Shenzhen Xunlong will appeal to anyone who wants to control external hardware using a Linux-based computer offering a similar level of performance to a Raspberry Pi board. A particularly attractive feature of the Chinese boards is that they offer good Wi-Fi range. And perhaps some people will also prefer to look for an alternative to the Raspberry Pi if they do not find themselves aligned with the political inclinations of the Raspberry Pi Foundation.

### Availability

Although Orange Pi controller boards are available from various distributors all over the world, the true connoisseur will want to buy direct from the manufacturer in China. Shenzhen Xunlong offers a range of kits and accessories on their website, and the products are also available via AliExpress.

The question then arises as to which boards are the most interesting to experiment with. The author has divided up the available controller boards according to the scheme shown in **Figure 1**. It is worth noting that a good proportion of the boards (the 'Zero' family is an exception) use the pinout shown in **Figure 2**, which is by and large compatible with that of the

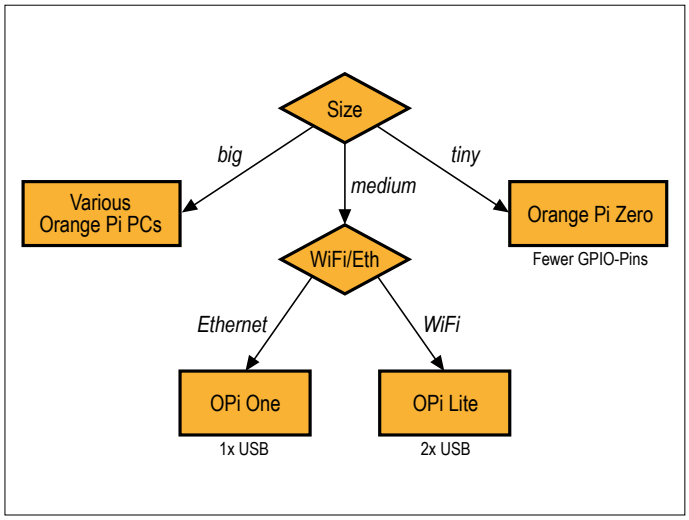



Figure 1. An overview of the family members.

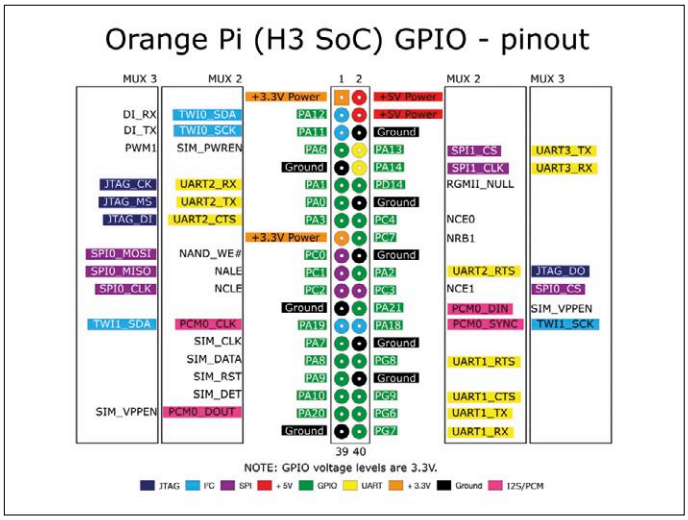



Figure 2. The layout of the 40-way header is in places reminiscent of that on the Raspberry Pi (source: www.orangepi.org).

Raspberry Pi; see also **Figure 3**.
In the following experiments we will use the Orange Pi Lite. Weighing in at some 28.8 grams it is lighter than the Raspberry Pi 3, and it also costs less, at around twelve dollars. Nevertheless it offers two USB ports and a Wi-Fi antenna. The Orange Pi One, on the other hand, has just one USB port instead of two, and Ethernet instead of Wi-Fi, and comes in even cheaper at around nine dollars.

## Configuration

We have to say one thing for the Raspberry Pi Foundation: if you buy their processor boards you will find that support in terms of the software and the ecosystem is excellent. Also, a Raspberry Pi can be powered from any ordinary micro USB charger, which makes it very easy to get started.
The Orange Pi Lite also has a micro USB port but, amusingly enough, cannot be powered from it because of the interface components used: you can find out more about this at [2]. Instead, power at 5 V has to be applied at the power jack or at pins 2, 4 and 6 of the header (pins 2 and 4 to 5 V, pin 6 to ground). This inconvenience is present across the Orange Pi family: if you need to be able to power your computer over the micro USB port, you will have to avoid Shenzhen Xunlong's products.
The next problem is software. The operating system images that can be found on the manufacturer's website at [3] are little more than window dressing. The links sometimes fail to work, and any file you do manage to download will likely be poorly supported.
The Armbian distribution has established itself more or less as a standard among the Orange Pi user community: downloads are available at [4]. Note that the Armbian distribution builds are different for the different members of the Orange Pi family. With our Orange Pi Lite we will be using a Debian-based distribution, but the Ubuntu-based version should work just as well. The differences between the versions are in general only small but they do manifest themselves from time to time and that means that you need to take extra care when testing.
Download the installation image [4a] and write it onto a sufficiently high-capacity SD card. Connect the processor board to an HDMI display, mouse and keyboard, and then apply power. Don't be surprised if it takes a minute or so before you see anything happen: the Orange Pi boards generally take a long time to start up and present an image on the screen.

## Android on board

The internal flash memory of the Orange Pi PC Plus contains a Chinese version of Android that is reportedly quite usable.

## It runs Linux!

Armbian images are classical computer operating systems: when first started up they present the user with a login prompt. The credentials to use at this point vary from image to image: on the Orange Pi Lite you can log in with the user name 'root' and password '1234'.
After the first login the operating system will prompt you to change your password for security reasons and to run an installation wizard. The user name has to be supplied, but the other fields can be left blank by simply repeatedly pressing 'Enter'. After that, you can reboot the system and you will be presented again with Armbian's graphical user interface (see **Figure 4**).
Armbian is Debian, a Linux distribution with a long history which also serves as the basis for the Raspbian distribution used on the Raspberry Pi. It follows, therefore, that at the application level the Raspberry Pi and the Orange Pi are more or less compatible. The first small difference comes in how external hardware is controlled. Where Raspberry Pi uses the WiringPi library, the people of Shenzhen Xunlong bring you WiringOP. The library must be downloaded and compiled before it can be used, which (assuming all goes well) can be done using the following sequence of commands:

```
cd ~
mkdir opigpiolib
git clone https://github.com/zhaolei/WiringOP.git -b
```

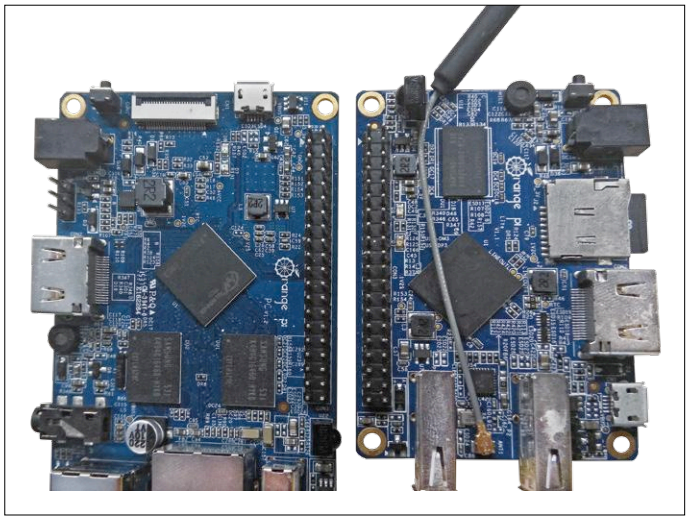


Figure 3. Watch out! The orientation of the headers on the two boards is different.




Figure 4. Just like the Linux we know and love.

```
$ gpio readall
 +-----+-----+----------+------+---+-Orange Pi+---+---+------+----------+-----+-----+
 | BCM | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | BCM |
 +-----+-----+----------+------+---+----++----+---+---+------+----------+-----+-----+
 |     |     |     3.3v |      |   |  1 || 2  |   |   |      | 5v       |     |     |
 |  12 |   8 |    SDA.0 | ALT5 | 0 |  3 || 4  |   |   |      | 5V       |     |     |
 |  11 |   9 |    SCL.0 | ALT5 | 0 |  5 || 6  |   |   |      | 0v       |     |     |
 |   6 |   7 |  GPIO.7  | ALT3 | 0 |  7 || 8  | 0 | ALT3 | TxD3     | 15  | 13  |
 |     |     |       0v |      |   |  9 || 10 | 0 | ALT3 | RxD3     | 16  | 14  |
 |   1 |   0 |     RxD2 | ALT3 | 0 | 11 || 12 | 0 | ALT3 | GPIO.1   | 1   | 110 |
 |   0 |   2 |     TxD2 | ALT3 | 0 | 13 || 14 |   |   |      | 0v       |     |     |
 |   3 |   3 |     CTS2 |   IN | 0 | 15 || 16 | 0 | ALT3 | GPIO.4   | 4   | 68  |
 |     |     |     3.3v |      |   | 17 || 18 | 0 | ALT3 | GPIO.5   | 5   | 71  |
 |  64 |  12 |     MOSI |   IN | 0 | 19 || 20 |   |   |      | 0v       |     |     |
 |  65 |  13 |     MISO |   IN | 0 | 21 || 22 | 0 | ALT3 | RTS2     | 6   | 2   |
 |  66 |  14 |     SCLK |   IN | 0 | 23 || 24 | 0 | ALT4 | CE0      | 10  | 67  |
 |     |     |       0v |      |   | 25 || 26 | 0 | ALT3 | GPIO.11  | 11  | 21  |
 |  19 |  30 |    SDA.1 | ALT4 | 0 | 27 || 28 | 0 | ALT4 | SCL.1    | 31  | 18  |
 |   7 |  21 |  GPIO.21 | ALT3 | 0 | 29 || 30 |   |   |      | 0v       |     |     |
 |   8 |  22 |  GPIO.22 |  OUT | 0 | 31 || 32 | 0 | ALT3 | RTS1     | 26  | 200 |
 |   9 |  23 |  GPIO.23 |  OUT | 1 | 33 || 34 |   |   |      | 0v       |     |     |
 |  10 |  24 |  GPIO.24 |  OUT | 0 | 35 || 36 | 0 | ALT3 | CTS1     | 27  | 201 |
 |  20 |  25 |  GPIO.25 |  OUT | 0 | 37 || 38 | 0 | ALT3 | TxD1     | 28  | 198 |
 |     |     |       0v |      |   | 39 || 40 | 0 | ALT3 | RxD1     | 29  | 199 |
 +-----+-----+----------+------+---+----++----+---+---+------+----------+-----+-----+
 | BCM | wPi |   Name   | Mode | V | Physical | V | Mode | Name     | wPi | BCM |
 +-----+-----+----------+------+---+-Orange Pi+---+---+------+----------+-----+-----+
```

Figure 5. The GPIO options on the Orange Pi are rather complicated.

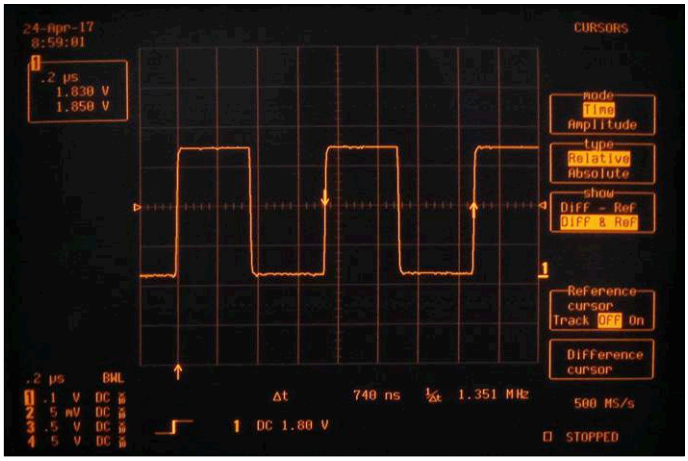



Figure 6. The square wave looks usable.

```
   h3
cd WiringOP
chmod +x ./build
sudo ./build
```

In the next step we look at a small example program that generates a square wave on a GPIO pin. To do this, create a file called 'main.c' with the following contents:

```
#include <wiringPi.h>
int main (void) {
  wiringPiSetup () ;
  pinMode (0, OUTPUT) ;
  for (;;) {
    digitalWrite (0, HIGH) ;
    digitalWrite (0,  LOW) ;
    digitalWrite (0, HIGH) ;
    digitalWrite (0,  LOW) ;
 }
  return 0 ;
}
```

The process of compiling this code is largely identical to the familiar process on the Raspberry Pi:

```
$ gcc -Wall main.c -lwiringPi
$ sudo ./a.out
```

## A sensitive flower

Orange Pi boards are extremely sensitive to a supply voltage that is too low: this can lead to crashes when graphics mode is enabled and read failures on the SD card. If you have problems booting up your new computer, one of the first things to check is the voltage drop across the supply cables. Note that the board can draw currents of up to 800 mA!

On some versions of the Orange Pi you will see compilation errors that stem from the fact that the 'pthread' library is not present:

```
$ gcc -Wall main.c -lwiringPi
//usr/local/lib/libwiringPi.so: undefined reference
   to 'pthread_join'
//usr/local/lib/libwiringPi.so: undefined reference
   to 'pthread_create'
//usr/local/lib/libwiringPi.so: undefined reference
   to 'pthread_cancel'
collect2: error: ld returned 1 exit status
```

This in turn results from misconfiguration of the gcc compiler, which is not automatically linking in the pthread library. To solve this it is simply necessary to add another option to the compiler command line as follows:

```
$ gcc -Wall main.c -lwiringPi -pthread
$
```

It should be possible to obtain the layout of the individual pins on the GPIO header using the command '`gpio readall`'. For simplicity we give an example result from this command run on a 40-pin Orange Pi PC Plus in **Figure 5**.

We can now team up the processor board with a duo of instruments: a modulation domain analyzer and a digital storage oscilloscope. **Figure 6** and **Figure 7** show the results you might get.

### Enabling the UART

In our next example we will show how to read an RFID card. The Seeed RFID transponder module, which operates at 125 kHz, is ideal for this task as it only requires a small amount of configuration. It is available from Mouser (order code 713-113020002) for around twelve dollars including antenna. **Figure 8** shows the electronics required to connect up this module: whereas the Raspberry Pi 3 is to some extent 5 V tolerant, this is not the case for the Orange Pi and a (not especially sophisticated) level shifting circuit is required.

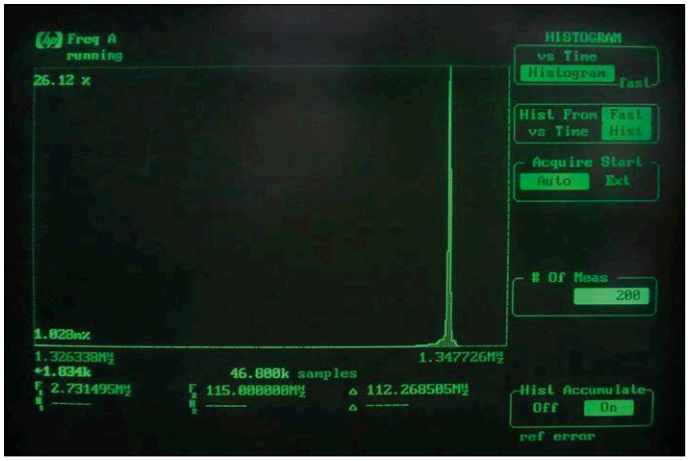



Figure 7. The stability of the waveform, too, leaves not much to carp about.

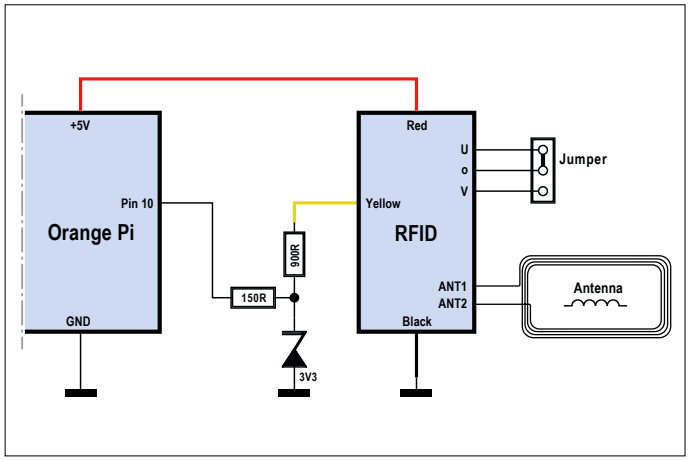



Figure 8. A voltage level shifter is essential.

Allwinner supplies the hardware configurations for its SoCs in the form of a binary file which is loaded into the device by the kernel as part of the boot process to set up the voltage regulators, clock sources and various other subsystems. The UART used in our program example is not normally accessible, as can be seen from a look at the /dev directory where only one UART (ttyS0) is present:

```
$ ls -l

. . .

crw--w---- 1 root tty      4,   9 Apr 12 06:07 tty9
crw--w---- 1 root tty    251,   0 Apr 12 06:08 ttyS0
crw------- 1 root root   248,   0 Apr 12 06:07 tv
```

To get around this problem the first step is to modify the configuration file that is loaded into the hardware during the boot process. The scripts used for the various boards can be found on the SD card in the directory '/boot/bin', which under Armbian looks like this:

```
$ pwd
/boot

$ realpath script.bin
/boot/bin/orangepipcplus.bin
```

These configuration files can exist in two different forms: as well as the binary version that the processor itself understands there is a '.fex' format, which is human-readable. We can convert the binary file (the one for our processor is, as we discovered above, called '/boot/bin/orangepipcplus.bin') to readable form using the following command:

```
$ sudo bin2fex orangepipcplus.bin orangepipcplus.fex
[sudo] password for spotter: . . .
fexc-bin: orangepipcplus.bin: version: 1.2
fexc-bin: orangepipcplus.bin: size: 36392 (84
   sections), header value: 36392
```

The structure of a .fex file is reminiscent of that of a .ini file, and is documented in detail at [5]. Of interest to us at the moment is how the UART that is connected to the RFID reader is configured. Edit the block so that it contains the following information:

```
[uart3]
uart_used = 1
```

```
$ ls
aw-som-a20.bin          bananapiprolcd7.bin  lime-a10-lcd.bin  nanopiair.bin       orangepilite.bin    pcduino2.bin
bananapi.bin            beelinkx2.bin        lime-a10.bin      nanopim1.bin        orangepione.bin     pcduino3.bin
bananapilcd7.bin        cubieboard.bin       lime-a33.bin      nanopim1plus.bin    orangepipc.bin      pcduino3nano.bin
bananapim1plus.bin      cubieboard2.bin      lime.bin          nanopineo.bin       orangepipcplus.bin
bananapim1pluslcd7.bin  cubieboard2dual.bin  lime2-emmc.bin    olinux-som-a13.bin  orangepiplus.bin
bananapim2plus.bin      cubietruck.bin       lime2.bin         orangepi.bin        orangepiplus2e.bin
bananapipro.bin         lamobo-r1.bin        micro.bin         orangepi2.bin       orangepizero.bin
```

Meanwhile, the file that is actually used to initialize the board lies in the directory immediately above this one under the name 'script.bin'. However, this is in fact a symbolic link, and the `realpath` command can tell us the real file to which it corresponds:

```
uart_port = 3
uart_type = 4
uart_tx = port:PA13<3><1><default><default>
uart_rx = port:PA14<3><1><default><default>
uart_rts = port:PA15<3><1><default><default>
uart_cts = port:PA16<3><1><default><default>
```

For comparison, a block that describes a disabled UART looks like this:

```
[uart2]
uart_used = 0
uart_port = 2
. . .
```

However, it is not sufficient simply to modify the .fex file and reboot, as this will not change the behavior of the machine. We first have to convert the edited .fex file back into a .bin file, using the command `fex2bin`.

```
$ sudo fex2bin orangepipcplus.
fex orangepipcplus.bin
```

And now at this point we reboot the machine and admire the fruits of our handiwork: the new serial port appears in the /dev directory

```
crw--w---- 1 root tty
   251,   0 Apr 24 13:34 ttyS0
crw-rw---- 1 root dialout
   251,   3 Apr 24 13:34 ttyS3
crw------- 1 root root
   248,   0 Apr 24 13:34 tv
```

Enabling and configuring a UART is just one of the possibilities opened up by the ability to edit a .fex file. Another example would be changing the processor clock frequency to gain more processing speed or to save power, and more information on this topic can be found in the relevant forums. But meanwhile, let's crack on with reading some RFID data.

**Listing 1. Reading an RFID card over a UART.**

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringSerial.h>

int main ()
{
  int fd ;
  int count ;
  unsigned int nextTime ;

  if ((fd = serialOpen ("/dev/ttyS3", 9600)) < 0)
  {
    fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno)) ;
    return 1 ;
  }

  if (wiringPiSetup () == -1)
  {
    fprintf (stdout, "Unable to start wiringPi: %s\n", strerror (errno)) ;
    return 1 ;
  }

  nextTime = millis () + 300 ;

  for (count = 0 ; count < 256 ; )
  {
    delay (3) ;

    while (serialDataAvail (fd))
    {
      printf (" -> %3d", serialGetchar (fd)) ;
      fflush (stdout) ;
    }
  }

  printf ("\n") ;
  return 0 ;
}
```

```
tamhan@TAMHAN14: ~
$ sudo ./a.out
 ->   2 ->  48 ->  49 ->  48 ->  70 ->  53 ->  67 ->  67 ->  65 ->  51 ->  66 ->
  65 ->  51 ->   3

 ->   2 ->  48 ->  49 ->  49 ->  48 ->  57 ->  49 ->  69 ->  65 ->  67 ->  68 ->
  65 ->  55 ->   3
```

Figure 9. Two different RFID cards duly recognized: press 'Enter' after each scan to split up the lines of data.

## Scanning an RFID card

All we need to make a fully-functional card reader is a program to read serial data delivered from the UART. Its first job is to open the connection to the UART: see the first part of **Listing 1** [6].

The next step is to read in the incoming data. In view of the ever-increasing importance of RFID technologies the author feels a brief digression is justified. There are two basic types of reader. The first group, to which our module belongs, reads all the information it can find on the card and sends this as a serial data stream to whatever it is connected to. Readers in group number two implement a Wiegand interface. This involves a protocol that communicates using pulses on one of two data lines.

## Risk of collision!

When several programs try to access the same serial bus at the same time, collisions can occur. Experience shows that the operating system does not detect this and the individual programs only receive fragments of the data they are expecting.

Despite being limited to data lengths of a mere 26 bits or 31 bits, the Wiegand protocol shows no sign of going away and so remains relevant in embedded control applications. Most RF readers, in fact, are available in both Wiegand and UART editions, with the UART edition being the more sophisticated. For simplicity our program gathers data one byte at a time (see Listing 1) and displays them in the terminal window.
We are now ready for a first test. Connect the card reader to the computer, compile the program in the usual way, and bring a card near to the red reader coil. The display in the terminal window should look something like **Figure 9**.

### Going further

As we have seen, the Orange Pi is a classical embedded controller board that runs the Linux operating system. That means that it supports a range of protocols such as I²C and SPI in much the same way as the Raspberry Pi and its friends.
Anyone who is willing to get to grips with the Linux I²C and SPI APIs (or who is willing to use the WiringOP library) will find it easy to connect a diverse range of peripheral devices, including displays and sensors in all shapes and sizes.

### Conclusion

When the first Orange Pi computer saw the light of day, many had their doubts regarding the expected longevity of the manufacturer. Two years later, and Shenzhen Xunlong is still going strong and has brought many new models to market. Also, practically all of the devices are still readily available.
Using an Orange Pi is particularly beneficial in mass-produced systems: a cost reduction to the tune of 50 % or even more is not something to be sniffed at. In small quantities there is the question of development costs: if you are not an expert Linux user, you will probably be better off with the support of the larger Raspberry Pi community. ⏮

(160456)

### Web Links


[1] www.orangepi.org/
[2] https://www.youtube.com/watch?v=4zICevDOhr8
[3] www.orangepi.org/downloadresources/
[4] www.armbian.com/download/
[4a] https://dl.armbian.com/orangepilite/
[5] http://linux-sunxi.org/Fex_Guide
[6] www.elektormagazine.com/160456



Advertisement

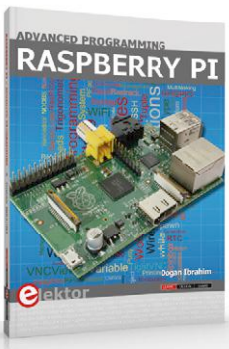
LEARN DESIGN SHARE

# Raspberry Pi @ Elektor

It's very easy to connect the Raspberry Pi 3 to a computer or a television set. You can program this high-performance computer yourself, use it to play videos, play games with it, and use it do develop your own highly innovative concepts.

**www.elektor.com/rpi3**




Raspberry Pi Advanced Programming
www.elektor.com/rpi-book




Raspberry Pi 3 Starter Kit (Deluxe)
www.elektor.com/rpi-deluxe



pi-top DIY Laptop Kit
www.elektor.com/pi-top

elektor

More Raspberry Pi at www.elektor.com/rpi

HOMELAB PROJECT

# Ultimate Bugblaster

## No more flabby flyswatters

By **Bernhard Schriefer** and **Hannes Stein** (Germany)



Graphic: shutterstock.com

For some time electric flyswatters have been available at giveaway prices. Made with a metal grid and a plastic handle, they look rather like miniature tennis racquets. Two AA-size batteries produce a voltage in the region of 1 to 3 kV and use this to charge a small capacitor. This level of energy is just enough to annihilate mozzies (mosquitoes) and render house flies unflightworthy for just a short while. This project describes some improved high-voltage electronics that take us right to the limits of what is technically achievable.



Figure 1. The second prototype (center) compared with some commercial flyswatters.

Electric flyswatters have a number of advantages over their clumsier mechanical counterparts. These weapons of mozz destruction are an elegant means of denying mozzies permission to fly — without leaving unsightly evidence of these encounters on your white-painted walls. Even fruit flies, which are smaller than the separation between the electrified wires (about 4 mm), are unfailingly terminated. They bridge enough of the spark gap to cause a flashover and initiate their new incarnation. The style of grid that uses parallel wires (**Figure 1**) has proven itself in practice better than those variants with three mesh gratings in which the voltage is applied between the gratings.

Larger flying insects pose a greater problem, however. Whilst mosquitoes and fruit flies are eliminated quite reliably, houseflies and bluebottles usually fall to the ground, then fly back and forth again within a few seconds. If they are not 'dispatched' while on the ground level, your labors are in vain. At this stage every electronicist will ask the obvious question: can we achieve greater effect with more volts and amps?

Contact between fly and grid sets off an immediate flashover (electric arc). In **Figure 2** you can see how the voltage

between the wires of the grid collapses immediately on contact with a fly, and consequently the storage capacitor (C in the simplified diagram = **Figure 3**) is drained of its energy. It stands to reason now that we need to increase the stored energy and intensify the arc. It's clear that a minimum firing voltage is necessary to trigger a spark, even with smaller fruit flies. In our experience, around 3 kV is a good value with 4 mm separation between the wires.

### Optimization

Commercial products use a flyback converter that generates about 2.8 kV from a battery voltage of 3 V at best. Cheap models manage barely 1.6 kV. In the best flyswatters the HT (high tension or high voltage) is loaded with a 22 MΩ resistor (R in Figure 3). Presumably this has two functions: on one hand it limits the voltage, simply because the flyback converter has only a restricted output, and on the other this ensures rapid discharge of the storage capacitor when the pushbutton is released. Some models, however, have a mechanical contact ($S_b$ in Figure 3) installed, which, after the pushbutton is released, simply short-circuits the voltage — a very elegant and safe solution.
Two conceivable improvements could be implemented quickly:

- Connecting additional capacitors in parallel for more stored charge. For testing we installed an additional 22 nF with 2 kV load capacity.
- Increasing the value of the resistor. A good 2 kV is feasible with 100 MΩ, even using low-cost swatters.

As simple as these optimizations were, they remained equally ineffective. Our experiments showed that these improvements were measurably detectable, but this did not really impress or scare the flying insects we confronted. At this point our investigative spirit was really challenged.
The case molding of the swatter we first used did not have any room for extra capacitors. So another flyswatter was procured that offered more space for after-market add-ons. Unfortunately the flyback generator in this specimen was not quite as effective, as it delivered only 1.6 kV. **Figure 4** shows that room was found for several additional 2-kV capacitors. This measure alone increased the stored charge from originally 45 µC (micro-coulombs) to at least 465 µC, which meant more than a tenfold improvement!
This step-up may sound impressive but once again the emergence of quality from quantity did not produce much of an experience overall. The zap was louder but that was it. The desired goal of a far better flyswatter was still not achieved.

### Brute force

The motto 'more is better' is a fundamental rule of thumb for every electronics. If the previous 'more' is still insufficient, then we must increase the quantity all the more so. The next consideration was therefore: how far can we raise the voltage? What will this achieve? How can we increase the capacitive charge more effectively?
As you know, and as **Figure 5** demonstrates impressively, for the same physical volume electrolytics deliver appreciably greater capacity than foil capacitors. The high-voltage capacitor

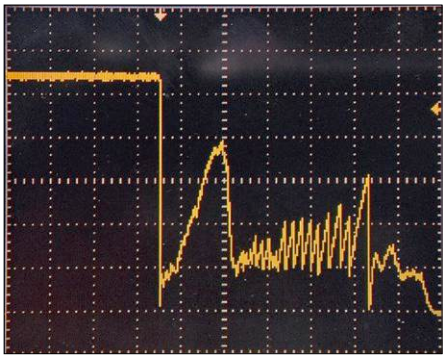


Figure 2. Voltage characteristics on the grid when a fly makes contact.

used can develop a charge of 44 µC using 22 nF at 2 kV, corresponding to a stored energy of 88 mJ. Believe it or not, at 450 V a 10-µF electrolytic stores 2,000 mJ. In terms of volume, the electrolytic contains 62 µC/cm³ against a pitiful 2.7 µC/cm³ in the blue casing. Energetically speaking, that's a ratio of 124 to 1.2 J/cm³. Using an electrolytic capacitor, you can store an entire order of magnitude more energy

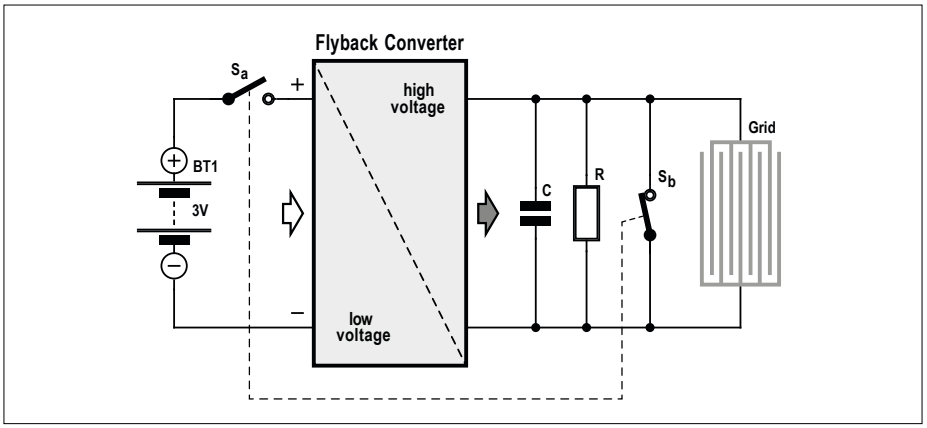



Figure 3. Simplified diagram of the internals of a commercial flyswatter. These either include a resistor R for slowly discharging the storage capacitor or else a secondary contact $S_b$ short-circuits the capacitor when S1 is released.

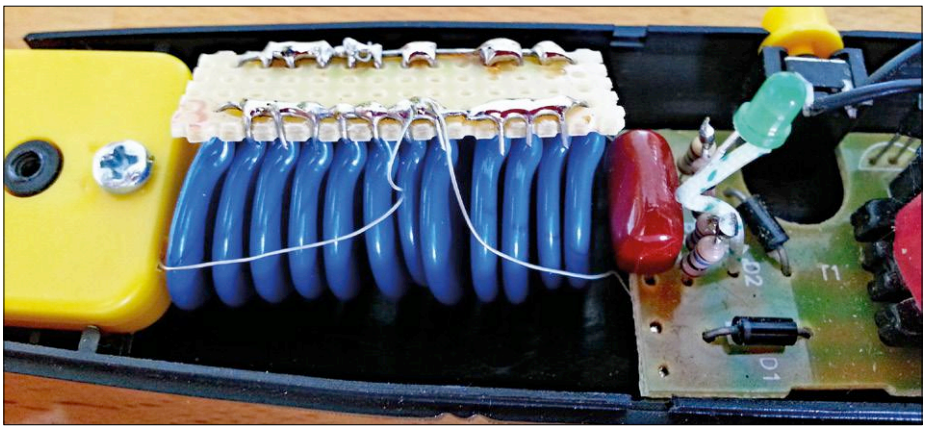


Figure 4. This flyswatter had enough room for a number of additional capacitors.

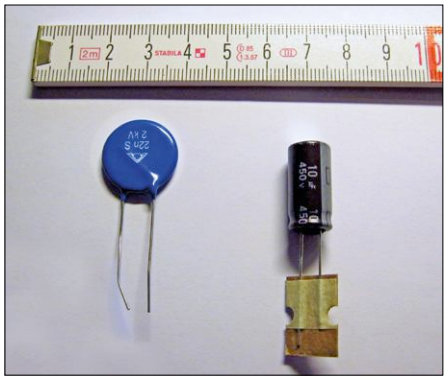


Figure 5. Size comparison of electrolytic and normal HT capacitors.

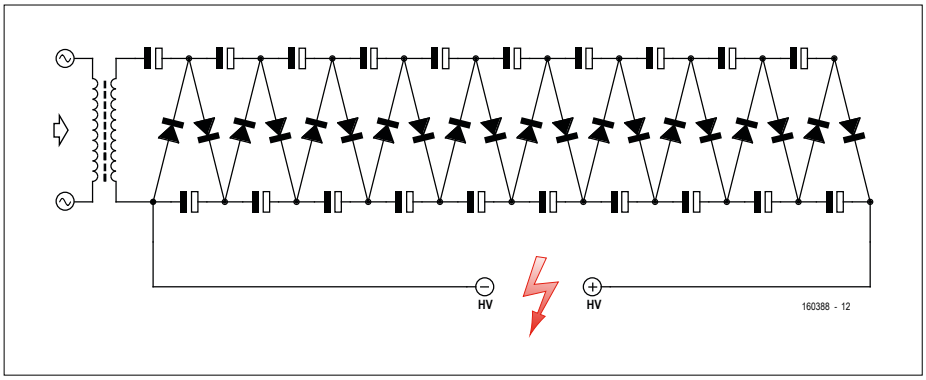



Figure 6. Voltage multiplier for a factor of 10 in a Villard circuit. Each electrolytic is charged to no more than 450 V.

with a given volume. And that should go a long way in delivering the 'more' that we need...
General-purpose electrolytics are available only up to 450 V rating. Since our preliminary tests indicated that we should aim for a total voltage of at least 4 kV, a series chain of electrolytics is mandatory.

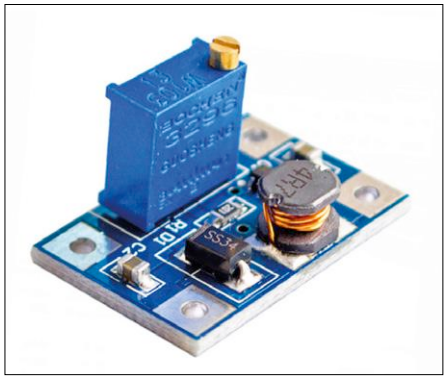


Figure 7. The SX1308 step-up converter module provides up to 2 A and can be bought cheaply on eBay.

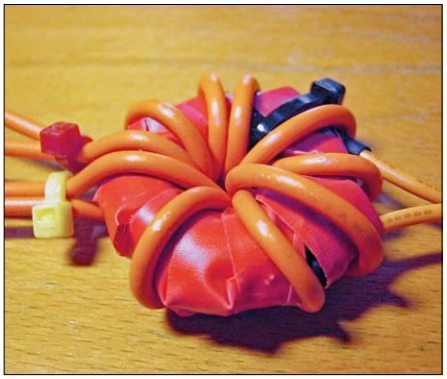


Figure 8. Homebrew HT ring core transformer. Note the insulation between the secondary and primary sides.

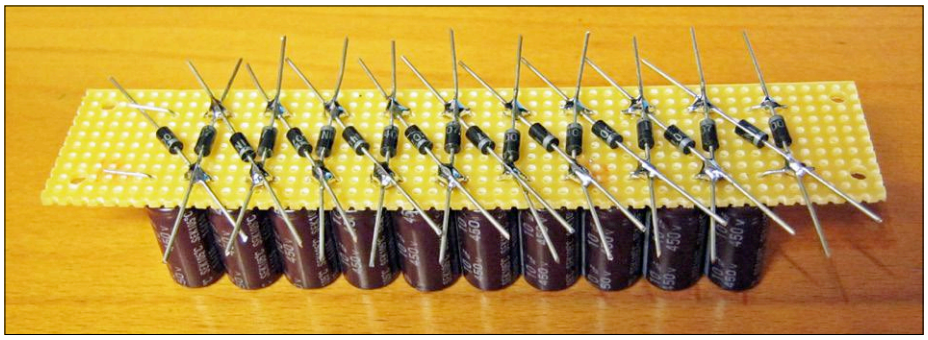


Figure 9. HT cascade made up of 20 diodes and 20 HT electrolytics.



Figure 10. Voltage divider made up of 30 × 10 MΩ and a 100 kΩ precision resistor in the center.

Accordingly and to have a little in reserve, we must connect ten electrolytic capacitors in series, which is adequate even for 4.5 kV. Together with the charging capacitors in the Villard circuit [1] (**Figure 6**), we need to incorporate 20 electrolytic capacitors! When electrolytics are connected in series it is vital that the voltage is distributed evenly and no single capacitor is overloaded.

A further requirement was the ability to still produce the specified HT voltage with partly discharged batteries. The flyswatter should operate constantly up to a certain limit independently of the state of the batteries. Initially it was planned to use two AA cells for powering it, as with off-the-shelf swatters. Problems arose with this approach, however:

- the voltage transition from 3 to 1.6 V (from fresh to flat batteries) was equivalent to virtually halving the supply voltage;
- the microcontroller management system (this is after all a hi-tech flyswatter) required a minimum of 2.7 V;
- with 20 electrolytics rated 450 V and 10 µF, the stored energy level amounted to around 20 Ws (watt-seconds). For a 10-s charging time the HT generator would require an output capacity of 2 W. To this you would need to factor in the conversion losses. The resulting current draw (around 1 A) would be impossible with partly discharged AA cells;
- the voltage ratio required of the transformer would be high, particularly when a demand of 450 $V_{pp}$ needs to be produced from below 2 V.

The first solution was to raise the voltage to a constant 4 V using a 2 A step-up converter module of the type SX1308 (**Figure 7**), which you can find on eBay for €2 to €3 / £2 to £3 / $2.5 to $3.5. The step-up converters tested were unable to produce 1 A across the entire discharge range of the batteries, however. So this strategy was rejected.

**Developing the circuitry**

In place of AA cells we now changed to lithium rechargeables. As well as the advantage of a higher nominal voltage of 3.6 V, the discharge rate of these batteries remains desirably stable over a broad range. In addition the internal resistance is really low. So the circuit was dimensioned for an operating voltage range of from 3.3 to 4.2 V. And why not go the whole hog? Two 18650-type Li-Ion batteries were wired in parallel. Additionally the SX1308 step-up converter was called into service to produce a constant operating voltage of 4 V for generating the HT.

Our first notion was to drive the primary winding with a normal half or full-wave bridge. Using bipolar transistors, we lose approx. 2 x 0.2 V in saturation voltage. MOSFETs would reduce the loss but they are more demanding in terms of drive circuitry. The converter can be simplified and made more efficient if we use two bipolar transistors in push-pull to drive the two primary windings. Then the transformer needs only to raise around 4 V to 225 $V_p$ = 450 $V_{pp}$.

We were unable to find a suitable transformer with the 2x1:65 turns ratio required for this. AC line (mains) transformers have significantly smaller voltage ratios. This also applies to the transformers that we removed from switch-mode power supplies for testing out. So we had to make the necessary transformer ourselves. Using a ring core that we had available (outer diameter 28.9 mm, inner diameter = 8.7 mm, width = 7.7 mm) we made 2×5 turns for the primary and 325 turns for the secondary winding. The cross-section of the wire is not really critical and conforms to the size of the core. We used 0.2 mm diam. (34 AWG) enameled copper for the secondary and 1 mm diam. (18 AWG) for the primary winding. **Figure 8** shows the result of these efforts. Following the transformer comes the classic HT cascade of Figure 6 with 20 1-kV diodes of the type 1N4007 each plus one high-voltage electrolytic each (**Figure 9**).

For measuring the resulting HT and reducing it to a voltage usable with the A-to-D converter of the microcontroller we initially provided a voltage divider from 300 MΩ to 100 kΩ. With this arrangement, however, the bulk of the electronics would lie at the potential of one of the two high-voltage sections. If you now touched grounded material like a radiator or a water tap (faucet), the insulation of the handle and the transformer would then have to withstand 4 kV, increasing the risk of electric shock for the person using the flyswatter unnecessarily. A better idea would be to divide the series resistor into 2 x 150 MΩ units and put the precision resistor and ground centrally. The potential difference between the grid wires and the electronics (and consequently the handle as well) is then halved to a little over 2 kV. Incidentally, with such high voltages and resistance, the number of resistors connected in series is determined not by their power loss but

Advertisement

Join the Elektor Community
Take out a GOLD Membership now!




GOLD MEMBERSHIP
- 6x Elektor Magazine (Print)
- 6x Elektor Magazine (PDF)
- 6x Elektor Business Magazine (PDF)
- Access to Elektor Archive
- Access to over 1,000 Gerber files
- Elektor Annual DVD
- 10% Discount in Elektor Store
- Exclusive Offers

GREEN MEMBERSHIP
- 6x Elektor Magazine (PDF)
- 6x Elektor Business Magazine (PDF)
- Access to Elektor Archive
- Access to over 1,000 Gerber files
- 10% Discount in Elektor Store
- Exclusive Offers

Also available:
The all-paperless GREEN Membership!
www.elektor.com/member

by their dielectric strength. For safety reasons, 30 units of 10 MΩ resistors in 0603 SMD format were employed (**Figure 10**). Then 'only' 150 V needs to be dropped per resistor. According to Adam Riese mathematics, on the 100 kΩ precision resistor formula we have just 1.5 V — suitable the measuring range of the A-to-D input of a microcontroller.

The control center of the circuit (**Figure 11**) is an ATtiny26. This monitors the battery and HT voltages and controls the status information LEDs. Once the defined HT voltage is reached, the drive to the transistors is stopped.

In the circuit's quiescent (inactive) state the microcontroller is in sleep mode, consuming less than 1 µA. It can therefore remain permanently connected to the battery. Operating pushbutton S1 activates the Controller using an Interrupt and applies power via MOSFET T3 to the step-up converter driven with around 1.2 MHz, also to T1 and T2. Our self-wound transformer is driven at around 90 kHz. S1 doesn't have to stand up to any great current therefore. Diodes D4 to D7 serve to produce the gate voltage (charge pump), in which all the voltage drop across D2 to D4 does is to ensure the maximum allowed gate voltage for T3 is not exceeded. With a somewhat more expensive MOSFET with a lower 'on' threshold you drop the need for the charge pump. The step-up converter is adjusted for 4 V. When T3 is switched on, the voltage multiplier is initially supplied directly from the battery

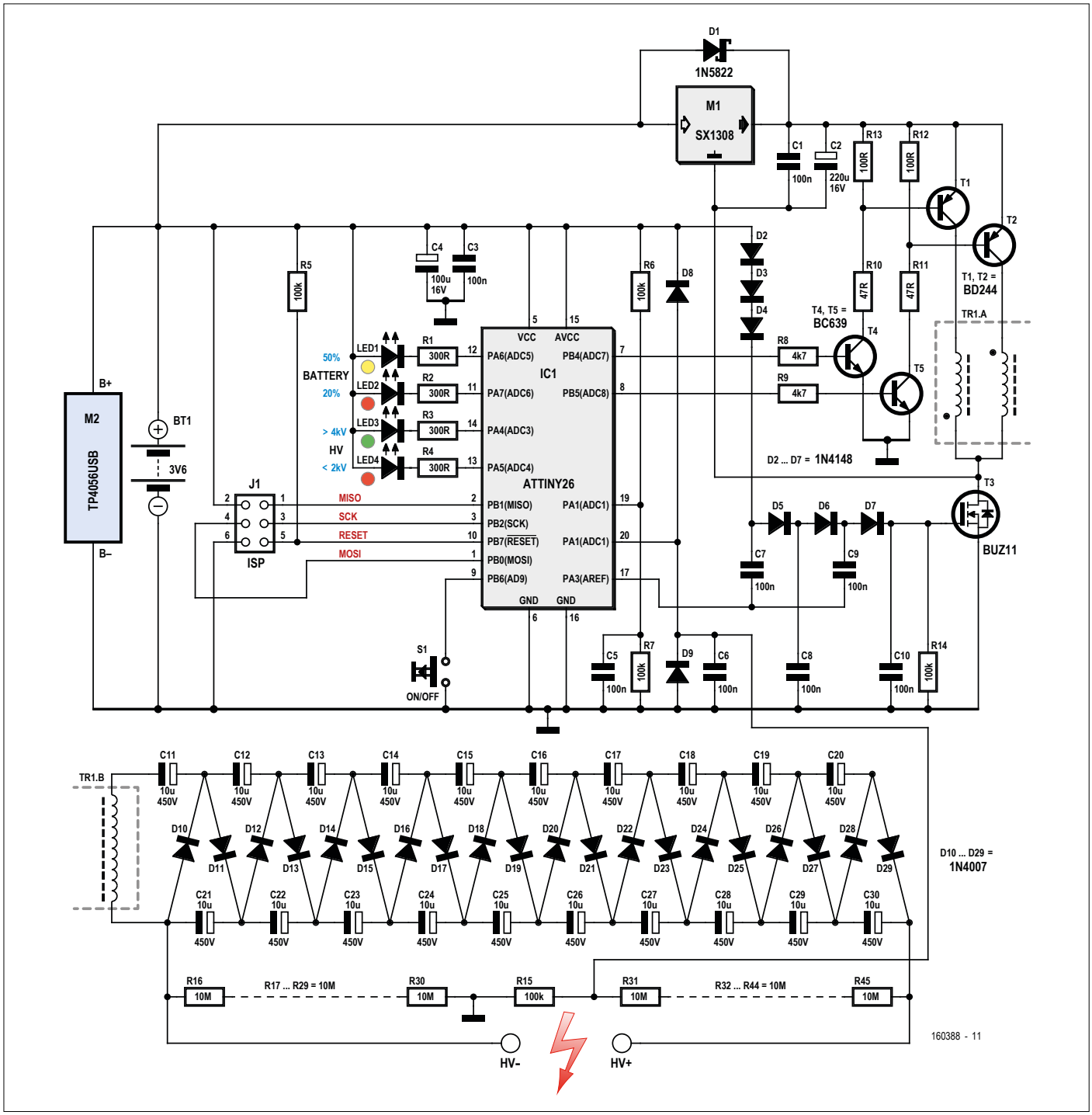



Figure 11. Complete schematic of the power swatter.

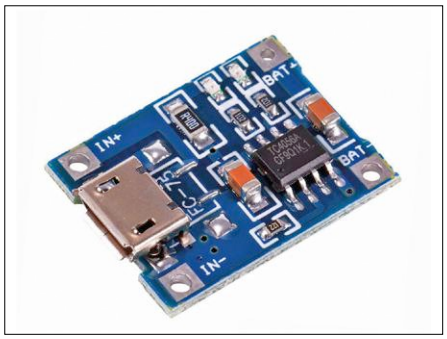



Figure 12. This ready-built module using a TP4056USB can charge a Li-Ion rechargeable battery via USB correctly with a current of up to 500 mA and is very affordable.

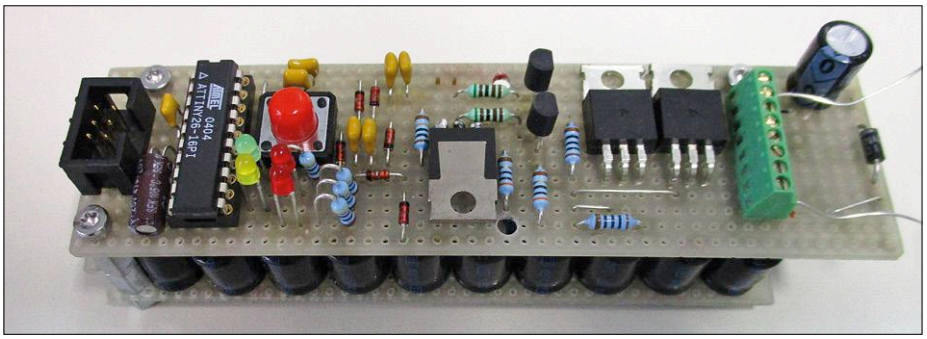


Figure 13. This is how the finished electronics look when mounted on perf board. On the lower deck is the HT section.

via D1; until the step-up converter turns on the voltage, it is bypassed by D1. Charging the Li-Ion batteries is done by USB, using the TP4056 charge controller. This ready-made module (**Figure 12**) is another thing that can be had on eBay for very little money. The completed electronics (without the HT voltage divider) are shown in **Figure 13**.

Firmware for the little Atmel microcontroller was written in Atmel Studio; the source code can be downloaded from the Elektor web page for this article [2]. IC1 is programmed for using the internal 4 MHz clock generator by Fuses (**Figure 14**).

So far two prototypes have been built, which demonstrated that the mechanical construction is not totally uncritical. With the grid style used flashovers took place in the framework and between the wires at voltages above 4.2 kV. Two measures will mitigate this:

- encasing the framework in electrical potting resin;
- limiting the maximum HT voltage to 4 kV by software.

With these two precautionary measures you can stay on the safe side and zapping will take place only when an insect approaches the grid wires.

### Test results

As already mentioned, the authors constructed two prototypes for test purposes using perf board. The larger 'man size' construction on a wooden batten (**Figure 15**) was simply a feasibility study. It does work but its practicality is limited by its length and the additional mass of the batten, both of which make it somewhat unwieldy. Swiping house flies with such a monster swatter is not a good idea, especially if you don't want your furnishings to suffer too much.

The second version built (**Figure 16**) is visibly easier to handle and more compact. The electronics are fixed above the high-voltage unit. So you can get a fix on the difference between this and the commercial flyswatters, we produced a graphic (**Figure 17**) showing the voltage characteristics of the devices in Figure 1. Incidentally the 'bug zapper' (far right in the photo) now comes in a better version. This chart shows how our flyswatter (gray line) produces more than 4 kV over the entire usable voltage range and is thus vastly superior to the commercial products.

In fact our swatter works brilliantly. It

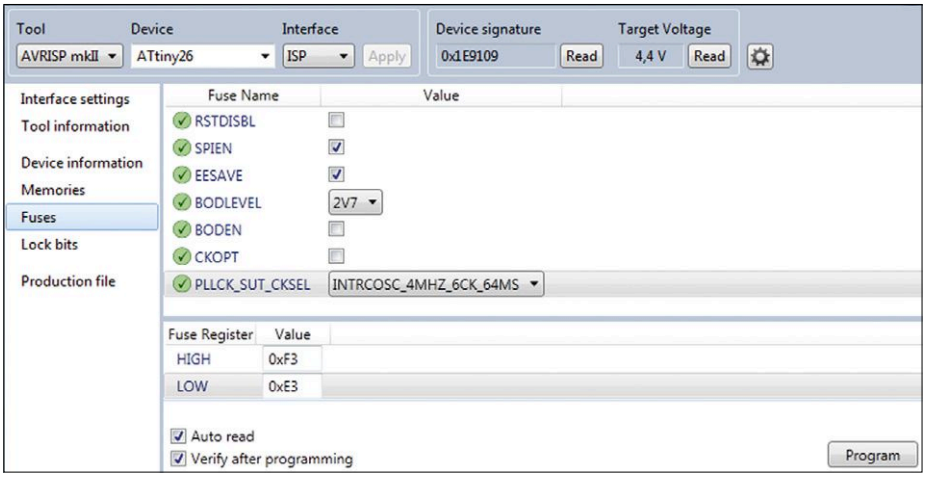



Figure 14. The Fuses for the ATtiny26 are set like this. The 4-MHz clock frequency is crucial.

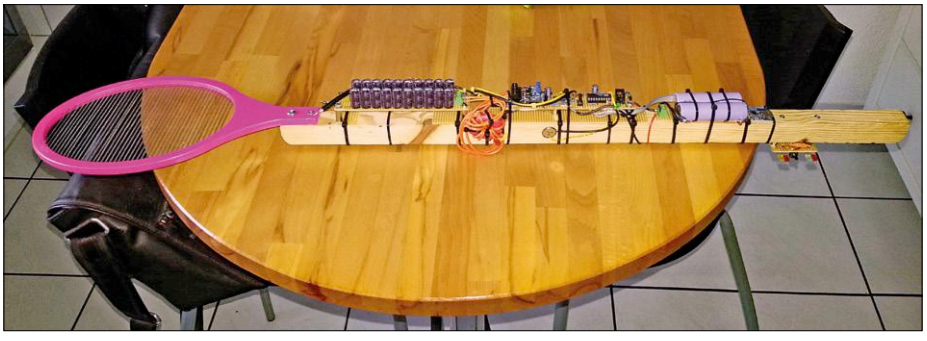


Figure 15. This monster prototype on a wooden batten turned out rather long and thus unwieldy. It is rather just a feasibility study.

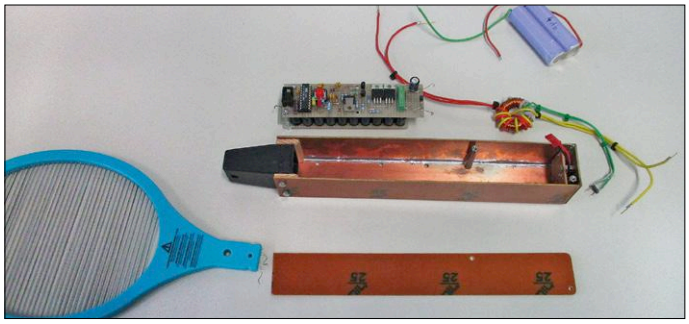


Figure 16. The second prototype is already significantly smaller and more practical to use.

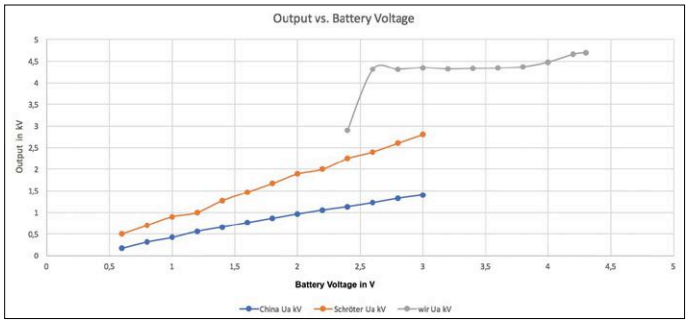



Figure 17. Comparison of the HT from a low-cost flyswatter, a better-class example and our self-built design in relation to the battery voltage. Building your own is definitely worthwhile, as you can see.

# Hi-tech + high tension = ultimate bug blaster



terminates fruit flies, gnats and midges dependably. On the other hand its effect on flies is unfortunately not quite so convincing as hoped. Even so, it takes them a couple of minutes to wake from their coma and fly away. This gives you adequate time to deal with them.

Currently the voltage is indicated with two status LEDs but in practice you follow flies with your eyes and don't look at the LEDs. An audible means of indicating the status of the HT might be better, for example by varying the pitch of a tone. The high resistance of the voltage divider means that loading on the HT cascade is minimal, which is also why a substantial voltage persists for a good minute after use. It would be pretty unpleasant to touch, so this is certainly an area for improvement.

The authors will continue to look into the effect on house flies too. It's likely the size of the capacitors will need to be increased drastically or else a new strategy may occur to us. If you have a bright idea, please share it with all of us!

## And finally: play safe

This must be said and heeded without fail: it is absolutely *not* safe to fool with electrolytics charged at 4 kV! With circuits like this, loud bangs may not be the only outcome, even at the construction test. If capacitors discharge themselves through the human body, the effect will be similar to what happens to a fly. It's only our larger size that protects us humans from the truly horrendous things that may occur. Even then our skin is not a perfect insulator and you can probably imagine the pain of an intense electric shock. And if someone has a weak heart, you may well get a visit from the emergency physician and later on from the police.

High-power flyswatters of this kind belong *only* in the hands of competent and responsible people. So never leave them lying around and far less ever allow children to operate them. All too often they cannot resist 'looking with their fingers' and when you hear a loud scream, it may already be too late for resuscitation. Their death will be forever on your conscience. So please be sensible.

(160388)

### Web Links

[1] Voltage multipliers: https://en.wikipedia.org/wiki/Voltage_multiplier

[2] Firmware: www.elektormagazine.com/160388


**About the authors**

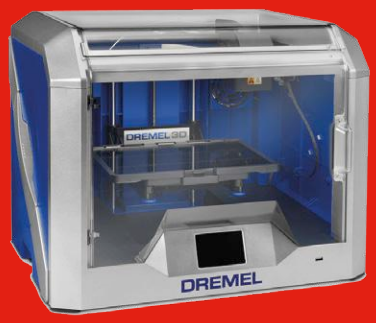
Bernhard Schriefer and Hannes Stein are keen engineers and radio amateurs from Berlin. Like many other engineers, both of them have enjoyed an interest in electronics since childhood.

# Elektor Video Olympics

## Bolt for that camera now!

**True, the Elektor Video Olympics are in full swing, but that doesn't mean that we're closed for sign-up and participation. Better late than never! And wouldn't the Grand Prize be a great asset to your humble homelab? Of course it would! So, camera, on your mark... get set... GO!**

### What can you win?




**Grand Prize**
Dremel 3D Idea Builder 3D40 (list price €1,449)

**First Prize**
PicoScope 2208M MSO, 100 MHz (list price €1,099)

**Second Prize**
LabNation SmartScope Maker Kit (list price €300)

**Third Prize**
Andonstar USB Digital Microscope ADSM201 (list price €200)

**Extra Prizes**

- Sunfounder Smart Video Card for Raspberry Pi
- Raspberry Pi 3 Starter Kit Deluxe
- Andonstar USB Digital Microscope V160
- Raspberry Pi Camera Module V2



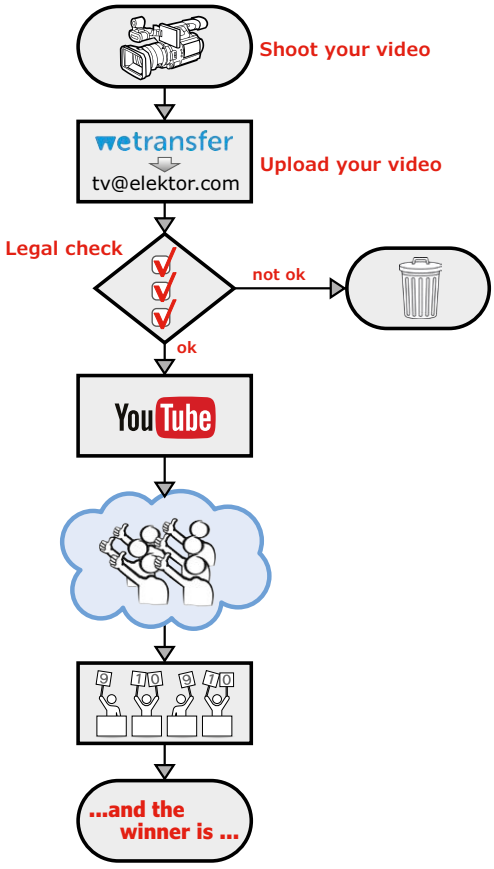




## Shoot a few minutes of e-DIY video and become an Elektor TV legend!

### Quick Start Guide

- Shoot and edit your video to enter the contest.
- Using WeTransfer only, send it to tv@elektor.com before November 15, 2017, 6 pm CET. Do **not** send entries by email.
- If your video meets the Terms and Conditions, it will be published on Elektor TV (www.elektor.tv) for everyone to see and like.
- The 15 entries with the most "Likes" will be judged by a jury to determine the Grand Prize winner as well as the winners of the first, second, third and extra prizes.
- Every competitor whose video is approved will be rewarded with a copy of the book 'Analogue Video' by Angelo La Spina.




**www.elektor.com/video-contest**

eLABS PROJECT

# Weather Display



## Visualize current weather data on an LC display



By **Markus Hirsch** (Germany)

Good weather data is more extensive than what is displayed by low-cost desktop devices that show the air temperature, air pressure and relative humidity on a monochrome LCD along with the time. A modern microcontroller with suitable peripherals can now quickly fetch the relevant and current data from the Internet and present the results in color, along with a realistic weather forecast.

**PROJECT INFORMATION**

Weather
Display WLAN
Home & Garden

entry level
➡ intermediate level
expert level

5 hours approx.

FTDI-USB / serial cable
Software downloads
(see text)

€50 / £50 / $55 approx.

Thanks to the Internet and advanced microcontrollers, it has never been easier to obtain the exact time of day, weather forecasts and a lot of other interesting information with your own electronics projects. This information can also be effectively processed for presentation on a low-cost color display. If you want to know the current weather and always have an up-to-date weather forecast, you can either hang a tablet computer with a suitable app on the wall or build the necessary hardware yourself. The latter option is clearly much more interesting, especially because it is not especially complicated with modern technology. The modular approach is the fastest way to the desired result. This means you get your hands on a suitable microcontroller board and a matching color display module, and for the rest of the necessary circuitry you build a shield or some other sort of add-on board in the appropriate format. Then you link these parts together and add the right firmware to the mix. That's exactly the recipe described below.

### Weather data and more

There is a lot of data available that can be visualized on a display. In the current version of this project, it consists of weather conditions and a five-day forecast. The aim is to display all weather data, including temperatures, air pressure, wind direction and speed, using icons for some of this data. Some useful extras are the calendar week, zodiac sign and moon phase. The latter two items are not normally provided by weather services, so the moon phase and zodiac sign have to be calculated directly on the microcontroller board. It is also helpful if the display can additionally show technical data such as the SSID of the WLAN, the received signal strength,

and the relevant IP address. The user interface is basically simple: a button press opens a small menu, a short button press scrolls to the next menu item, and a longer button press selects the current menu item. The WLAN connection to a nearby hotspot can be configured conveniently with a smartphone or other device. Before plunging into the hardware details, you can see what the end result looks like in **Figure 1**.

## Hardware

Now that we have defined what the overall system has to do, we have to consider which hardware is suitable for this purpose. Obviously the computation

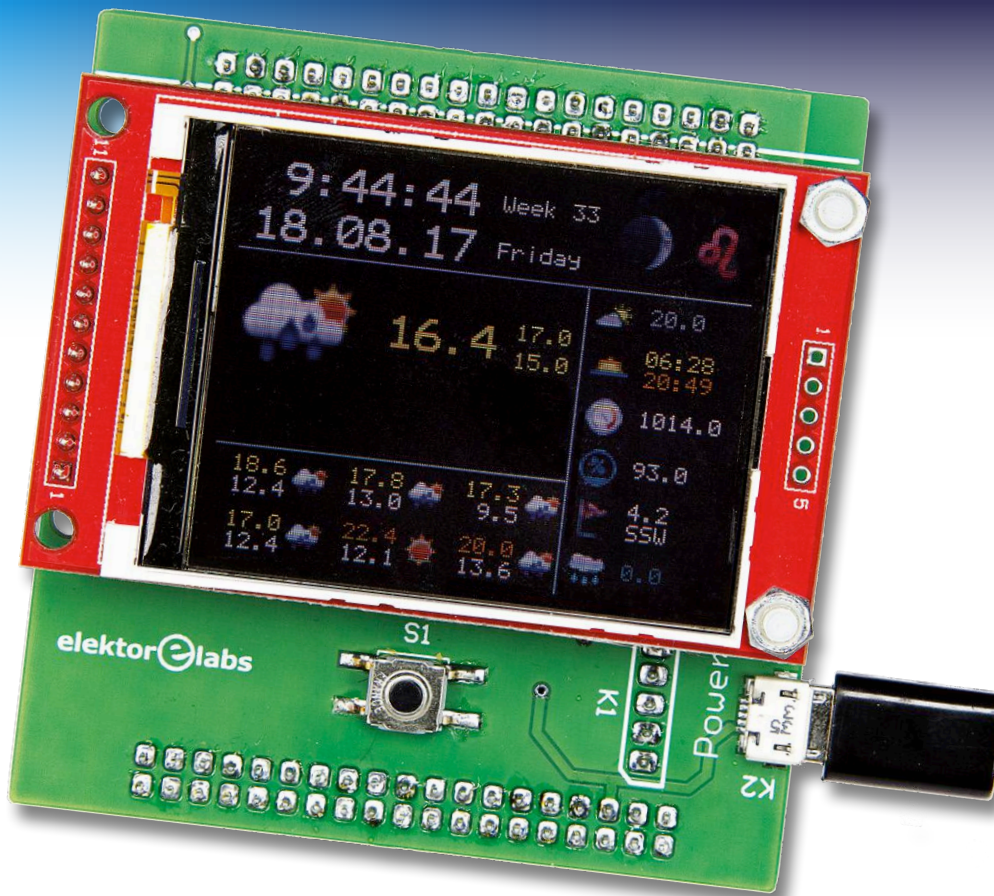



Figure 1. The prototype weather display with a screen full of weather and forecast data.

Graphic: shutterstock.com

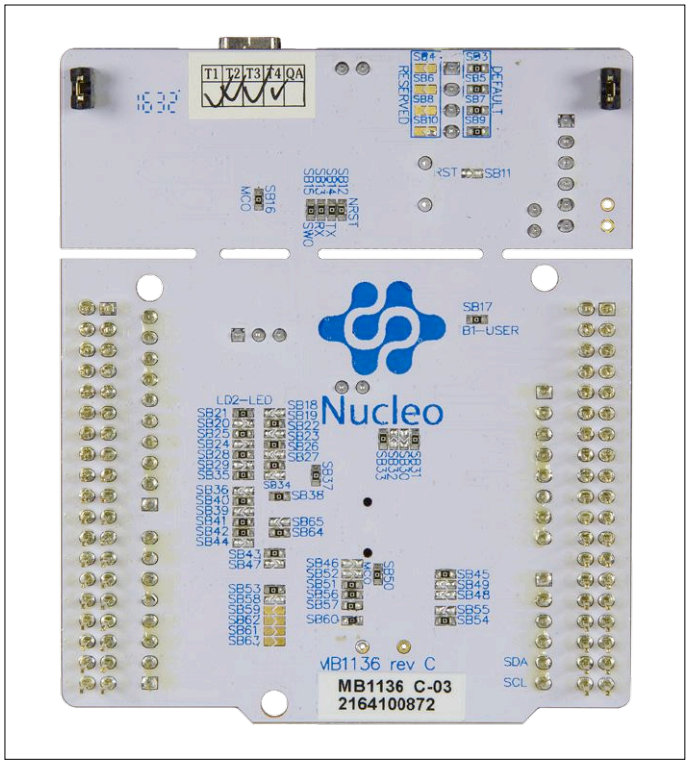



Figure 2. Bottom side of the Nucleo board. Here you can see the locations of the solder bridges and other components.

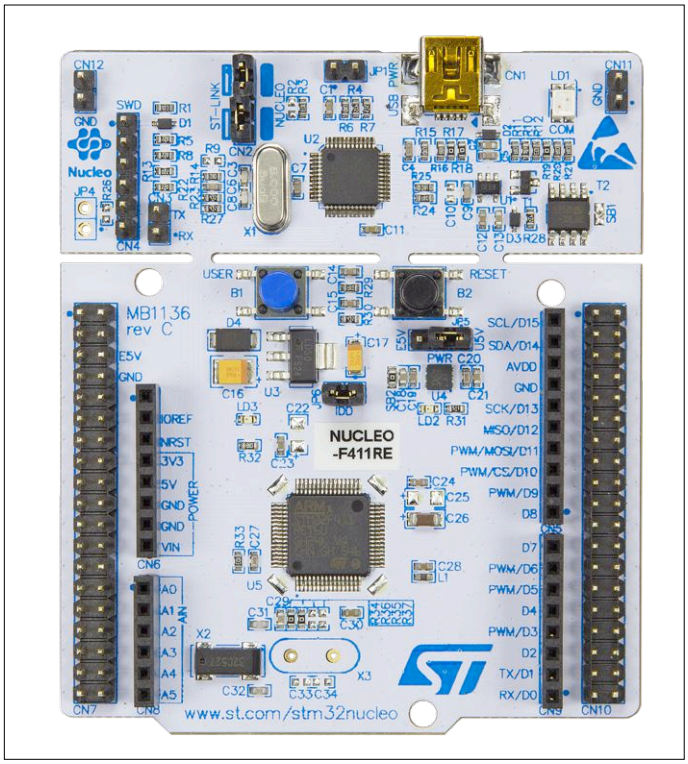



Figure 3. Top side of the Nucleo board. The upper part of the board is the ST Link section, which should be separated. The compatibility with the Arduino platform can be seen from the headers.

tasks for this sort of project can be handled by many current microcontrollers and perhaps even more ready-made boards. Fetching data from the Internet, processing it appropriately and displaying the results are straightforward tasks with the right firmware. On the other hand, embedded boards are now so cheap that it doesn't hurt to choose one with a bit more processing power than necessary, since the spare capacity can easily be used for your own extensions, or even entirely different projects, without creating resource issues. For this reason, the author chose a high-performance board from ST.

From the range of available boards, the author opted for the **Nucleo F411RE**.

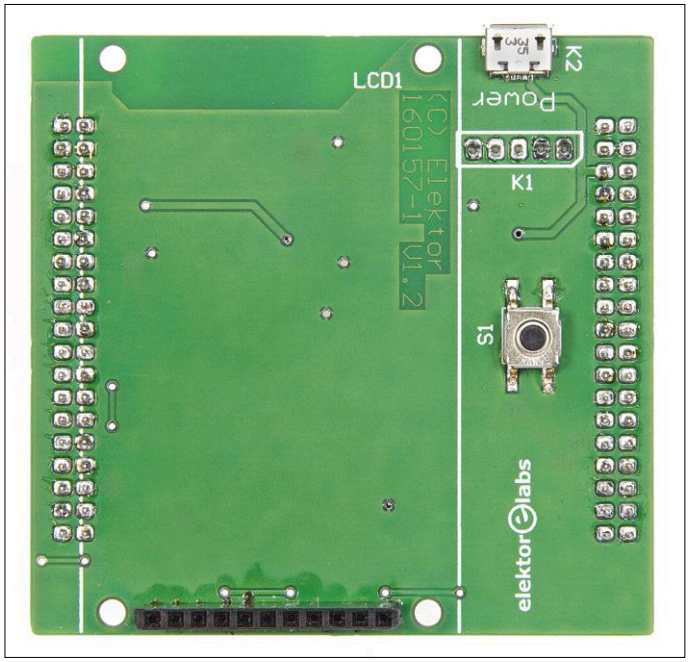



Figure 4. Except for the pushbutton and the USB connector, the display side of the shield is nearly empty.

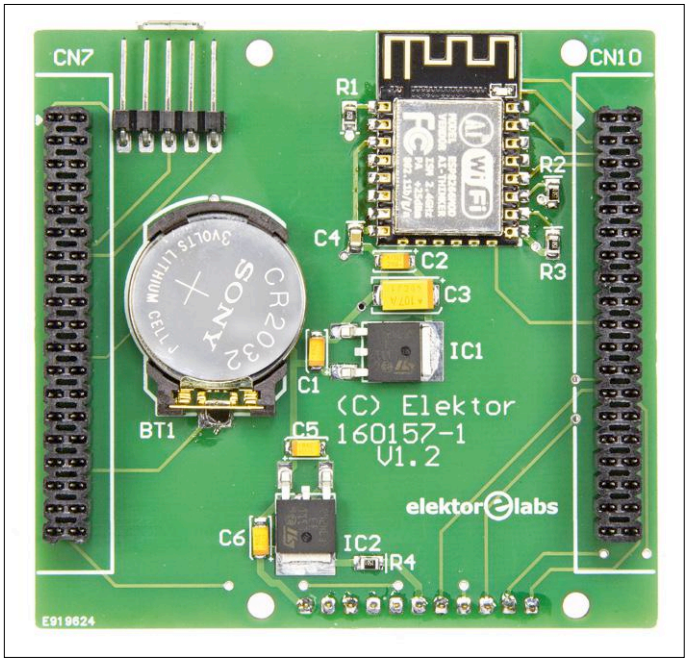



Figure 5. On the Nucleo side of the shield forming a bridge between the Nucleo board and the display, you can see the button cell, two voltage regulators and the WLAN module.

Along with a powerful STM32F411RE microcontroller, this development board comes with a debugger/programmer (ST Link), and the I/O pins of the MCU are available on suitable headers. An attractive feature of this board is that the connectors are compatible with the Arduino Uno layout. A wealth of information about the board is available on the ST website [1]. **Figure 2** shows the bottom of the board, and **Figure 3** the top.

The display is a ready-made module with a 2.2-inch (diagonal), 220 x 176 pixel screen based on the **ILI9225** display controller. If you search for this IC on eBay, you will find a multitude of modules with prices from 5 to 10 euros (or similar in pounds or dollars). The main advantages of this sort of module are that it can be driven in serial mode, its signal levels are compatible with 5 V and 3.3 V, and it even has a suitable voltage regulator on board.

The display module is not mechanically or electrically compatible with the Arduino or the Nucleo. The conventional way to resolve this problem is to design a simple **shield PCB** that serves as a bridge between the microcontroller board and the display, and which can also hold other necessary hardware in addition to the display. **Figure 4** shows the nearly empty side of the shield facing the display. All you see there is a micro USB connector for the power supply and a pushbutton for the user interface. **Figure 5** shows the side that plugs onto the Nucleo board. Along with some resistors and capacitors, the most noteworthy feature is a button cell. It is not intended for battery-powered operation, but instead as a backup power source for the integrated real-time clock on the Nucleo board in the event of a power outage. Aside from the two 3.3-V voltage regulators, the only other thing you see is a small breakout board with a serpentine PCB track. That looks like an RF component, and in fact it is an ESP8266 WLAN module, which has previously been used in a fair number of Elektor projects. The four holes in the PCB are for screw mounting the display module.

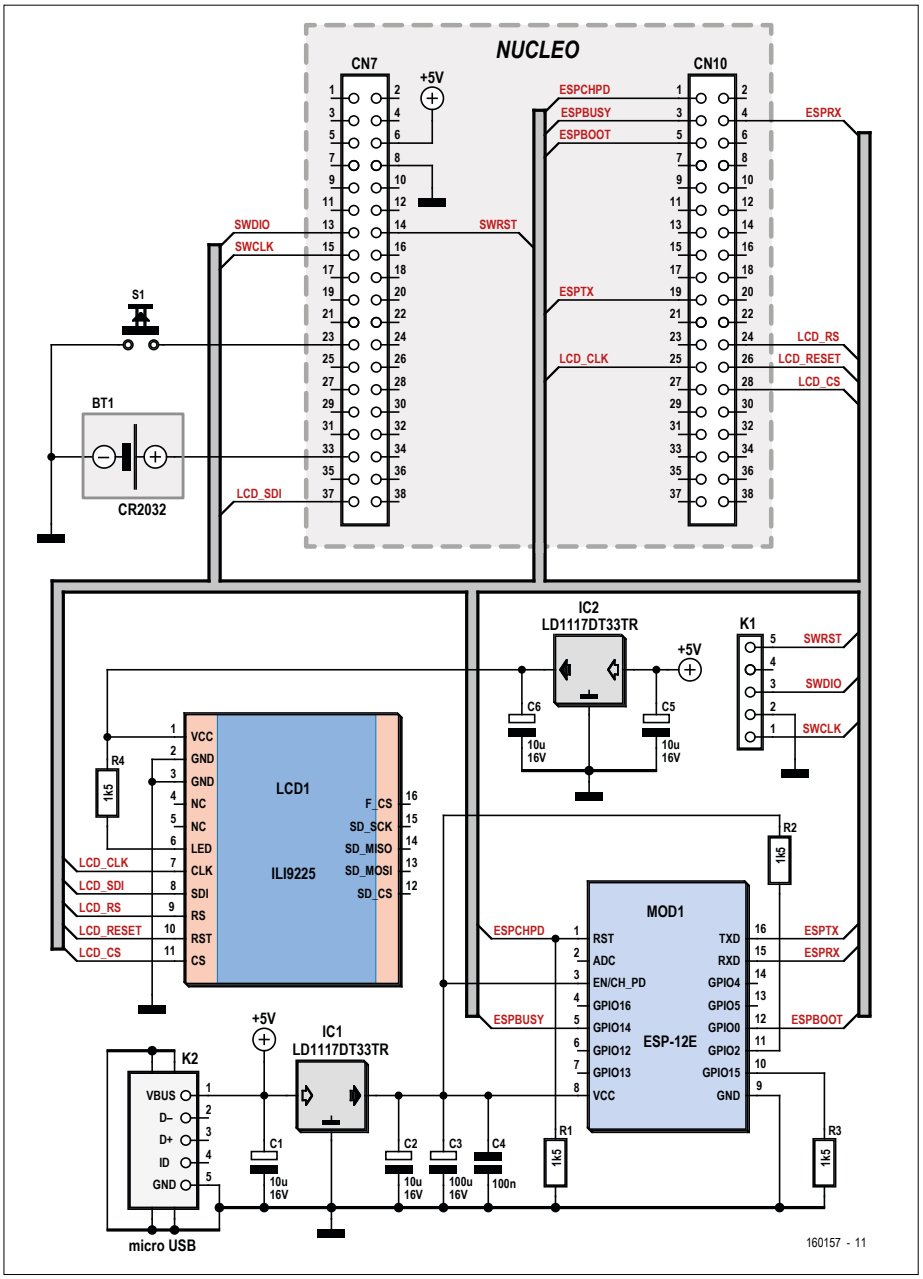



Figure 6. Circuit diagram of the shield. The component density on the board is low.

## Shield

Along with the holding the necessary headers on both sides and some electronic components, the main role of the shield is to provide mechanical support for the display and a base for the pushbutton. The circuitry shown in **Figure 6** is correspondingly simple — something this shield has in common with many other shields. It essentially consists of two voltage regulators with associated block capacitors (for separate power supply to the display and the WLAN module) and four resistors, a USB connector (with no data line connections in the current version), a pushbutton, a button cell, the display, the WLAN module, and the two headers for plugging onto the Nucleo board. Connector K1 is an additional small header for a serial interface, which is used for programming with the ST Link programming interface included with the Nucleo. By the way, S1 is connected through the header in parallel with the blue button B1 on the Nucleo board

The WLAN module is connected to the Nucleo board through a serial interface (including control lines), which makes it possible to program the module (i.e. its integrated microcontroller) via the Nucleo board. We will do that with the Arduino IDE, as described later in this article. The simplicity of the circuitry results from the use of the Nucleo board, which already has a wealth of functions and components.

It's all so simple that it is scarcely necessary to describe the board layout

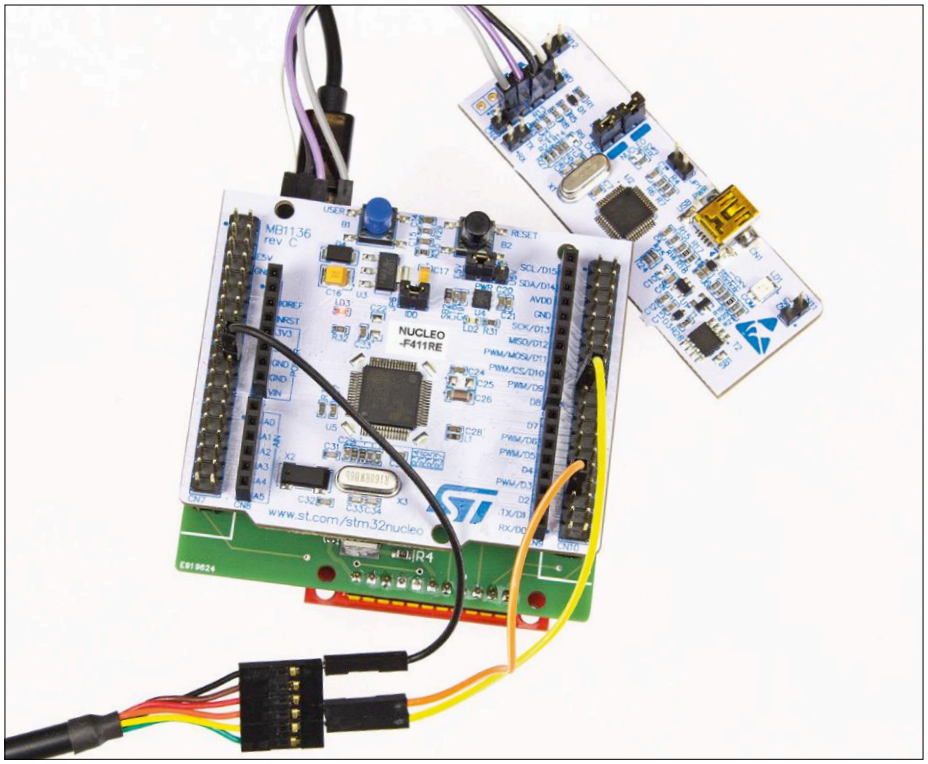



Figure 7. Here you can see how to connect the ST Link board to the Nucleo board (upper part) and how to connect an FDTI USB/serial cable for programming the WLAN module (lower part).

and assembly – provided that you have some prior experience with soldering SMDs. However, you don't need to be afraid of the SMDs used here, since nearly all of them are in solder-friendly 0805 packages. We should however point out that you should not try to use a 10-watt miniature soldering iron to heat the solder pads for the metal tabs of IC2 and IC3. The lower limit is 30 watts — more is better.

**Table 1. Modifications on the Nucleo board.**

| Component | Action | Function |
|---|---|---|
| X3 | Solder on 16 MHz crystal | X3 |
| C33 | Solder on 22 pF capacitor | X3 |
| C34 | Solder on 22 pF capacitor | X3 |
| SB54 | Open | X3 |
| SB55 | Open | X3 |
| R35 | Closed | X3 |
| R37 | Closed | X3 |
| SB16 | Open | MCO |
| SB45 | Open | VBAT |
| SB50 | Open | MCO |
| SB62 | Closed | UART |
| SB63 | Closed | UART |
| JP5 | Set to E5V | Voltage |

## Development environment

The source code was written in C++. Both commercial and free development environments are available for STM32 MCUs. For this project the author selected TrueSTUDIO from Atollic [2]. This IDE is based on the well-known Eclipse environment. Although TrueSTUDIO is basically a commercial environment, there is also a free version with no code size restriction. Among other things, the versions differ with regard to some debug features that are probably only important for professional users.

TrueSTUDIO is tailored to development with ARM processors and provides a practical preconfiguration function, which in the original Eclipse version first had to be set up manually (and in addition some components still have to be installed retroactively). With TrueSTUDIO the connection to ST Link takes just a few clicks. The environment also includes a compiler. HAL libraries and many example project are available for many processors. For hobby use, that makes the free version of this development environment a simple and easily installed, but nevertheless high-performance solution for software development with the Nucleo board.

## Nucleo configuration

Some small modifications to the Nucleo board are necessary before using it in this project. They are described in the following nine steps. For this it may be helpful to download the user manual [3] in the form of a PDF file.

*Note: If you are programming the Nucleo board for the first time, you can skip steps 1 and 3.*

1. Separate the ST Link part from the main Nucleo board (using a Dremel tool or a saw). Check that all PCB tracks have been cleanly parted and no shorts are present on the edges of both boards.
2. Carefully make the modifications to the hardware of the Nucleo board listed in **Table 1**. This includes closing or opening solder bridges and mounting a crystal and two capacitors. Tip: For closed solder bridges, you can reuse the 0-Ω resistors in 0603 format removed to open solder bridges at other places on the board.
3. Make the four connections between the ST Link board and the Nucleo board listed in **Table 2**. They are shown in the upper part of **Figure 7**.
4. Connect a 5 V power supply to CN7-6 (+5 V) and CN7-8 (GND). If you have not yet separated the ST Link board from the Nucleo board, you can leave JP5 in the default U5V position. In that case the Nucleo

**Table 2 Connections between ST Link and Nucleo.**

| Designation | ST Link pin | Nucleo pin | Shield |
|---|---|---|---|
| SW CLK | CN4-2 | CN7-15 | K1-1 |
| GND | CN4-3 | CN7-8 | K1-2 |
| SW DIO | CN4-4 | CN7-13 | K1-3 |
| SW Reset | CN4-5 | CN7-14 | K1-5 |

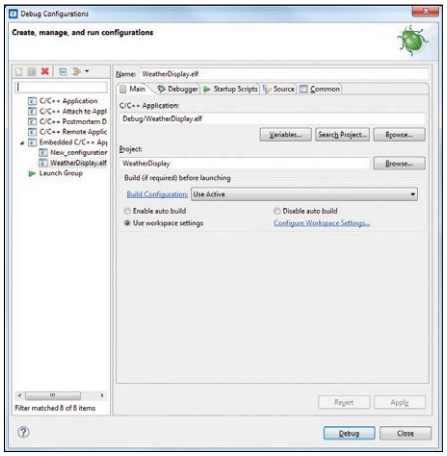


Figure 8. Nucleo configuration: the Main tab.

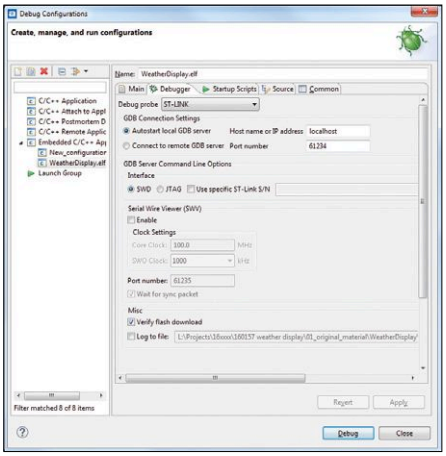


Figure 9. Nucleo configuration: the Debugger tab.

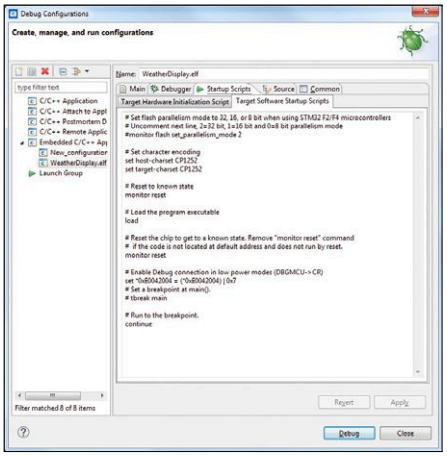


Figure 10. Nucleo configuration: the Startup Scripts tab.

board will be powered from the ST Link board via USB.

5. Download and install the evaluation version of Atollic TrueSTUDIO [2], and download the software necessary for the weather display.
6. Launch Atollic TrueSTUDIO and import the WeatherDisplay project: *File* → *Import* → *General* → *Existing Projects into Workspace.*
7. Configuration: go to *Run* → *Debug Configurations* and double-click on *Embedded C/C++ Application*. Then copy the configurations shown in the screenshots for the *Main* tab (**Figure 8**), the *Debugger* tab (**Figure 9**) and the *Startup Scripts* tab (**Figure 10**).
8. Connect the ST Link board via USB to your computer if you have not already done so. If you are using an external power supply instead of USB to power the Nucleo board, switch it on (see step 4).
9. Now program the Nucleo board by selecting *Debug Configuration* on the *Run* menu.

### ESP8266 programming

The ESP8266 module is provided here for a WLAN connection to the Internet and must first be programmed for this purpose. We do this by writing the source code for this module in the Arduino IDE and then programming the module with the resulting machine code. You will also need one of the well-known FDTI USB/UART interface cables for this. With the Nucleo acting as a programming interface with 5 V tolerant I/O ports, you are not compelled us use a cable with 3.3 V signal levels.

For programming, plug the assembled shield with the display (after attaching the display to the board with suitable screws) onto the Nucleo board. The CN7 and CN10 headers of the two modules mate perfectly. To power the system through connector K2, use a conventional 5 V USB AC adapter and a cable with a micro USB plug. Note that jumper JP5 on the Nucleo board must be in the “E5V” position for this.

Next, make the connections listed in **Table 3** (see also the lower part of **Figure 7**).

Now you have to adapt the Arduino sketch to your own geographic location and your own APPID, so that the weather forecast data will match your location. You can obtain a free APPID from *openweathermap.org* by registering at [4]. The locations list is available at [5]. Then edit the following lines in the Arduino sketch *WeatherTimeget_el.ino*:

```
Const String APIID = "your_APPID";
Const String LOCATION = "your_location"; // Example: "london"
```

These are the first two declarations at the start of the sketch.

Now is a good time to enter the name (SSID) and password of your WLAN in the Arduino sketch. These parameters can also be modified later in the Nucleo menu, but it is more convenient to enter them directly in the sketch at this point.

**Table 3. FDTI cable connections to Nucleo.**

| FDTI cable | Nucleo board |
|---|---|
| GND | CN7-20 |
| TxD | CN7-31 (PB3) |
| RxD | CN10-17 (PB6) |

See your local weather in color 24/7

In the sketch, go to the *Setup* function and replace the line

```
WiFi.begin ()

by

char cssid[] = "YOUR SSID";
char cpasswd[] = "YOUR PASSWORD";
WiFi.Begin(cssid, cpasswd);
```

Of course, you should enter the SSID and password of your own WLAN here.

If you have never previously used the Arduino IDE to program the ESP8266 module, you have to modify the IDE settings under *File* → *Preferences* as shown in **Figure 11**.

Then use the Board Manager to add the ESP module via *Tools* → *Board ...* (see **Figure 12**). More information about the ESP8266 library is available at [6].

Now select *Generic ESP Module* under *Tools* → *Board.* As previously mentioned,

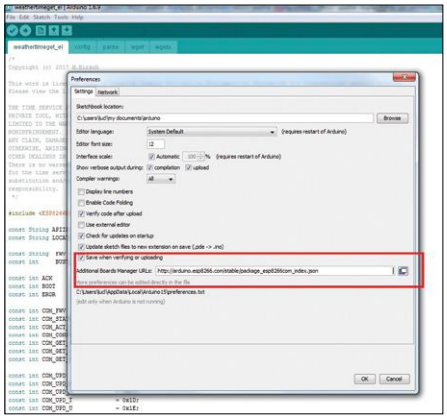


Figure 11. The Arduino IDE configuration for the ESP8266.

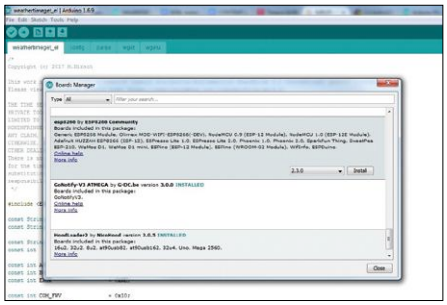


Figure 12. Adding the ESP8266 module in the Arduino IDE.

the Nucleo acts as a programming interface for the ESP module. For this you have to set the baud rate to 9600 under *Tools → Upload Speed*. Higher speeds can lead to errors. It goes without saying that you should also select the right COM port for programming under *Tools → Ports*.
Now the Arduino IDE is ready to put the Nucleo board in programming mode for the ESP8266. Press S1 below the display (or the blue button on the Nucleo board) to open the Nucleo main menu. Scroll down with short button presses until you reach the entry *ESP menu*. Then press somewhat longer to select this option. Next, select *Program loop serial* (see **Figure 13**).
Now everything is ready for programming the WLAN module with the appropriate sketch from the Arduino IDE. To start the programming process, select *Sketch → Upload*. It may take a while for this operation to complete. It is done when the status is shown as 100%. Following this you can exit Nucleo programming mode by briefly pressing S1 twice at *Exit*, by simply resetting the Nucleo board with the black button, or by brutally switching everything off and back on again.




Figure 13. Configuring the Nucleo as a programming interface for the ESP8266 module.

## COMPONENT LIST

**Resistors**
R1–R4 = 1.5kΩ, SMD0805

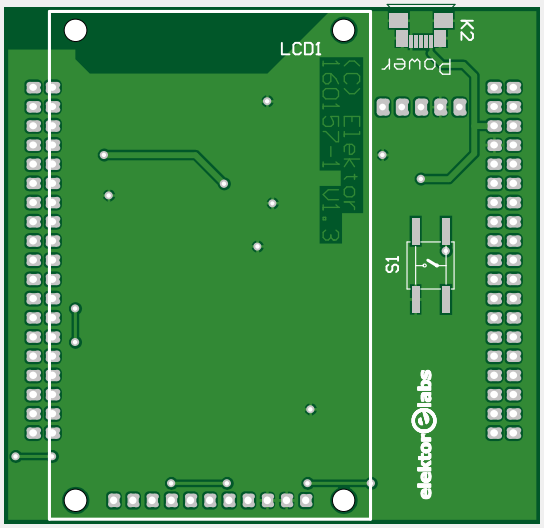
**Capacitors**
C1,C2,C5,C6 = 10µF 16V, SMD1206
C3 = 100µF 16V, SMD2312
C4 = 100nF 50V, X7R, SMD0805

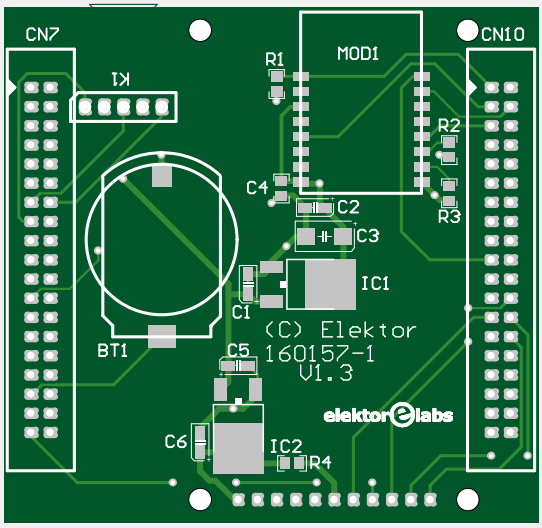
**Semiconductors**
IC1,IC2 = LD1117DT33, LDO, 3.3V, 800mA

**Miscellaneous**
LCD1 = 2.2" TFT module, 220x176 pixel, with ILI9225 (eBay) *
MOD1 = ESP-12E (ESP8266 WLAN module) *
CN7,CN10 = 38-pin (2x19) stacking header, vertical
K1 = 5-pin (1x5) pinheader, right angle
K2 = micro USB type B female, PCB mount, horizontal
S1 = pushbutton, SPDT, 6x6mm pitch
BT1 = CR2032 lithium button cell, 3V, with 20mm battery holder
PCB 160157-1 V1.3 (Elektor Store)
ST Nucleo F411RE board

**Components for Nucleo modification ***
C33,C34 = 22pF 50V, SMD 0603
X3 = 16MHz crystal for 18pF load

* See text



| Table 4. Status bits. | | |
|---|---|---|
| **Bit no.** | **Value = 0** | **Value = 1** |
| 0 | WLAN not connected | WLAN connected |
| 1 | Error in weather data update | Weather data update successful |
| 2 | Error in forecast update | Forecast update successful |
| 3 | Error in time data update | Time data update successful |
| 4 | No UV data | UV data loaded |

That completes the configuration and programming of the weather display. After a short time, the LCD should show the weather forecast for your location as entered in the Arduino sketch.

### WLAN configuration

If you did not previously enter your WLAN login data in the Arduino sketch or you want to change it afterwards, you can use the menu of the Nucleo board to enter or edit the data. To do so, go to the Nucleo main menu, select *Config WiFi*, and confirm with *YES* in the next window (see **Figure 14**).

Now you can use a smartphone or a tablet to establish a connection to the WLAN generated by the ESP8266. Bear in mind that this WLAN is only active when you are in *Config WiFi* mode or the weather display is not able to establish a connection to your own WLAN. The SSID is *WeatherNet* and the password is *WeatherPass*. Both terms can be changed in the Arduino sketch.

To configure the module, log in to this WLAN and open the ESP8266 page at IP address 10.0.0.1. There you will see a list of available wireless networks. Select the SSID that you want to use and enter the associated password. Then click on *Save* to save the configuration. After a few seconds, the WeatherNet WLAN is switched off and the Nucleo automatically returns to its main menu. If you now select *Exit*, the WLAN configuration process will be closed and the weather forecast will be shown again. The login data entered in the Arduino sketch is used after a restart.

To check the WLAN settings, select the *Status* entry on the menu. Then you will see the Nucleo and ESP firmware versions displayed at the top of the screen, followed by the IP address of the ESP, the SSID of the WLAN, and the WLAN signal strength. Of course, the status of the network connection is also shown. A value of 0x0F is good news because it means that the system is connected to the WLAN and all data has been loaded. The meanings of the individual status bits are listed in **Table 4**.

### Miscellaneous

The routines for fetching UV data are unfortunately not included in the weather display firmware because this data is not available for free. If you happen to find a free source of UV data, the author would be very pleased to hear about it.

As mentioned at the start of this article, the STM32 is by no means fully loaded with the tasks described here. This means that there is plenty of opportunity to go hunting for other interesting data, such as ozone concentration etc., and modify the firmware so that this data can also be downloaded from the Web and displayed. Other conceivable options include several alternating views, animated weather icons, a feed reader, email notifications, or a WLAN boot loader. As usual, the Nucleo firmware for this project (along with the Arduino sketch for the WLAN module) is available on the Elektor web page for this article [7] for free download. We hope you enjoy this project! ⏮

(160157-I)




Figure 14. Configuring the WLAN settings via the Nucleo menu.

### Web Links


[1] ST Nucleo F411RE board: ww.st.com/en/evaluation-tools/nucleo-f411re.html

[2] Atollic: https://atollic.com/resources/download/

[3] Nucleo manual: www.st.com/resource/en/user_manual/dm00105823.pdf

[4] APPID: https://openweathermap.org/appid

[5] Locations list: http://openweathermap.org/help/city_list.txt

[6] ESP8266 library info: https://esp8266.github.io/Arduino/versions/2.3.0/

[7] Downloads: www.elektormagazine.com/160157



**About the Author**

Markus Hirsch originally studied industrial engineering but ended up in R&D due to his many years as an electronics hobbyist. After spending several years in quality management, he now works as a hardware and software developer in a mid-size industrial firm. He also enjoys developing hardware and software projects in his spare time. In addition, he is active in 3D printing, app and Web programming, and other projects. Although these projects have practical or educational uses, the main consideration is that they are fun. Just like the project presented here.

# Spy Camera Detector

## Reflections give away the position

By **Jörg Trautmann** (Germany)

Although some people don't seem to mind revealing their innermost thoughts online, we know from recent revelations about the NSA activities that we can also be victims of unwelcome surveillance. Help is at hand!





in memoriam: Antonio Prohias

As electronic equipment is getting ever smaller, tracking down devices such as hidden cameras becomes almost impossible. These spy cameras are freely available from online auction sites and are available hidden inside lighters, pens, sugar cubes etc. With this project you can build a device for detecting such hidden cameras using just a handful of standard components and a little craftwork.

### Red, red, red... all the spies are red

All surveillance and spy cameras have one thing in common: a lens, which must be directed into a room to make recordings. How is it then possible to detect a small hidden or sometimes camouflaged camera lens? Well it really is simple: the detector shown in the **title photo** uses LEDs to emit pulses of red light which then get reflected by the spy camera lens. To see these reflected glints the user records the scene using a camera in the detector unit with a red filter (the red filter helps reduce the Purkinje effect [1], also the reason why submarines use red lights inside). The glint of red light bouncing back from the lens can now be seen. The LEDs are flashed so it's easy to spot the reflection when scanning the room. A similar system is used to locate astronomical comets by superimposing and comparing two photos of the same area of sky.

The two most important things to consider for a successful spy camera detector are firstly the LEDs which should be

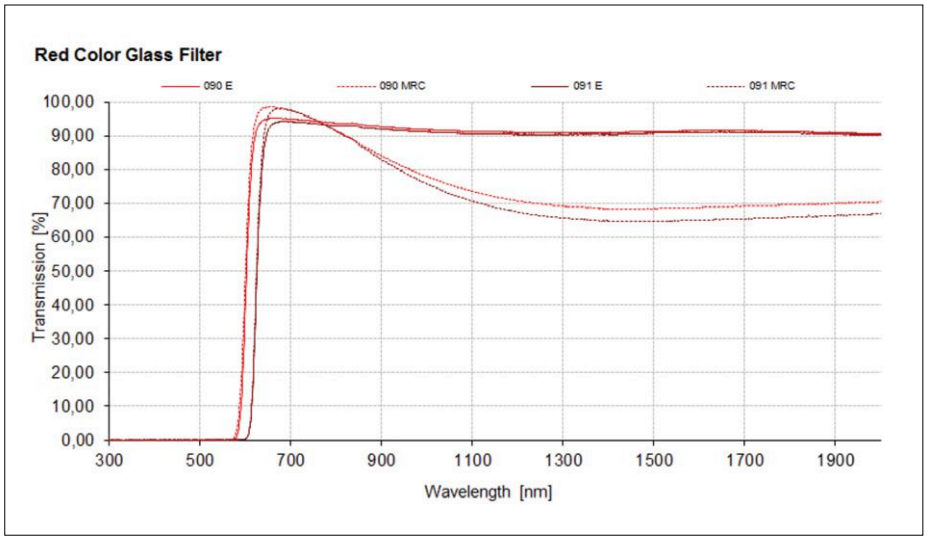



Figure 1. Photographic red filter frequency response (picture: Jos. Schneider Optische Werke).

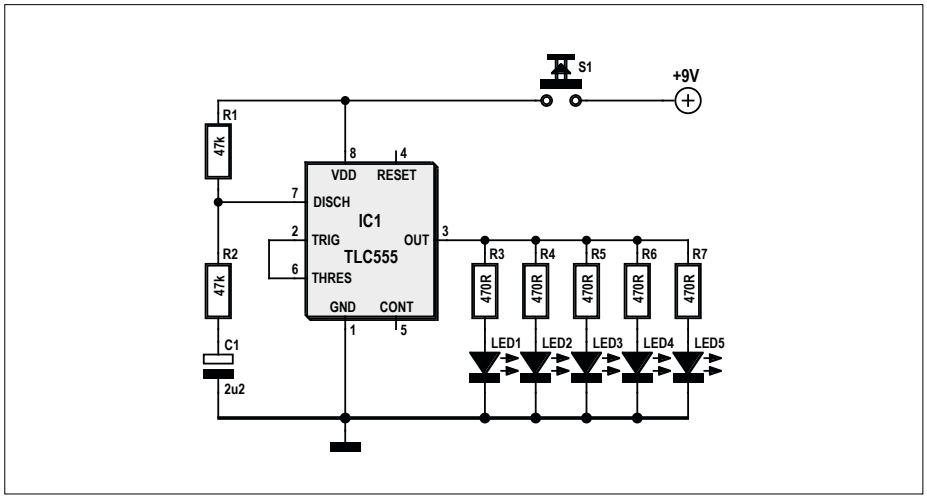



Figure 2. The circuit: a 555 plus a few components.

ultra-bright types and secondly the filter color. The brighter the LEDs the more likely you are to detect the glint from the spy camera. The color pass band of the transparent red or red/orange filter should correspond closely with the peak wavelength of the LED light. When there is a mismatch in their properties there will be little or no chance of detecting the spy camera.

When the detector is used together with an SLR-type camera, a red filter can be fitted to the filter mount on the front of the lens. The well-known manufacturer of photo filters, Schneider Optics (B + W filter), offers two different red filters (IF090 and IF091) with two different coatings (E and MRC). **Figure 1** shows that the transmission characteristics of these red filters are surprisingly similar to the frequency response of a low-pass filter made with electrical components. The corner wavelengths are 600 nm and 625 nm, respectively, showing a transmission percentage of less than 1% for higher frequency light waves! In the filter's pass band the transmission is nearly constant for the filter with the E-coating, whereas with the MRC-coating it decreases by about 30%. There are also specialist narrowband band pass filters but these are normally not offered through conventional hobby photo outlets (and are also not so well suited, because LED color is not constant, but slightly influenced by the LED's operating current).

This brings us to the second component in the circuit: the red LEDs. Looking at the transmission characteristic of the filter its peak response should of course coincide with the LED's transmitting wavelength, the closer to the cutoff frequency the better. The color must of course also lie in the visible part of the spectrum. When choosing an LED don't pay too much attention to the maximum brightness value given in the data sheet, the LEDs will be used here with a current of less than 20 mA. The most important parameters are the LED color, forward current (depends on forward conduction voltage), brightness, package outline. You can enter these parameters into a component selection filter available at many online component distributors and get a list of suitable LEDs.

### A touch of electronics

At the heart of the circuit in **Figure 2** is the tried-and-tested timer module type 555, which is configured as a multivibrator. The RC values are calculated in such a way that the LEDs flash at approximately 250-ms intervals when the pushbutton is pressed. In order to achieve best possible room illumination, five ultra-bright red LEDs are used. The LED series resistors are calculated at 470 Ω which results in a total current of about 75 mA divided between the LEDs at an operating voltage of 9 volts. The TLC555 timer module used can drive up to 100 mA, so there is no need for a driver transistor.

The plastic case used here needs a large hole at the front and rear surface, the diameter of which is determined by the size of the camera lens. If you are not using an SLR with a screwed-on filter with this detector housing, you should ensure that any light leaking from the sides of the LEDs cannot enter the camera lens directly. It may be necessary to fit a black cardboard tube through the two holes in order to avoid this.

The electronics can be easily built up on a small square of perfboard, fitted inside the case and wired to the operating pushbutton S1 and the battery holder. The LEDs are fitted using reflector mountings so that the light is concentrated in a forward direction, these can then be wired to the 555 board. With the circuit wired up and a 9 V battery connected, the five LEDs should flash at quarter second intervals when the pushbutton is pressed. If everything is working as it should the search for a

hidden camera can begin. The scan can be carried out in normal, but diffuse daylight conditions, so that slight differences in brightness are more easily detected. Now you can start out on the trail for the hidden camera. Fit the spy camera detector to the camera lens (**Figure 3**), start recording in video mode and slowly scan around the whole room preferably at three different heights (for example, 60 cm, 120 cm and 180 cm). The recorded video material can then be studied more closely later and evaluated using a PC. **Figure 4** shows in two individual images how a spy camera gives its position away.



Figure 3. The detector flashes, the camera records the video evidence.

### Do's and don'ts

The system has a few limitations that you can't really do much about and some that you can. The camera detector works perfectly when the spy camera lens is spherical and also reflective. If it has a planar lens, a successful detection is only possible at certain angles between the lens and the illuminating LEDs. When the lens does not reflect, the detector will not work.
You can stick to certain guidelines to improve your chances of discovering a spy camera:

- The distance between the two cameras should be small; the smaller the distance, the better is the chance of detection. You are less likely to find it by standing in the middle of a room and turning through 360°, it's better to get up close to suspicious areas.
- Shade any direct sunlight from behind! When the spy camera has a bright background it is more difficult to see than when the background is diffusely lit.
- Shade any direct sunlight from the front! When the detecting camera has bright coming from behind it is more difficult to see the glint from the spy camera lens. Try to avoid any direct light.
- Look closely at highly reflective surfaces. The reflection from the spy camera lens can be easily missed.
- And finally the most important don't… do not stare directly into the LEDs when the detector is working!

(140459)

### Web Links


[1] https://en.wikipedia.org/wiki/Purkinje_effect

[2] www.schneiderkreuznach.com/industrial-solutions/industriefilter/produkte/filtertypen/colorfilter/

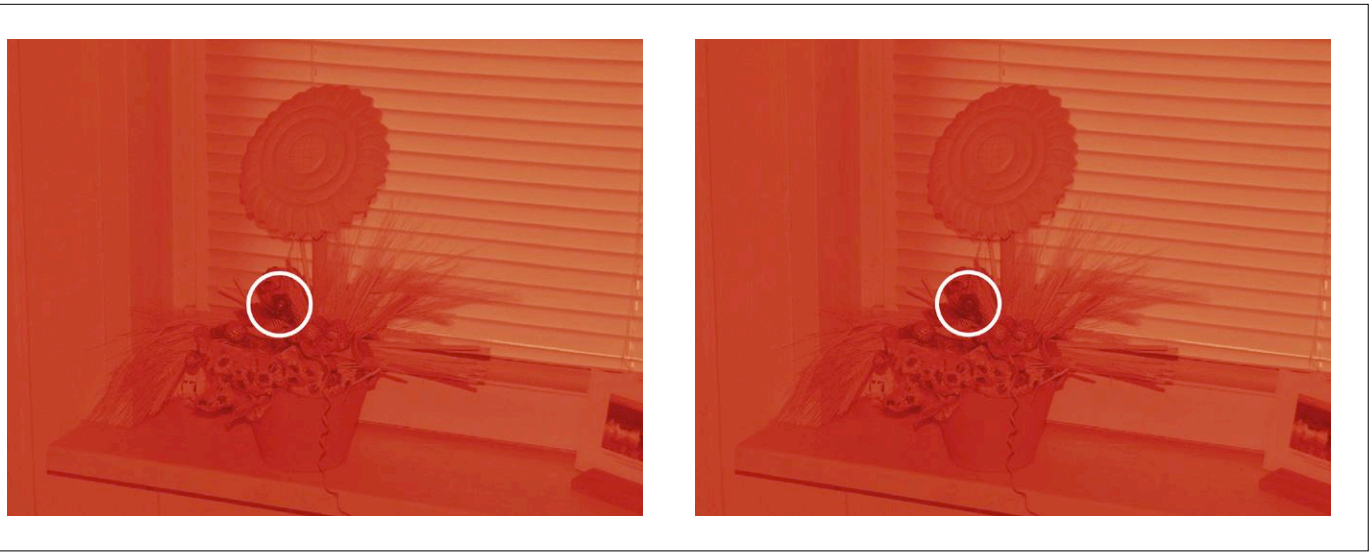



Figure 4. The small but distinct giveaway: A spy camera hidden in the plant pot is given away by a glint!

# (almost) everything you ever wanted to know about...

# connectors for video, HF and test equipment

Replies from
**Rémy Mallard** (France)

We ask less of audio connectors than we do of those for video, RF and test equipment input/output. There are certainly good reasons for this. We asked Remy Mallard to tell us more.

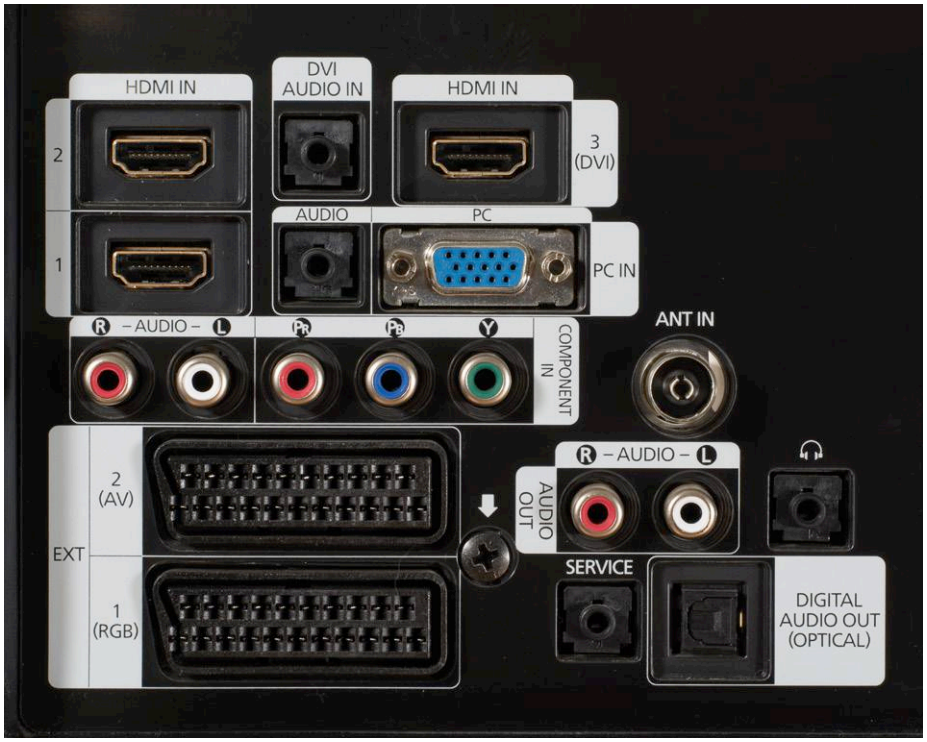



*photo : Shutterstock*

**Q** *In general terms, what can you say about video connectors?*

**A** Video has passed from black and white to color, then from analog to digital. Video connectors have also evolved, and the manufacturers will always offer rugged connectors to the pros, and 'bargain' (but quite solid all the same) connectors to the general public. In the professional world, analog video is transported over one or more coaxial cables terminated with robust 75-Ω BNC plugs: one conductor for composite (CVBS) video signals (PAL, SECAM or NTSC, luminance Y and chrominance C over the same wire), two distinct conductors for S-Video or Y/C (separate luminance and chrominance), or three conductors for separate components **R**ed **G**reen **B**lue or YUV [1] or $YP_BP_R$. Digital video may also be carried as SDI over coaxial cable with BNC plugs. For long distances correctors / equalizers are needed to compensate for high-frequency losses caused by the cables, and Ethernet links (twisted pairs and RJ45 connectors) or optical fiber may also be used (**Figure 1**).

For analog video for general consumer use, the very costly BNC connectors have been replaced by cheaper ones: RCA/Cinch (composite video on the yellow plug), SCART (French Péritel) with 21 pins, imposed on television manu-

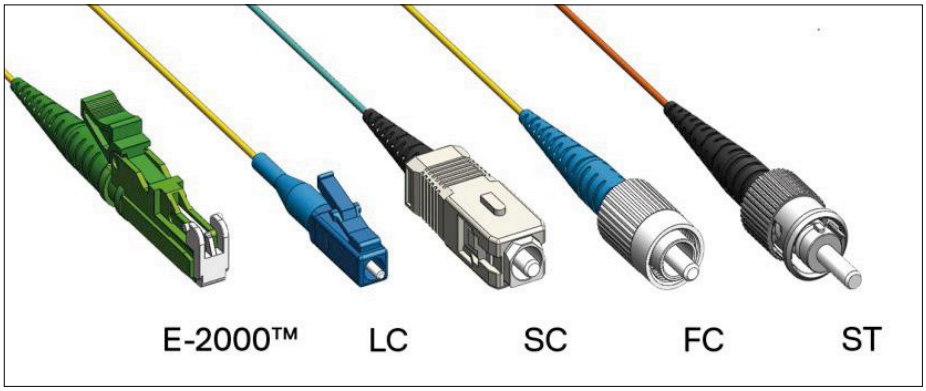



Figure 1. Fiber optic connectors. (Source: www.insys-ch.com)



Figure 2. HDMI Connector.
(Source: kabeldirekt-store.de)



Figure 3. DisplayPort Connector for digital screen interface, from VESA.
(Source: kabeldirekt-store.de)



Figure 4. Type 7/16 (7-16) female connector.

facturers from 1980 to 2015 and well known for its bad connections (composite video, Y/C and/or RGB), or even Ushiden with four pins (S-Video). Digital Video made its entry with DV cameras and their famous IEEE 1394 connector (FireWire at Apple; i.link at Sony) which allowed lossless transfer of picture and sound. Very practical…even if it only worked one way (to prohibit making digital copies, maybe? Today, DVI, HDMI (**Figure 2**) or DisplayPort (**Figure 3**) connectors are in common use, and transport digital video signals… with cables sometimes very costly, where an Ethernet cable would do as well. 'All-digital' eliminates the A/D and D/A conversion steps (**A**nalog, **D**igital) which contributes to a better image quality and a reduction of cost and complexity of equipment.

If digital video is now part of our daily lives, compatibility with legacy equipment is often desirable. That's why you still find on some equipment (TVs, game consoles, home cinema systems, camcorders and digital cameras) analog inputs and outputs on RCA plugs, 3- or 4-way jacks, or even some proprietary connector that makes users complain… but makes sellers of adapters very happy.

| Acronym | Meaning |
|---|---|
| **BNC** | Bayonet Neill–Concelman |
| **CVBS** | Chroma Video Blanking Synchro |
| **DV** | Digital Video |
| **DVI** | Digital Visual Interface<br>-A *(analog only, equivalent to VGA)*<br>-D *(digital only)*<br>-I *(analog and digital)* |
| **FC** | Ferrule Connector |
| **LC** | Lucent Connector *or* Local Connector |
| **HDMI** | High Definition Multimedia Interface |
| **NTSC** | National Television System Committee |
| **PAL** | Phase Alternating Line |
| **RCA** | Radio Corporation of America |
| **SC** | Subscriber Connector, Standard Connector *or* Siemon Connector |
| **SCART** | Syndicat des Constructeurs d'Appareils Radiorécepteurs et Téléviseurs (European TV manufacturers consortium) |
| **SDI** | Serial Digital Interface |
| **SECAM** | SÉquentiel Couleur À Mémoire (French TV standard) |
| **SMA** | SubMiniature version A |
| **ST** | Straight Tip |
| **VESA** | Video Electronics Standards Association |

**Q** *Why do RF (radio frequency) connectors differ from video connectors?*

**A** You will find BNC connectors (75 Ω) used for video and also in low-power RF (BNC 50 Ω). But where RF connectors really come into their own is working with higher frequency or high power signals.

RF connectors are generally coaxial, with a central conductor surrounded by grounded metallic screening. They allow the various parts of a radio frequency system to be interconnected (antenna, receiver, cable, transmitter, filter, etc.). Some of them are lockable (screwing for SMA or TNC, bayonet for BNC) and others are not (TV 'PAL' antenna connector, 9.52 mm diameter, which only holds by friction, like an RCA/Cinch).

BNC connectors are fine for a few watts, even a few tens of watts, but are no good for high powers. There, one would use the robust "7/16" connectors (max 3 kW/5 GHz) (**Figure 4**), N connectors (max 1 kW/10 GHz) (**Figure 5**), or even, for more modest needs, SO-239 female (**Figure 6**) and PL-259 male connectors (strangely, these last two are often just called 'UHF connectors' although they do not work well above 100 MHz). Mini-UHF, F and SMA (**Figure 7**) connectors are suitable for signals in the GHz range, but are not good for high powers. Their small size makes them ideal for compact HF couplers or distributors (distribution of TV or satellite signals in buildings, for example). For very high powers, specific large connectors are used, which will not be known in your local components shop. At hyper-frequencies (several GHz) connectors and coaxial cables give way to



Figure 5. Type N male connector.



Figure 6. Type SO-239 connector.



Figure 7. Type SMA connector.

waveguides. That's where you discover what 'deluxe plumbing' means.
It must be recognized that the world of RF is very particular, and demands a good knowledge of the laws of physics and electromagnetism. It takes very little for an HF equipment or link to work badly or not at all. My first FM transmitter, put together as if it was an audio mixer (I had no knowledge of the rules governing RF) was sadly lacking in efficiency!

**Q** *Do RF connectors need special materials?*

**A** Yes. Because at RF, any connector is a source of loss of signal transmission. Their choice, which depends on the frequency and the power of the signals being carried, is often very critical. Various types of conductor may be used for the different parts of the connector (center pin, body, intermediate rings): stainless steel, brass, gold plated brass, bronze plated brass, nickel plated brass, gold beryllium bronze, copper-beryllium, nickel, gold or silver. The insulator (the dielectric) which keeps the pin separate from the body is often made of Teflon® (PTFE), which shows very low losses at high frequencies (it is usable up to around 10 GHz). The thickness of the plating (gold for example) on the moving parts depends mainly on the number of connections foreseen by the manufacturer.

**Q** *Can you use adaptors (gender changers) at RF?*

**A** Yes, but be careful! That causes yet more losses, often considerable even for high priced connectors. In an RF link, always avoid gender changers and extensions (always use one 6-m cable rather than three 2-m cables in series). And if you're looking at an RF link of several tens of meters, it's worth looking at the losses from the coaxial cable itself, which may be appreciable!

**Q** *And what of the connectors used on test equipment?*

**A** Test equipment needs to be accurate, reliable and robust. On lab equipment, connected cables are often used. But there are different requirements for portable equipment: their connectors must withstand multiple connections and disconnections, but also small accidents like being dropped (yes, I know you're always careful!). The BNC connector is without doubt the most common connector on oscilloscopes, connecting probes with 'high impedance' screened cable or 50 Ω cables (my first oscilloscope used banana plugs, but its bandwidth was severely limited — so let's forget that). Equipment such as vector analyzers, spectrum analyzers and wattmeters often use N connectors, made for 50 Ω, and usable up to several GHz. ⏮

(160537)

## Web Link

[1] YUV color space: https://en.wikipedia.org/wiki/YUV



Figure 8. Connectors for very high frequency, but these don't handle high power.

# DIY Kitchen Scale

## Weigh up to 49.05 kg·m/s² with Arduino

Based on an idea by **Bera Somnath** (India)




The starting point for this article was a project posted on Elektor Labs describing the construction of a kitchen scale from an ATmega328 microcontroller, a load cell, a load cell interface board, an LCD, and some software. A bit of cutting and drilling of wood and metal was required to transform it all into a useable weighing scale.

### Features

- Up to 5 kgs
- OLED display
- Arduino-based

At Elektor Labs, when we decided to post-engineer this project [3] for publication in *Elektor Magazine* and started looking for the parts, we discovered that one of our regular suppliers carries a kit for exactly such a scale in their catalog. This kit contains an Arduino Uno-compatible board, a load cell, an interface board for the load cell, an LCD and laser-cut plastic parts to make a nice, transparent kitchen scale, and so we ordered one for inspection. Because the kit comes without a shield to hold all the parts — everything is supposed to be hooked up with jumper wires (included) — we decided to enhance it with an OLED display, providing not only more display options and graphical fun, but also freeing up I/O pins that may come in handy when the scale has to be interfaced to some other equipment. We added a 'Hold' button too as that handy function that is missing from so many kitchen scales. To cap it all, a trimmer was added for easy calibration of the scale.

Shall we have a look at how it all works?

### Weight or mass?

In modern science weight is defined as mass × gravitational force and as such it is a force, expressed in newton (N) or, in SI base units, $kg \cdot m \cdot s^{-2}$; the SI unit of mass is kilograms (kg; kgs). A weighing scale measures weight, not mass, hence

if it displays results in kilograms, strictly speaking it talks nonsense. But, since for most purposes the gravitational force can be considered the same everywhere on Earth (9.80665 $m \cdot s^{-2}$), the scientific community allows us to express weight in kilograms.
Because weight is a force, the weight of an object can be determined by measuring the force it exerts on another object, like a spring or a beam. A spring is compressed or stretched by an object's weight, while a beam is bent. The weight of an object can also be determined by a balance comparing it to a known reference weight.

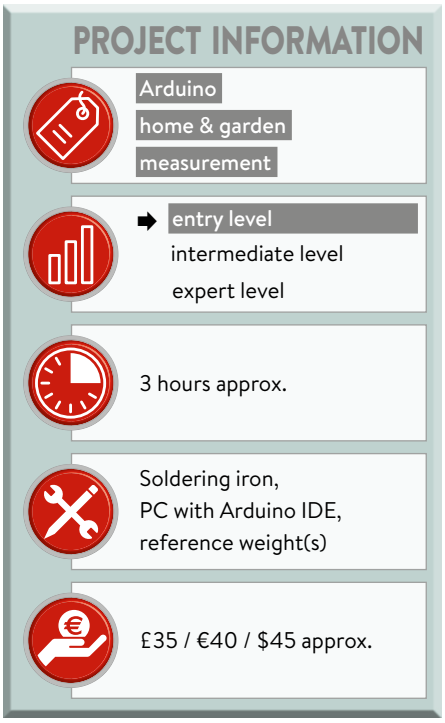
### The strain gage

Basically a kind of variable resistor, a strain gage (gauge in Br. E) is good for measuring mechanical deformations, its resistance being a function of mechanical compression or tension. The strain gage was invented in 1938 on the west coast at Caltech by Edgar E. Simmons Jr. (who liked to dress in tights, a tutu, a turban and white ballet slippers). Simultaneously the strain gage was invented at the east coast by Arthur Claude Ruge (pronounce as Roogee) at MIT. Simmons and Ruge share the original patent.
The basic strain gage consists of a long, thin folded "wire", a foil actually, printed on a flexible backing support (see **Figure 1**). When the device is stretched, the foil becomes more resistive; when it is compressed, its resistance decreases. The variation in resistance is small, and to measure it with any precision a Wheatstone bridge is the way to go. To eliminate the temperature factor it is best to use two strain gages for one leg of the bridge. When the other leg of the bridge is also made up of strain gages, the output signal level increases and we call it a full-bridge strain gage.
Strain gages are fragile devices and attaching them to the object of interest is difficult. That's why they often come mounted on some sort of easy-to-use carrier, the so-called load cell. Everything you ever wanted to know, and more, about strain gages can be found in the free book at [2].

For our kitchen scale we use a load cell shaped as parallel beams with four strain gages attached to it configured as a full bridge (**Figure 2**). One end of the beam is fixed to the scale's frame; the object to weigh is placed on the other end of the beam, effectively bending it. This unbalances the Wheatstone bridge and the resulting voltage difference appears at the load cell's output where it can be measured.
Even though we are using a bridge with four strain gages, the output signal is still very small, like a few tens of millivolts. Consequently an amplifier is required to take the signal to a level that can be digitized and processed further.

### The circuit

Now that we know how our scale works, let's have a detailed look at its schematic in **Figure 3**. Since load cells are fairly common devices it is not surprising that specialized integrated circuits exist capable of amplifying and digitizing a load cell's weak output signal. For this project the HX711 from Avia Semiconductor was chosen, a 24-bit analog-to-digital converter (ADC)



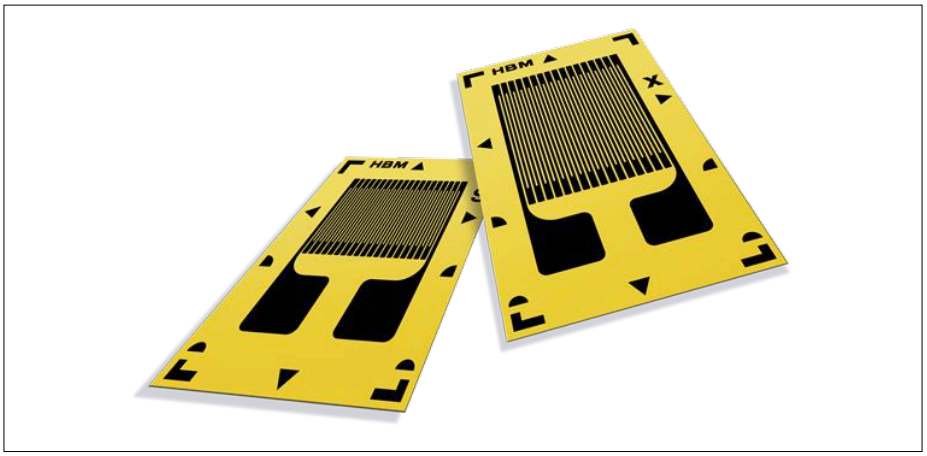




Figure 1. A strain gage is only sensitive in the longitudinal direction. (Photo: HBM)

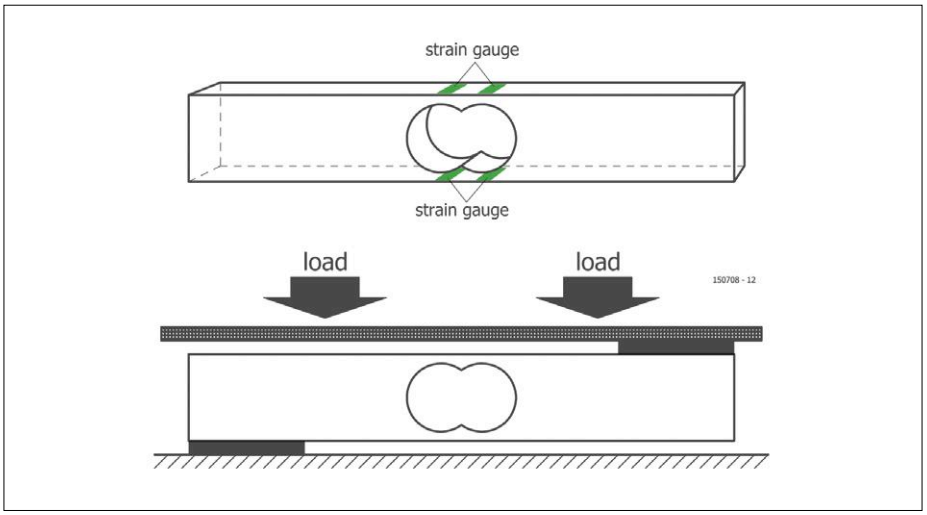



Figure 2. A parallel beam load cell carrying four strain gages in a full bridge configuration.

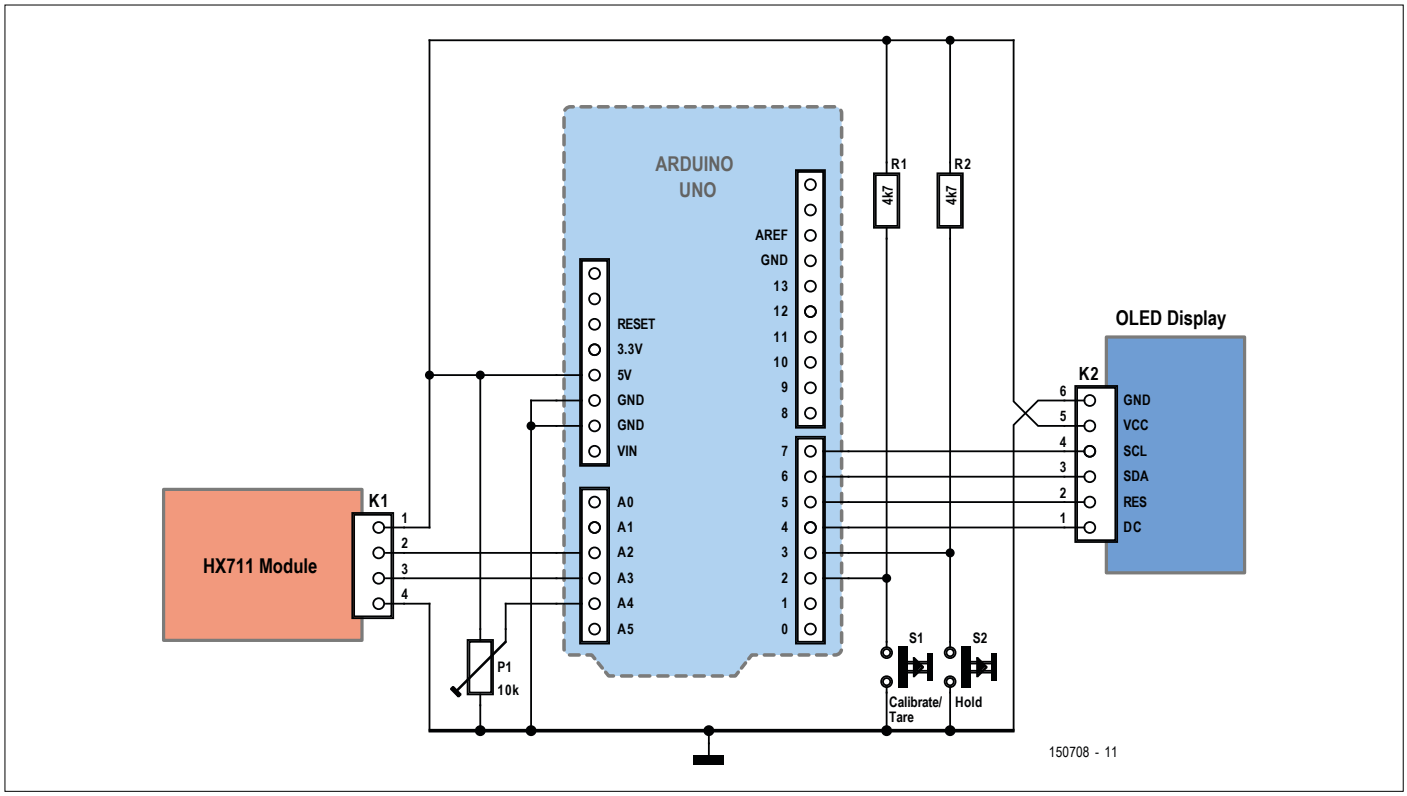



Figure 3. The schematic of the kitchen scale is very simple because all the clever stuff is done by the HX711 breakout board, the OLED display and the Arduino Uno.

for weighing scales, mainly because it is widely available on the Internet in the shape of breakout boards (it's actually quite difficult to obtain just the IC). The breakout board connects to K1. Note that this connector should be mounted floating above the PCB, otherwise the breakout board will not fit.

The output of the HX711 is a datastream (K1, pin 3) clocked (K1, pin 2) into the ATmega328 microcontroller living on an Arduino Uno board. The microcontroller also reads the position of a 15-turn trimmer for calibrating the scale.
The user interface consists of a graphic OLED display connected to K2, and two pushbuttons 'Tare' (S1) and 'Hold' (S2), pulled up by R1 and R2 respectively. K2 is mounted at an angle to improve readability.

## Software

The microcontroller is executing a rather straightforward program or sketch, since we are using Arduino. Most of the sketch is actually dedicated to switching pixels on and off in the right places on the display; reading the load cell and converting the measured values to grams (g) and ounces (oz) is done in two lines of code.

```
gram = scale.get_units(10) *
   int(factor) / 1000;
ounce = gram * 0.0352739619;
bar = gram*40/5000;
```

This is a bit misleading, because the sketch heavily leans on a library that takes care of the communication with the HX711. The measured weight returned by `scale.get_units` is a 32-bit signed integer, converted to grams with the help of the calibration factor set with P1.
The graphics are handled by the 'Universal 8bit Graphics Library' (U8glib). The display shows the measured weight

## Tare?

Relatively useless, the following information is great to have handy when a conversation stalls: […awkward silence…] *Say, do you know what 'Tare' means? You know, that pushbutton on a kitchen scale? No? Well…*" and you spill your knowledge. "*Jeez, I didn't know that, that's amazing! Imagine, all these years…*", and the conversation will run smoothly for another hour (at least). So what does 'Tare' mean? It's the weight of the container in which something is placed, the weight of an unloaded transport vehicle, used to calculate the weight of the goods inside it. Now you finally know why you would want to press the Tare button after placing an empty beaker on a scale.

**Table 1. How to wire the load cell to the HX711 breakout board. The B– and B+ inputs remain unconnected**

| Wire | Signal |
|---|---|
| Red | E+ |
| Black | E– |
| Green | A+ |
| White | A– |




Figure 4. The OLED display shows the weight in grams and ounces, together with a bar graph.

in grams (g) and ounces (oz), and as a bar graph up to 5 kgs (**Figure 4**). When the Hold button (S2) is pressed, the display freezes and the object can be removed without losing the measured weight. Pressing Hold again will return the scale to normal operation.

S1 is the Tare button (see inset), pressing it will set the displayed weight to 0 grams no matter if an object is on the scale or not.

The software for this project is available free of charge from the *Elektor Magazine* website [1].

## Calibration

Connecting the load cell to the HX711 module (see **Table 1**) completes the assembly of the scale. However, before it can be used in your next baking or cooking project (like tonight's dinner), it should be calibrated. This can be done with the sample weights included in the kit (or with other objects of which you know the exact weight):

1. Turn on the scale.
2. Press the Calibrate button (S1).
3. Place a reference weight on the scale, for example 150 grams.
4. Adjust trimmer P1 until the display shows the right weight in grams (150 in this case).
5. Remove the weight from the scale.
6. Push the Calibrate button (S1) again; the scale should display 0 grams.
7. Put the weights back on the scale. If the weight is right, the scale is set up correctly; if it isn't, repeat the calibration procedure starting from step 3.

That completes the calibration of the DIY kitchen scale project. Please note that the project is not approved for commercial use. ⏮

(150708)


**FROM THE STORE**

→150708-1
Kitchen scale PCB, bare

→150708-71
Kitchen scale kit of all parts

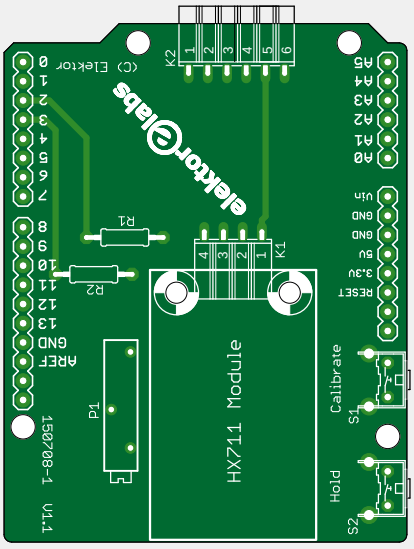

## COMPONENT LIST

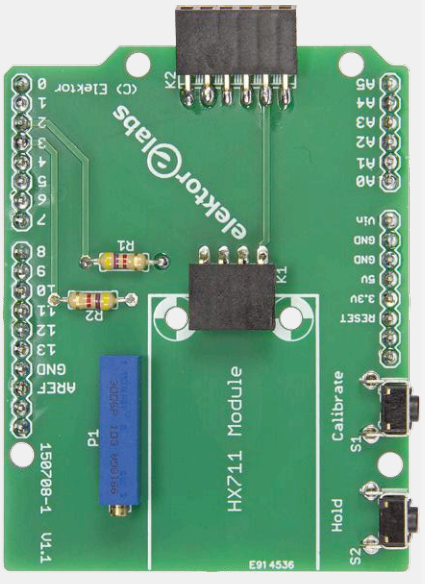
**Resistors**

R1,R2 = 4.7kΩ
P1 = 10kΩ 15-turn trimpot

**Miscellaneous**

K1 = 4-way SIL pinheader socket, 0.1" pitch, horizontal
K2 = 6-way SIL pinheader socket, 0.1" pitch, horizontal
1 pc 6-pin SIL pinheader, 0.1" pitch, vertical
2 pcs 8-pin SIL pinheader, 0.1" pitch, vertical
1 pc 10-pin SIL pinheader, 0.1" pitch, vertical
S1,S2 = tactile switch, PCB side mount
HX711 breakout board
2 pcs M2×10 mm bolt
4 pcs M2 nut
5 kg load cell, e.g. YZC-1B
PCB # 150708-1 from Elektor Store





**Web Links**

[1] www.elektormagazine.com/150708
[2] An Introduction to Stress Analysis and Transducer Design using Strain Gauges, www.hbm.com
[3] Original project: www.elektormagazine.com/labs/5-kg-kitchen-scale-built-on-arduino

# Hoverboard: Under The Hood

## Reverse-engineering a fun product

By **Dr. Thomas Scherer** (Germany)

As Christmas approaches so does the obligation of delighting your children, grandkids or yourself with a present that will give great joy. If a Segway is beyond your budgets and the garage is already full of bicycles, a hoverboard must be worth a shot. Or is it? These devices are now widely available at affordable prices, on eBay for instance. So we dismantled one and checked it out to discover precisely what lies 'under the hood'.

A self-balancing Segway scooter is a marvelous fun vehicle. Maybe you have already 'ridden' on one while on vacation, or taken part in a Segway tour, and experienced first-hand how much pleasure this form of motion provides in the open air. But a Segway is not just heavy and bulky but at around €9,000 / £8,120 / $10,540 it's plenty pricey too. So expensive in fact that back in 2009 the Elektor Wheelie [1] provided an affordable construction kit for a fun chariot of this type and enjoyed great popularity.



Figure 1. An elephant on a hoverboard. But don't try this at home or at the zoo, as it's a slight overstatement unless you buy a proper jumbo version.



Figure 2. The contents of the carton are clear to see.

Time moves on… and further developments and variants arose. Segway was bought up by the Chinese 'cloner' Ninebot [2], which had previously offered cheaper (and nastier) near-replicas for under €2,700 (£2,440 / $3,160). As a subsidiary of the electronics giant Xiaomi, Ninebot had sufficient capital for this 'absorbing' version of technology transfer. Since then Ninebot has brought to market, alongside its classic 'Elite' and various one-wheelers, some additional mini models of a Segway replica for below €1,000 — in particular the 'Mini-Pro' design for an unbelievable €520.

How come? And doesn't safety suffer as a result of price reduction? There is talk on the Internet of some "unpredictable results" with Ninebot products and videos are circulating on YouTube of heavy falls involving these models. To be fair, they also occurred with the original Segway, although these were down to driver error or simple over-enthusiasm. And if Ninebots are not unproblematic, then how do things look for hoverboards, which have been around for about two years now and are even cheaper?

### Basics

Under the titles 'Self-balancing scooter' (English), 'e-Board' (German) and 'Gyroskate' (French), Wikipedia contains relevant articles on this subject that also cover Segways. Some imprecision may arise from this broad classification. Specifically the hoverboard resembles a kind of mini Segway without a steering column; it is steered instead by displacing one foot against the other on the tread surfaces.

A hoverboard is therefore not a single, rigid footboard fitted with two wheels; as an alternative the tread is divided in the center with a joint between the two halves. By tilting sensor pads on these the rider can hopefully control the speed and direction of travel. The vehicle tries to maintain the tread plates horizontal. If the user leans slightly forward, the engines accelerate forwards, thus restoring the balance. When the user leans slightly backwards, the device logically also goes backwards. So far, so simple.

### Purchasing pointers

In order to form a personal impression of how safe these hoverboards are, the Elektor editorial team decided to buy one of these personal travel devices. According to Google, the widest

selection of hoverboards and best prices are to be found on eBay. But first we did some deeper digging on the Net. Here you can encounter some remarkable whoppers, as **Figure 1** effectively demonstrates.

In our searches and comparisons it became clear not only that all models stem from China but also that there is fundamentally only one design. It comes in several price ranges, numerous colors and with a profusion of warranted features for which you need a wild imagination. In practically every country eBay has several thousand examples on offer. In terms of performance there appears to be little to differentiate one from another, as they all have two motors with apparently 350 W of power each. For that reason they all employ a lithium battery pack with 36 V and 4.4 Ah, in other words around 160 Wh of standing energy available, which should be adequate for up to 20 km. In most cases the battery pack is supposed to originate from Samsung. According to sellers, their vehicles move at between 12 and 16 km/h (8–10 mph) and can transport people weighing up to 120 kg (260 lb).

Aside from their gimmicks and colors, hoverboards are differentiated above all in the diameter of their wheels. These can be had with solid rubber tires in diameters of 6.5" and 8" (from €140 / £130 / $160 and €170 / £150 / $200 respectively). Deluxe models with 10" pneumatic tires start at €230 / £ 210 / $ 270 or with 8.5" air tires from around €250 / £225 / $290. You can buy carrying cases, remote controls and even integrated Bluetooth speakers that let you pollute the environment with music if you hook them up to your smartphone. Once this was clear, the author decided for one of the 10" variants without all the Bluetooth kerfuffle in a decent shade of black [3]. One click and two days later the parcel had arrived.

## Unboxing and preparation

What you get in the carton can be seen clearly in **Figure 2**: the shiny brand-new Hoverboard (still adorned with excess strings of hot glue not removed after production) and to the left an AC power cable and a charger device (on top of the instruction booklet).

As for the power cable, this was crushed so badly in three places that there was a significant risk of short circuits (not to mention a 'stimulating' electric shock). An e-mail to the distributor's German warehouse brought a small package two days later with a new cable — plus a new power supply that had not been asked for. Unnecessarily good service!

A note in the carton explained that the battery pack was 'empty' and therefore need to be charged before using. I did as told and in 1.5 hours it was full. So what about the other 3 Ah? At this point, I also realized that it is not just a rumor when people on the Net talk about hoverboards catching fire during charging. **Figure 3** shows that the charger must be used unobstructed and well ventilated, and should never be covered.

On the bottom (**Figure 4**) you can see three stickers. The CE label is impressive only if you are unaware that manufacturers or importers self-certify their products. The safety instructions are to the right. Hardly any additional information is imparted by the amusingly so-called 'Other Sticker' on the left. To the left of the center joint is the start button and on the right a rugged, male charging socket with three pins.

Last but not least we come to the 'instruction booklet' — rightly put in quotes. It is four sheets of A4 paper. Of these eight pages two, set in very large print, contain very little content. Instead




Figure 3. After a while the small charger device gets decidedly warm.

of blather about CE symbols and so on plus the assuredly sensible warning to use the hoverboard only when wearing protective clothing (helmet, gloves and knee protectors). But some more in-depth information for newcomers would have been better. You can, after all, do yourself real harm, particularly during your first trials, and damage your home furnishings! To correct this shortcoming, we have assembled some tips for beginners in the text inset **Getting On Board For Beginners**. If you still want to buy a hoverboard after reading this report, study the text closely!

## Inner workings

For people like us having confidence in technology depends on knowing it intimately. So the plastic casing has to come off, in order to get a proper view of the inner workings. **Figures**



Figure 4. Hoverboard underside with stickers.

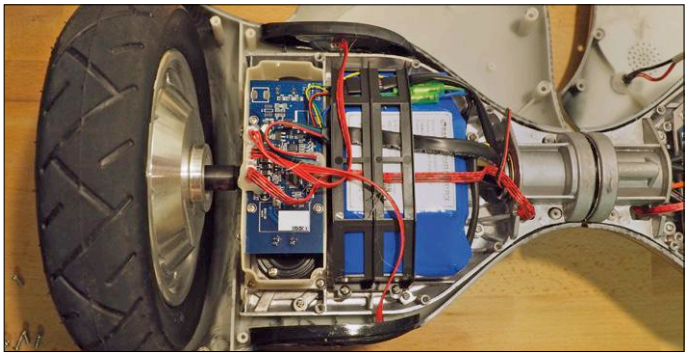


Figure 5. Left side uncovered, showing the foot switch and battery.

## Getting On Board For Beginners

**The best advice is**

### *trust the hoverboard!*

With hoverboards you play the key role in the balancing process. If you attempt to get onto the board from standing on the ground in the normal way while trying to keep your balance, then you have problems. The two dynamic control systems try to regulate in opposition, with different control speeds and amplitudes. Things can only go wrong! So let the hoverboard balance, then within a few seconds you get a feel how the apparatus behaves.
Your first attempts should be like this:

- Park the hoverboard on as smooth a surface as possible. Concrete is ideal. Lawns are a no-no.
- On the underside of the hoverboard look for a metal press-switch (see Figure 4). Push it once and a green symbol lights up on the top surface.
- Now place your left (or right) foot horizontally on one of the tread plates. Left and right are irrelevant on a hoverboard, as it drives equally well in both directions. When contact is made the green lamp lights up clearly.
- Now teeter (seesaw) with your foot slightly and feel how the board reacts. Do not rock it too much, otherwise it will take off at speed (and with luck then switch off).
- Now comes the point of no return, getting on board. Leave your first foot where it is and stand with the other foot on the opposite tread plate. Don't do this hesitatingly, nor too violently, but briskly.
- If you trust the hoverboard, now stand fully upright. It should feel completely stable, with very little wobbling. Keep still to begin with, developing a feel for the board's attempts to stabilize itself.
- Now you can start leaning forwards a little. You will – ooh, er! – move forwards. And discover intuitively how to brake and move backwards. Try this cautiously.
- Now you can check out how to teeter (seesaw) your feet in 'push-pull' mode: raise one foot gently and depress the other simultaneously. In this way you can make the lowered side move forwards and the raised side go backwards. You can turn around on the spot.
- Rotate the other way around. Practice turning in both directions; it may not be entirely intuitive but it's vitally important.
- Combine moving forwards with rotating, to turn curves. After five minutes or so you will more confident.

Children should be warned to be careful. As soon as they believe they have mastered the hoverboard, overconfidence sets in. Taking a curve too fast is a painful way of getting a physics lesson on centrifugal force.
It can help if you offer beginners a strong hand. The psychological support is more important than physical hand-holding.

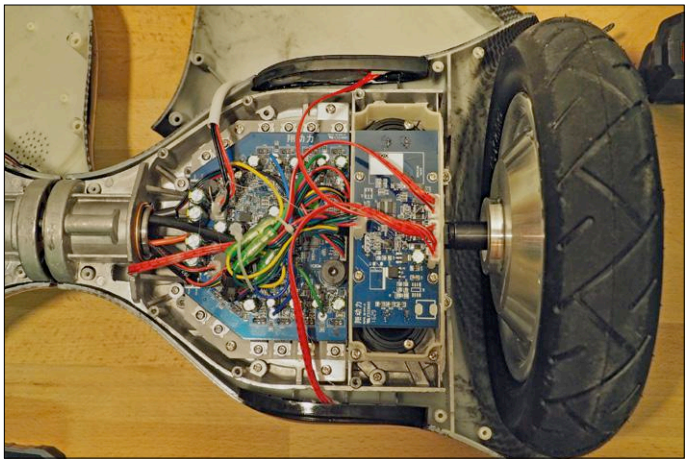


Figure 6. Right side revealed, with main board and foot switch.

**5** and **6** show the left and right sides. First you can see the metal frame made of aluminum. On the left is the blue-jacketed battery pack, on the right the main board with twelve power MOSFETs (for 2 x 3 half-bridge or totem pole circuits) for controlling the two brushless DC motors. In each case, the boards for evaluating the step signals are shown on the outside of the wheels. The joint (**Figure 7**), an active component with angle sensors (the red cables left and right), is located in the center. **Figure 8** shows how a rubber plunger on the underside of one of the tread surfaces actuates the fork light barrier soldered on the blue circuit board. Obviously an optical principle is trusted more than mechanical switches here. In **Figure 9**, the circuit board is disassembled and inverted so that the two photocells can be seen clearly. To the right, you can see the two rubber plungers, and the flattened axle of the right motor is fixed between them.
Now for the batteries. Because I was curious whether they had really fitted Samsung cells, I undid the battery pack and stripped off its blue plastic wrapping. Inside we find some standard commercial lithium-ion cells of the 18650 variety – in all likelihood type $LiCoO_2$ – of which two were wired in parallel and ten in series, indicating the designation of 10s2p and a nominal voltage of 36 V. There was no sign of the word Samsung any-

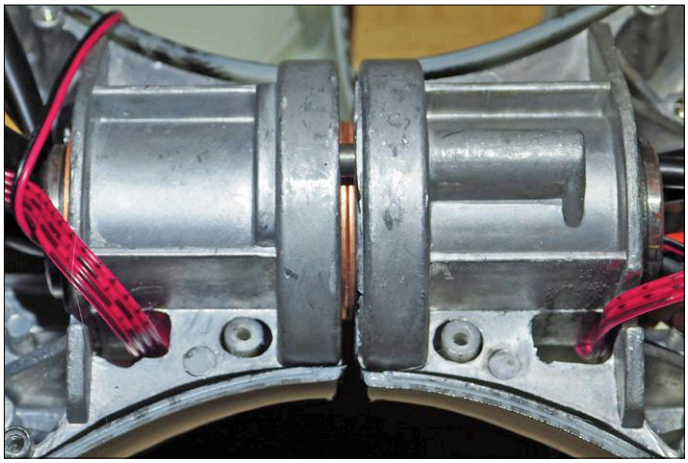


Figure 7. The swivel joint connecting the two halves.

where; instead it's a case of a no-name product. The capacity in Ah is nowhere to be seen either. However, as **Figure 10** indicates, the designation '7.2Wh' is clear to see, producing with a voltage of 3.6 V per cell a capacity of 2 Ah. In that case the battery pack's total capacity is definitely 4 Ah rather than the stated 4.4 Ah. A scam of 10%. But we can hardly quibble because all in all, considering the €230 demanded, we're getting a whole shedload of technology for our money.

### Driving experience and the bottom line

First off, I possess not only a Segway 'driver's license' (dating back to the time when this was still required in some regions of Germany) but also sufficient experience driving the Segway and on board numerous home-brew creations from the Maker scene. Seldom have I felt more unsafe on a self-balancing electro-scooter than on the hoverboard. Even after an hour it didn't feel nearly as intuitive as after just five minutes with a Segway. Why should this be?

## Balancing on pee-wee wheels

First up, owing to the method of turning curves by tilting the feet. This is far less intuitive than 'turning into the curve' on a scooter fitted with handlebars, where centrifugal force and gravity assist in maintaining your balance. If you make a fast turn with the hoverboard, you have to estimate how aggressively to turn into the curve, leaving the balancing entirely up to you. And there's the reaction time before you feel a response. Furthermore, it is difficult to coordinate the tilt angle of one foot against the other with the inclination of your body in the curve. The second aspect is the performance of the motors: 2 x 350 W, which we too would *like* to believe. But the Wheelie already had 2 x 500 W and other clones a solid 2 x 1 kW. An original Segway delivers a peak performance of 2 x 1.5 kW. Frankly a hoverboard is underpowered. My brow furrows at the 120 kg permissible load, because even with my 90 kg body weight the term 'acceleration' is fanciful. And when you have a shortage of power, even the most superb speed control is of little use.

On top of that, even at 10" the wheels are still on the small side, performing particularly badly on uneven surfaces. During my attempt to use the hoverboard on my back lawn, which is admittedly not quite up to English standards of evenness, I had to break off after 30 feet or so, with sweat beads on my brow. A hoverboard is no plaything then. It should be ridden only on flat ground without any loose gravel and using protective gear (never without a helmet!). And probably not by the older generation, because their offspring have better sensory and motor functions for dealing with this kind of device. Besides, they usually weigh less. Personally I would not let children under 10 years use them and above all never on the street (which is not allowed anyway). Out among traffic all of those CE labels provide no protection whatsoever.

By the way, the hoverboard survived teardown without ill effects...

(160508)

### Web Links


[1] Elektor Wheelie: www.elektormagazine.com/magazine/elektor-200909/3360

[2] We need to make clear that although Segway filed a trade complaint alleging Ninebot and other Chinese companies had violated its patents, Ninebot asserted that it "independently owns its intellectual property."

[3] Black 10" hoverboards on eBay: https://goo.gl/8oDgy2

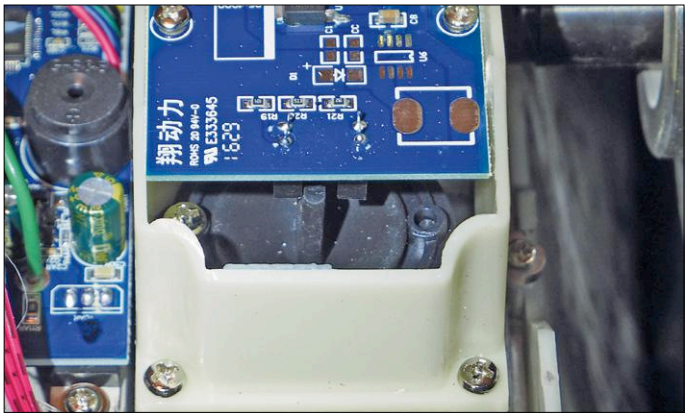



Figure 8. Foot switch with rubber plungers and light barrier seals.

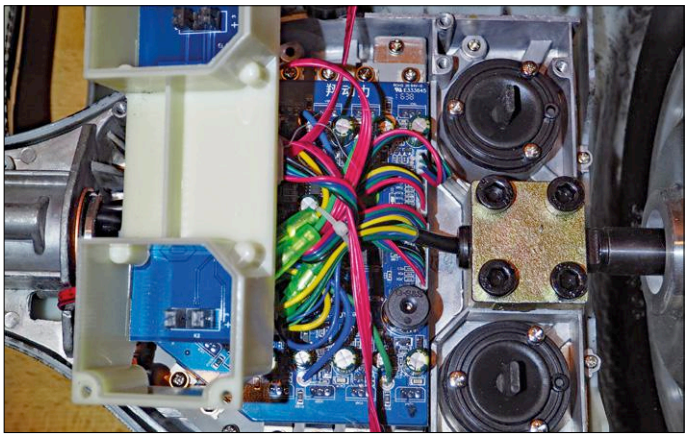


Figure 9. Foot switch opened up: two plungers and their light seals, also showing the attachment of the wheel axle.

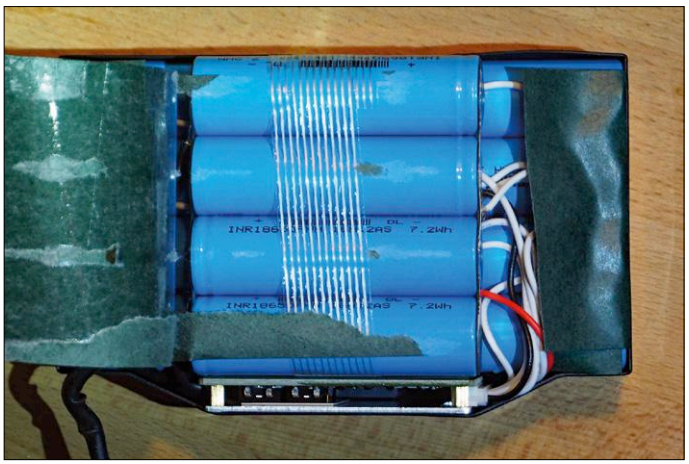


Figure 10. The battery pack stripped bare. No sign of any 'Samsung' markings.

# Raspberry Pi Internet Radio

## Tune in and chill out




By **Roy Aarts** (Elektor Labs)

With this project you can listen to any online radio station, without forking out on a dedicated Internet radio.

### Features

- Plays streaming audio from Internet radio stations
- Uses built-in audio jack
- Operates stand-alone
- Add your own favorite radio stations
- No soldering required

Internet radios are available widely and in many shapes and sizes, but they can get quite expensive. With this project you can make one yourself using the by now ubiquitous Raspberry Pi 2. Through the headphone jack it plays audio decoded from a radio stream on the Internet. To make it you need a Raspberry Pi, a 2.8-inch touchscreen, an SD card, and optionally a Wi-Fi dongle. The user interface is made in python using pygame. The code is easy to understand and can be adapted to your own taste.

### Installation

First we need to install Raspbian on the Raspberry Pi. Download Raspbian Jessie from [1] and use Win32DiskImager, free download from [2], to burn the *image* on an empty SD card with a capacity of at least 4 GB. When done, insert the SD card in the Raspberry Pi SD card slot, attach a monitor, mouse, keyboard and a USB Wi-Fi dongle (or a hard-wired Internet connection) and power it up. When the Raspberry Pi has finished booting we can set the network like in the top right corner in **Figure 1**. After connecting to the network, write down the IP address of the Raspberry Pi. This is useful for configuring it later. To get the IP address, hover the mouse over the icon used to connect to the network.

Now we have to install the touchscreen libraries. If you haven't already attached the touchscreen, now is the time. But not before having shut down the Raspberry Pi (top left corner, select Shutdown)! Attach the touchscreen and power the RPi on again. We will install the necessary drivers via the computer. Download PuTTY from [3], start the program and enter the IP address of your RPi in the bar that says "Host address (or IP address)". If a notification pops up, click "Yes". Now a terminal window should open, asking you to log on. Log on with username "pi" and password "raspberry". Now type

```
sudo wget -N https://github.com/watterott/RPi-Display/raw/master/rpi-display.sh
```

and press Enter. This will download the necessary files from github. When the prompt returns, type

```
sudo /bin/bash rpi-display.sh 270
```

This will install the github files in the system. After some time, some questions appear. Answer them as follows:

| | | |
|---|---|---|
| • Activate the console on the TFT display? | y/n | **Y** |
| • Install fbcp (Framebuffer Copy)? | y/n | **N** |
| • Install xinput-calibrator? | y/n | **Y** |
| • Install tslib (touchscreen library)? | y/n | **Y** |
| • Calibrate touchscreen now? | y/n | **N** |
| • Reboot the system now? | y/n | **Y** |

The Raspberry Pi will reboot and the connection with the computer will be lost. If everything goes right, you should see the user interface on the touchscreen when the Raspberry Pi boots up.

### Calibration & music player

To configure the touchscreen, reconnect to the RPi using PuTTY. Log in and type

```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/
input/touchscreen ts_calibrate
```

On the touchscreen you'll see the calibration program (see **Figure 2**). After completing the steps shown on the screen the touchscreen is calibrated.
Now we can install the music player client ('mpc') and daemon. To do this, type the following commands:

```
sudo apt-get update
sudo apt-get install mpd mpc
```

### User interface

Now we need to install the user interface. For this we use winSCP (free download from [4]) and the download provided on the Elektor magazine website for this project [5]. We have to copy the files from the download onto the SD card in the Raspberry Pi. So we close PuTTY and start winSCP. We log in with the same credentials as used in PuTTY.

If a notification pops up, click "Yes". From the download we copy the folder 'pi-radio' onto the Raspberry Pi. Then we copy the files 'launcher.sh' and 'shutdown.sh' to the Desktop folder of the Raspberry Pi. These two files launch the radio and shut down the RPi. Now we can close winSCP and open PuTTY again.

The two files we just copied onto the Pi's desktop do not work as clickable icons yet. To fix this, in PuTTY, type:

```
sudo chmod +x /home/pi/Desktop/launcher.sh
```

and

```
sudo chmod +x /home/pi/Desktop/shutDown.sh
```

### Radio stations

The radio is now installed, but there are no radio stations to be heard yet. To add a radio station, we need to have its streaming link. This is the address from where the audio is streamed. A practical website listing these links is SomaFM [6]. This website lists many free stations of different genres and provides streaming links we can use. To retrieve a usable link from SomaFM, we click on the station we want and click on "Direct stream links". The link we need is beside "Direct server:".

To add a station, we open PuTTY and enter mpc add followed by the station url. For example:

```
mpc add http://ice1.somafm.com/bagel-128-mp3
```

Links ending with '.m3u' will not work. There are many other stations available from other websites that can be added, you're not limited to SomaFM. A url can be tested by entering it in the web browser on your computer. If it's a valid link, a music player will open, playing the radio station.
To list the added radio stations, we can use the command mpc playlist. This lists all added radio stations. To remove a radio station, type mpc del followed by the number in the playlist in PuTTY, for example: mpc del 2 removes the station in the second position.

### All set to use

The radio player can be started now by double tapping the icon on the touchscreen and pressing 'Execute'. The interface (**Figure 3**) has been kept simple so that it can be operated with your fingers rather than a stylus (**Figure 2**). These are the available commands:

- 'Play' starts the mpc on the current playlist.
- 'Pause' stops the mpc on the current playlist.
- 'Volume down' reduces volume.
- 'Volume up' increases volume.
- 'Mute' sets the volume to zero.
- 'Refresh' stops the player and starts it again, refreshing the screen.
- 'Exit' closes the radio player interface but keeps the mpc playing.

To conveniently shut down the Raspberry Pi, double tap the shutDown icon and press 'Execute'. Happy listening! ⏮

(160043)

### Web Links


[1] www.raspberrypi.org/downloads/raspbian
[2] https://sourceforge.net/projects/win32diskimager
[3] www.chiark.greenend.org.uk/~sgtatham/putty/download.html
[4] https://winscp.net/eng/download.php
[5] www.elektormagazine.com/160043
[6] http://somafm.com

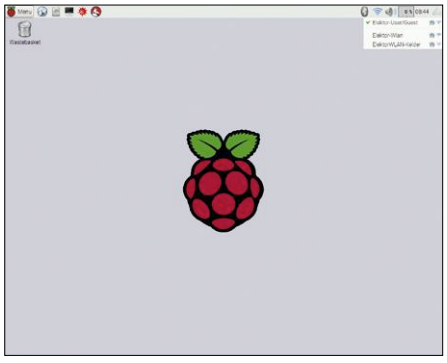



Figure 1. Here we can configure the RPi network settings.




Figure 2. Calibration the touchscreen before operation...




Figure 3. The interface can be operated by finger — no stylus needed.

# Err-lectronics

## Corrections, Updates and Feedback to published articles

### 2-Terminal Dimmer **Elektor Magazine 4/2017, p. 14 (160380)**

CORRECTION. The values of resistors R2 and R4 are given incorrectly as kΩ in the parts list. Their correct values should read 330 Ω and 560 Ω respectively, as shown (correctly) in the schematic. It's worth noting that the value of R4 used in the low-voltage version must be chosen according to the supply voltage used. Details of how its value can be calculated are given in the article.

### Lumina – a Bluetooth Low Energy connected lamp **Elektor Magazine 2/2016, p. 80 (130226)**

CORRECTION. The space provided for mounting the quartz crystal onto the PCB is not large enough. It will be necessary to either mount the crystal at the underside of the board or use the microcontroller's internal oscillator.

UPDATE. Under Android 7 it is now necessary to grant location permissions before a Bluetooth connection can be established. It was necessary to add two lines to the *AndroidManifest.xml* file and change its version number to *1.0a*. Two lines of code were also added to the *DeviceScanActivity.java* file, to handle this new permission. The complete code can now be recompiled using the latest version of Android-Studio and the app works under Android 7.0. These new files can be downloaded from the project web page: www.elektormagazine.com/130226.

### Dimmable Outdoor Lighting **Elektor Magazine 3/2017, p. 76 (140574)**

FEEDBACK. The connection between the optocoupler and mains voltage is made using four 0.25-W resistors connected in series. For safety you should always use 0.5-W resistors, which are suitable for operation at 250 V DC. The type of optocoupler specified and its PCB layout are also questionable. A VDE certified optocoupler with a wider 0.4" (10.16 mm) lead separation together with a slot in the PCB would provide air separation between the high and low voltage leads of the optocoupler, as well as increase the creepage path length.

*Volker Schulze*

The 0.25-W resistors used here are specified to handle 250 V. There is no issue from the safety point of view using these four series-connected resistors.

The optocoupler specified in this design is UL, CSA and IEC/EN/DIN EN 60747-5-2 compliant (The last spec referred to here was introduced on the 1st January 2004 and replaces the VDE 0884 safety standard).

We always use a creepage distance of 6 mm from tracks carrying AC line voltages on our PCBs but it wouldn't hurt to also cut a slot in the board under the optocoupler.

*Luc Lemmens, Elektor Labs*

### Bluetooth Controlled Relays **Elektor Magazine 6/2016, p. 104 (150648)**

FEEDBACK. The Bluetooth relay controller only worked for me after I programmed the ATtiny85 to run with a clock frequency of 1 MHz and not 8 MHz as suggested in the Arduino sketch.

I used the Arduino IDE to program the microcontroller; it was first necessary to install the following package in the IDE using the board manager: https://github.com/SpenceKonde/ATTinyCore

*Hans Schneider*

Our developer Roy Aarts used the 'attiny' board package maintained by David A. Mellis: https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json

After installation, use the Arduino IDE Boards Manager and select:

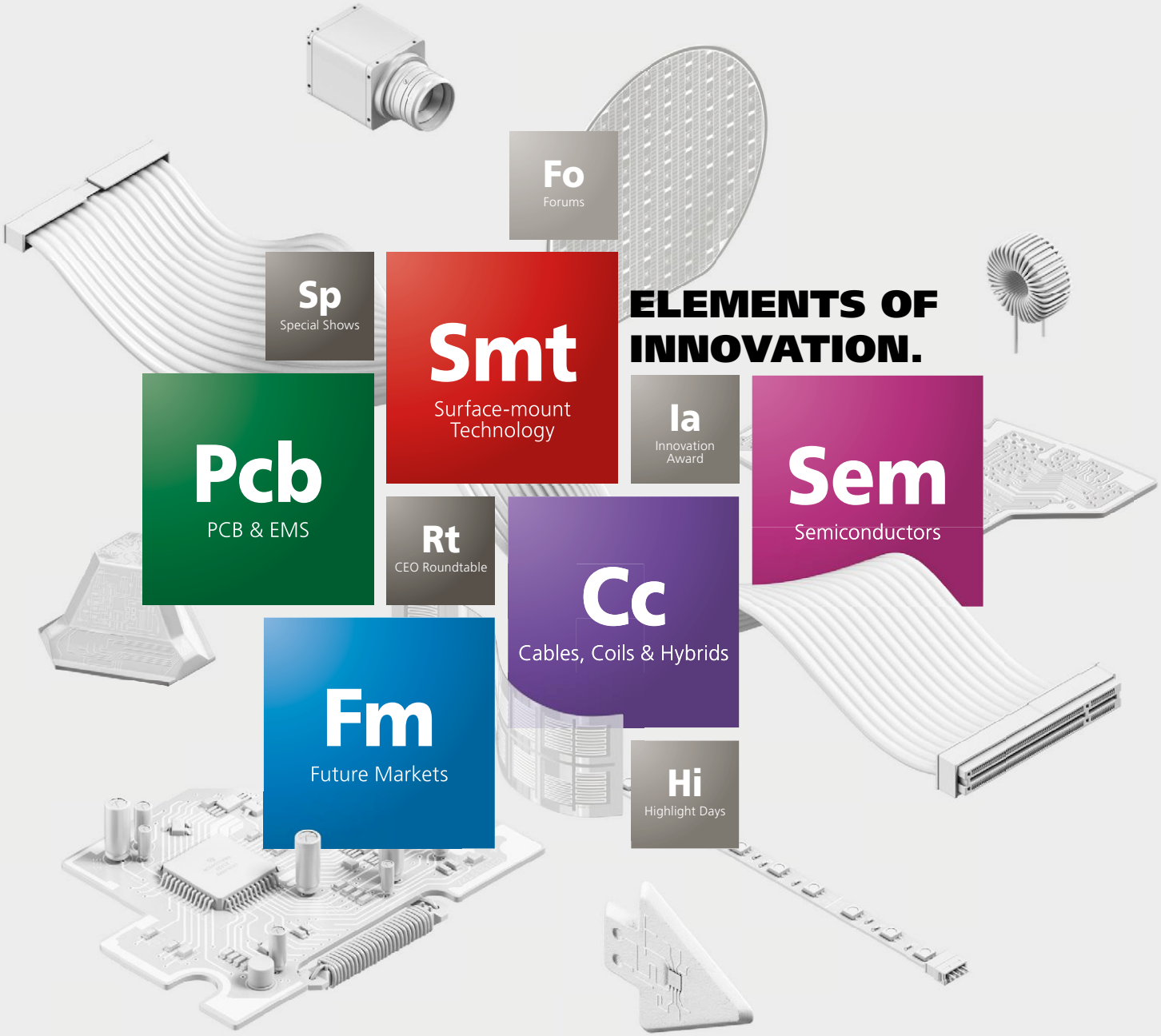
Tools → Board → ATtiny25/45/85 Tools → Processor → ATtiny85 Tools → Clock → Internal 8 MHz

*Clemens Valens, Technical Manager Elektor Labs*

Messe München

Connecting Global Competence

→ Buy tickets now!




World's Leading Trade Fair for
Electronics Development and Production
November 14–17, 2017
Messe München
**productronica.com**

SEMICON EUROPA
semi
Co-located event

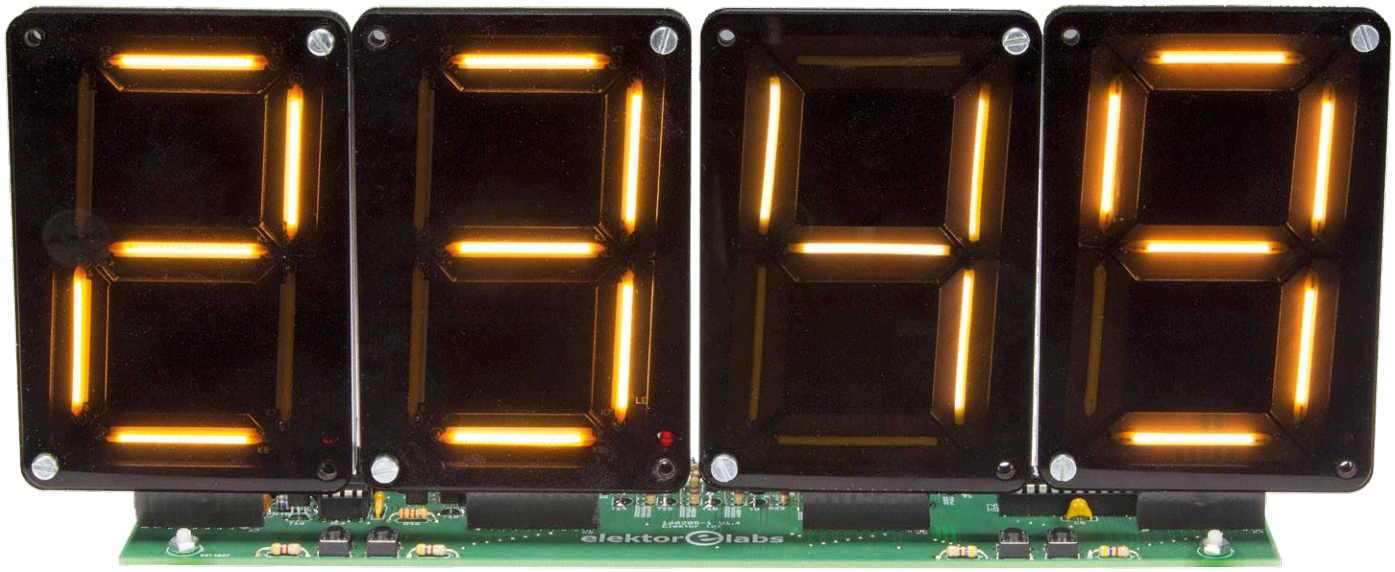
IT2 INDUSTRY

productronica 2017
innovation all along the line

eLABS PROJECT

# LEDitron Scoreboard

## With discrete 7-segment displays



By **Roy Aarts** and **Jan Visser** (Elektor Labs)

Believe it or not, there was a time when hobbyists found 7-segment displays with red LEDs (cutting-edge technology at that time) hard to come by and virtually unaffordable. If you absolutely needed a 7-segment display, you had to make it yourself from cardboard, translucent red film and incandescent lamps.

### Features

- Scoreboard and timer
- Microcontroller driven
- LEDitron displays
- No SMD components
- 12 $V_{DC}$ supply voltage

In the September & October 2016 issue of Elektor magazine we revisited that bygone era in certain sense. There we presented a discrete 7-segment display with impressive dimensions (about 8 x 11 cm / 3 x 4.25 in) made from modern LED filaments. This LEDitron display aroused a lot of interest, which prompted us to devise a practical use for it.

### The task

After giving a good deal of thought to how to best take advantage of the robust dimensions and good legibility of these displays — even at relatively great distances — we decided to build a combined scoreboard and timer. Because we wanted to use four displays in this project (with two pairs of digits, so the scoreboard could be used for many types of games, including basketball) and we naturally intended to use a microcontroller to drive them, we modified the original LEDitron power supply and designed a completely new PCB for the power supply and motherboard. And of course we also retained the friendly nature of the original LEDitron, which meant using only conventional through-hole components.

### The LEDitron

The schematic of the LEDitron display (from the original article) is reproduced in **Figure 1**. Actually you hardly need a schematic to describe this group of seven artfully arranged LED filaments, a normal LED and a series resistor. Aside from the relatively high supply voltage of around 70 $V_{DC}$ for the filaments, it is a perfectly normal common-anode display made from discrete components.

Let's briefly recapitulate the construction of the display, so we can set it aside and move on to the actual scoreboard.

The display PCB layout and component list are shown in **Figure 2**. If you nose around a bit on the Internet, you can readily find some addresses where you can get the LED filaments (search for repair — buy). However, you should

bear in mind that these filaments are very fragile and do not always survive shipping — the postal service is not noted for gentle handling of parcels. If you want to be on the safe side, you can buy several LED lamps with filaments and dismantle them (carefully) to salvage the filaments.

The best way to mount the filaments is to use socket pins extracted from an IC socket or a socket header. After fitting the pins on the PCB, bend the ends of the filament terminal strips at right angles and insert them in the socket pins on the board. Make sure you get the polarity right. The anode of an LED filament is marked by a red spot or a small hole in the terminal strip. Use the schematic in Figure 1 for reference. Note: Some filaments have terminal strips which cannot be bent, as we learned the hard way. In that case, you can fit solder pins on the PCB and then solder the filaments to these pins.

If you want to do a really nice job, you can fit barriers between the individual filaments (using strips of black acrylic, for example) and place an optical filter in front of the display. We used 3-mm "Umbra" plastic for this, which gives the display a nice appearance. If you are not keen on messing around with pieces of acrylic and a coping saw, you can purchase a complete kit with the PCB and acrylic parts (but not the filaments) in the Elektor Store.

Each display is equipped with a separate LED (LED1) to act as a decimal point.

For more information about LED filaments and the LEDitron display, please see the relevant Elektor article [1].

## The scoreboard hardware

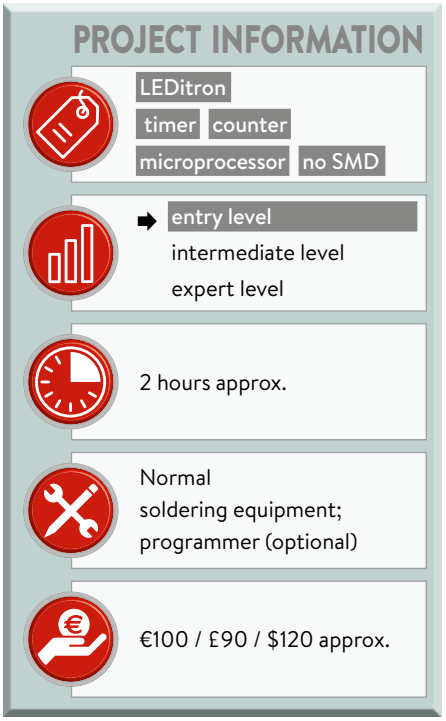
**Figure 3** shows the full schematic diagram of the scoreboard. Let's start with the description of the power supply. The 12 $V_{DC}$ external supply voltage (provided by an AC adapter or something similar) enters on connector K1, with a 2-A polyfuse for protection. Note that there is no provision here for reverse-polarity protection. Voltage regulator IC1 (a 78L05) converts the input voltage into a nicely stabilized 5 V supply voltage for the microcontroller (IC2).

For the displays we need a supply voltage of approximately 70 V, which we have to derive from the 12 V input voltage in a different manner. For this we use an old standby: a 555 timer (IC3), in this case the more modern CMOS version (type 7555). The timer IC is wired as an astable multivibrator and outputs a pulse waveform at about 76 kHz on pin 3. This arrangement acts as discrete step-up converter with capacitor C9 (note the 100 V voltage rating) charged through L1 and D1 to a voltage in the range of 35 to 100 $V_{DC}$. Voltage feedback to IC3 is provided by a voltage divider (R48, R49 and P1) and transistor T16, so that the output voltage can be set to the desired 70 V level by adjusting P1. Despite the simple design of this power supply circuit, it achieves an efficiency of nearly 85%.

The displays are driven in multiplex mode, as otherwise the current consumption would be much too high. Transistors T8/T9 – T14/T15 supply current to the common-anode terminals of the displays, while transistors T1–T7 switch the cathodes of the individual segments (filaments) to ground. These transistors are driven by an ATMega328P microcontroller (IC2).

If you are wondering what happened to the driver transistors for the decimal points of the displays, the answer is that there aren't any. These LEDs draw so little current that they can be driven directly by the ATMega outputs.

With regard to the ATMega, the clock signal is regulated by crystal X1 and the associated capacitors, and connector K8



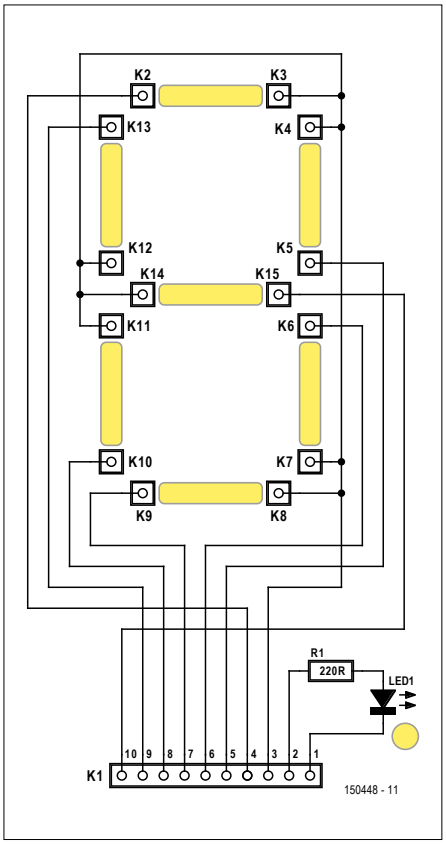




Figure 1. The schematic diagram of the LEDitron display is very simple.

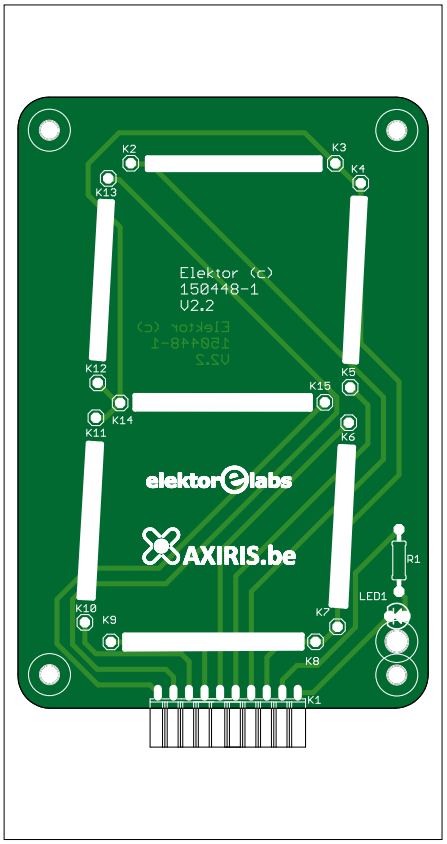



Figure 2. The seven LED filaments are mounted in the form of the digit 8. Make sure you get the polarity right.

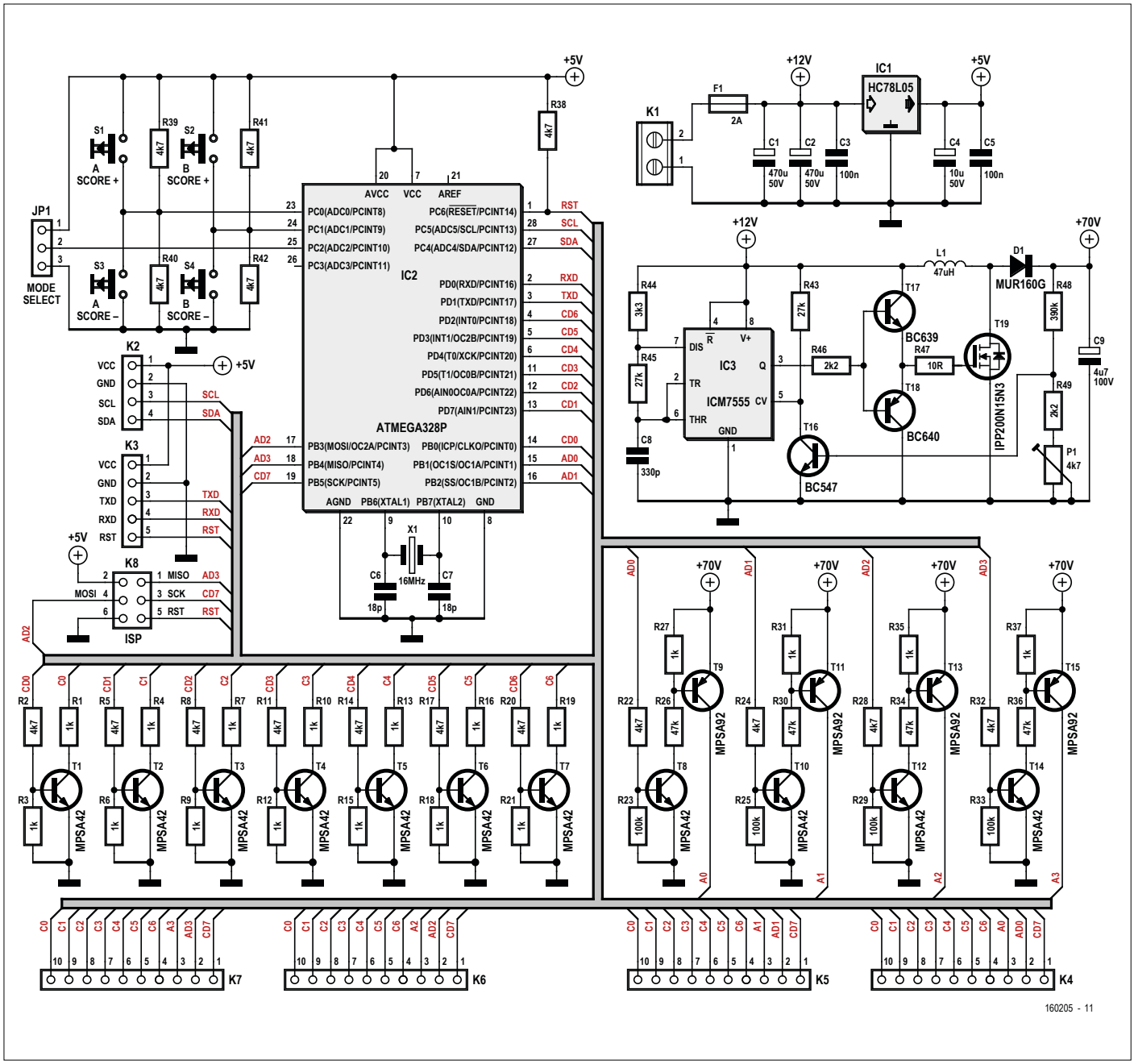



Figure 3. Schematic diagram of the scoreboard. The circuit is built around a microcontroller.

is provided for in-system programming (ISP). Connectors K2 and K3 support communication options, which are not necessary when the scoreboard is used stand-alone.

The user interface (more about this later) consists of a jumper (JP1) to select the operating mode (timer or counter) and four pushbuttons (two for each pair of displays). That confronted us with a problem: the ATMega328 has a limited number of I/O pins, and we need most of them for the multiplexed 7-segment displays. For the four pushbuttons we would need four I/O lines, and there are not that many left. We therefore had to look for a different solution, which we eventually found.

The microcontroller has several analog inputs (A/D converter inputs) which can be adapted to the purpose of sensing the four pushbuttons, so that we only need one input for each pair of buttons. A nifty solution. To see how it works, let's take pushbuttons S1 and S3 as an example. When neither of the buttons is pressed, the input voltage on pin 23 (PC0) is equal to half the supply voltage (from voltage divider R39/R40). When S1 is pressed, it shorts out resistor R39 and the full supply voltage (5 V) is present on the input. Pressing S3 connects the input directly to ground (0 V). If both buttons are pressed at the same time, the supply voltage is shorted out, which causes the timer/counter to be reset. Of course, the buttons should only be pressed briefly for this. The same story applies to buttons S2 and S4.

## The scoreboard software

We wrote the code for the ATMega328P in the Arduino IOE, which makes it easy

for users to adapt it their own wishes if necessary. Here we describe the key points in more detail. Most of the program is self-explanatory, without any baffling tricks.

```
#include <TimerOne.h>

const uint8_t leds[7] = {2, 3, 4,
   5, 6, 7, 8};
const uint8_t seg[4] = {9, 10, 11,
   12};
const uint8_t dotPin = 13;
int curSeg = 0;

//two buttons per pin. Connected
   to analog inputs
const uint8_t button1 = A0;
const uint8_t button2 = A1;
const uint8_t jumper = A2;
byte buttonsPressed = 0;
#define S1 0
#define S2 1
#define S3 2
#define S4 3
```

First we include the TimerOne library. We need it so that we can call the function which causes the right display to light up at the right time. Then we define an array of the pins that are connected to the segments of the displays. This make it easy to call them every time with the same function.

```
void setup() {
  // put your setup code here, to
   run once:
  for (int i = 0; i < 7; i++)
    pinMode(leds[i], OUTPUT);
  for (int i = 0; i < 2; i++)
    pinMode(seg[i], OUTPUT);
  pinMode(dots, OUTPUT);

  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
  pinMode(jumper, INPUT);

  Timer1.initialize(3000); //3ms
  Timer1.
   attachInterrupt(writeDisplay);
}
```

The setup() function configures the pins connected to the segments as outputs and the pins connected to the pushbuttons and the jumper as inputs. Finally, Timer1 is configured so that it invokes the function “writeDisplay” every 3 milliseconds.

```
void loop() {
  // put your main code here, to
   run repeatedly:
  if (digitalRead(jumper)) { //
   scoreboard
    if (analogRead(button1) > 500
   && ((buttonsPressed >> S1) & 1)
   != 1) {//S1 pressed -> score
   1 +
      buttonsPressed |= (1 << S1);
      if (count / 100 != 99)
        count += 100;
    }
    if (analogRead(button1) <=
   500)
      buttonsPressed &= ~(1 <<
   S1);

...

else { //timer
    if ((analogRead(button1) > 500
   || analogRead(button1) < 20) &&
   ((buttonsPressed >> S1) & 1) !=
   1) {
      buttonsPressed |= (1 << S1);
      running = true;
    }
```

The loop() function tests the state of the jumper (JP1) and then reads the buttons. If the jumper is in the “scoreboard” position (1-2), the pushbuttons are read and depending on which button is pressed, the score is incremented or decremented by 1.
As previously described, the pushbuttons supply analog values, so we use the analogRead() function to read them by comparing the voltage on the input to a fixed reference value.

```
void writeDisplay() {
  int number[4];
  number[0] = count % 10;
  number[1] = (count / 10) % 10;
  number[2] = (count / 100) % 10;
  number[3] = (count / 1000) % 10;

  for (int i = 0; i < 4; i++)
    digitalWrite(seg[i], LOW);

  delayMicroseconds(500);

  byte toWrite =
   toSegment(number[curSeg]);
  for (int i = 0; i < 7; i++)
    digitalWrite(leds[i], (toWrite
   >> i) & 1);

  delayMicroseconds(500);

  digitalWrite(seg[curSeg], HIGH);

  curSeg++;
  if (curSeg >= 4)
    curSeg = 0;
}
```

Finally, the writeDisplay() function shows the state of the counter or the timer on the display. One digit is refreshed each time this function is called, which occurs at 3 ms intervals as previously described. The function first splits the value to be shown on the display into two separate digits. Then the segments are blanked, followed by a 500-µs delay. Next the function toSegment() is called to select the segments that should be lit for the digit concerned, and the associated outputs of the ATMega328P are asserted. The appropriate display is enabled after a second 500 µs delay. This sequence is repeated for each of the four digits, after which the entire cycle starts over again.
If you order a pre-programmed microcontroller in the Elektor Store, you can simply plug it into its socket and start using the circuit. However, if you want to experiment with the firmware or you want to use the minutes version of the firmware, you will have to program the ATMega328P yourself. You can do that in-system via connector K8 or use a separate programmer. In that regard we can recommend the TL866A, which is also available in the Elektor Store [2]. The firmware for the scoreboard can be downloaded for free on the project page [3].

## Operation

This does not need a lot of description. Jumper JP1 configures the circuit as either a scoreboard or a timer. In the former case, buttons S1 and S3 increase and decrease the score shown by the left-hand pair of displays, while buttons S2 and S4 do the same for the right-hand pair.
In timer mode, the elapsed time (in minutes and seconds) is shown on the four displays. Pressing S1 or S3 starts the timer; pressing S2 or S4 stops it.

## COMPONENT LIST

### DISPLAY BOARD

**Resistors**

R1 = 220Ω

**Semiconductor**

LED1 = 5mm

**Miscellaneous**

K1 = 10-pin SIL pinheader, angled
PCB # 150448-1

### MOTHERBOARD

**Resistors**

Default: 5%, 0.25W

R2,R5,R8,R11,R14,R17,R20,R22,R24,R28,R32, R38,R39,R40,R41,R42 = 4.7kΩ
R23,R25,R29,R33 = 100kΩ
R1,R3,R4,R6,R7,R9,R10,R12,R13,R15,R16,R18,R 19,R21,R27,R31,R35,R37 = 1kΩ
R26,R30,R34,R36 = 47kΩ
R43,R45 = 27kΩ
R46,R49 = 2.2kΩ
R47 = 10Ω
R48 = 390kΩ
R44 = 3.3kΩ
P1 = 4.7kΩ trimpot, horizontal

**Capacitors**

C1,C2 = 470µF 63V
C3,C5 = 100nF
C4 = 10µF 63V
C6,C7 = 18pF
C8 = 330pF
C9 = 4.7µF 100V

**Inductors**

L1 = 47µH

**Semiconductors**

D1 = MUR160G
IC1 = LM78L05
IC2 = ATMega328P, programmed
IC3 = ICM7555
T1,T2,T3,T4,T5,T6,T7,T8,T10,T12,T14 = MPSA42
T9,T11,T13,T15 = MPSA92
T16 = BC547
T17 = BC639
T18 = BC640
T19 = IPP200N15N3
F1 = 2A PTC (polyfuse)

**Miscellaneous**

X1 = crystal 16MHz
K1 = 2-way PCB screw terminal block, 5mm pitch
K2 = 4-way SIL socket header
K3 = 5-way SIL socket header
K4–K7 = 10-way SIL socket header
K8 = 6-pin (2×3) pinheader
JP1 = 3-pin jumper
S1–S4 = pushbutton switch, NO

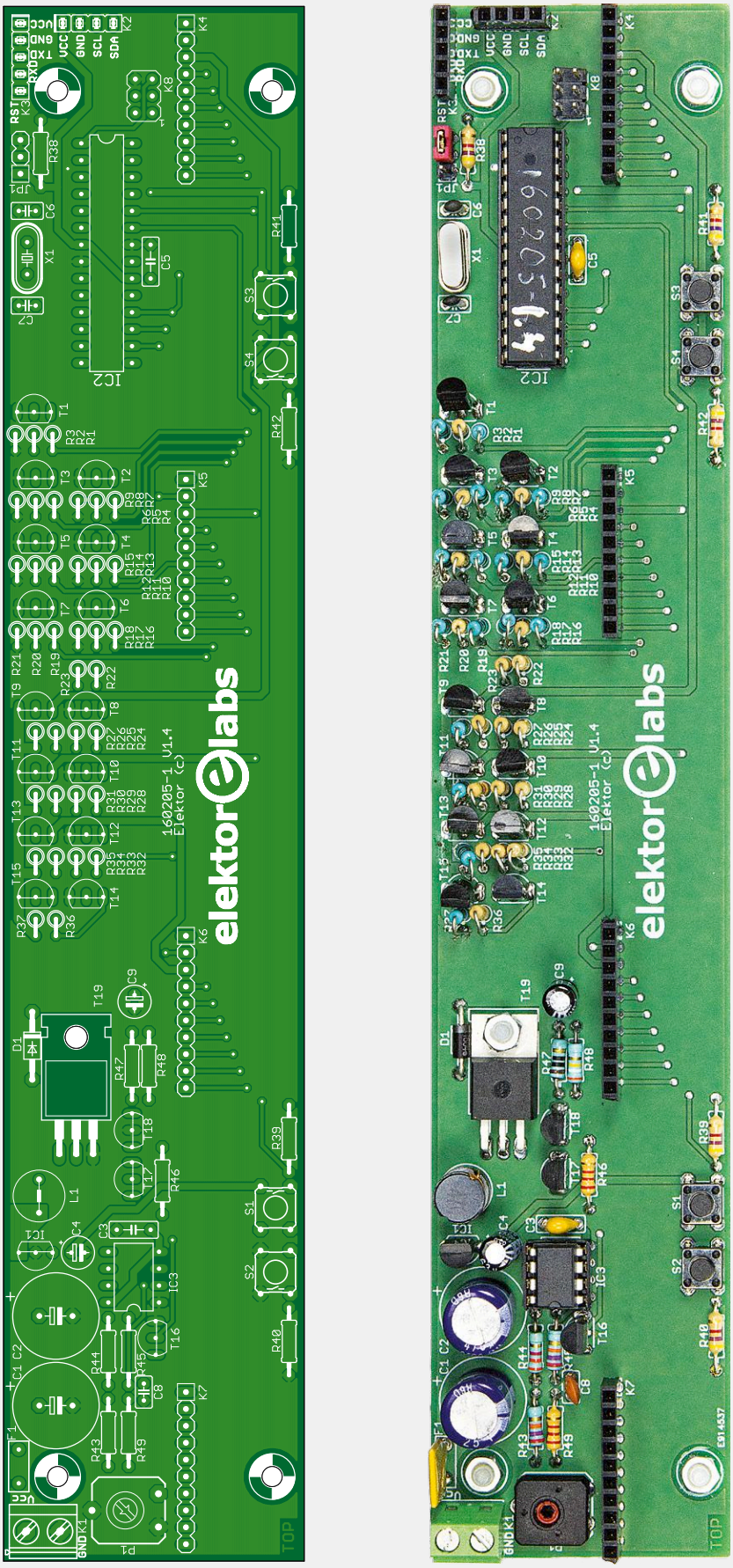



Figure 4. The PCB is double-sided, and only conventional components are used.

No reset button is provided. To reset the timer or counter, you must briefly switch off the power. Another option is to press S1 and S3 (or S2 and S4) at the same time (briefly!), which shorts out the 5 V supply voltage for the microcontroller, resulting in a counter/timer reset.

### Building and testing

Construction of the four displays has already been described briefly at the beginning of this article, and more detailed information is available in the original article [1].

For the scoreboard, we have designed a PCB (double-sided) as usual, which is shown in **Figure 4**. All components are conventional through-hole type, so anyone with basic soldering skills should be able to build the circuit without any difficulty.

Start with the sockets for IC2 and IC3. followed by the low-profile components, then the transistors and the taller components (electrolytic capacitors and connectors). **Figure 5** and **Figure 6** give a good impression of the finished circuit. Do not insert IC2 and IC3 in their sockets right away.

First do a careful visual inspection of the PCB — check the polarity of the electrolytic capacitors and the diodes, look for any misplaced solder splash or solder bridges, and so on. Then turn potentiometer P1 completely counter-clockwise. Next, insert IC3 (the 7555) in its socket (note that the notch in the package should face toward P1) and connect the 12 $V_{DC}$ supply voltage (from an AC adapter, for example).

Then check the 5 V supply voltage on the board to make sure that it is okay (for example, between pins 7 and 20 of the socket for IC2). Next, measure the voltage between the cathode of D1 and ground. It should be approximately 35 V. If that is okay, you can relax – the hardest part is behind you now. Now slowly turn P1 clockwise until the voltage between the cathode and ground is approximately 70 V. Then switch off the power and insert the pre-programmed microcontroller in its socket. The notch in the package should face toward K8. Now insert the four display boards in connectors K4–K7, and the scoreboard is ready to use.

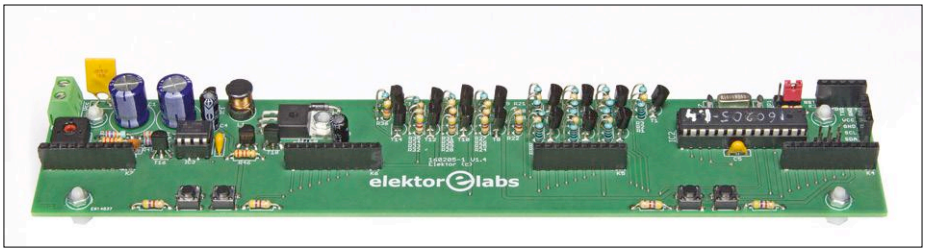



Figure 5. The fully assembled board.

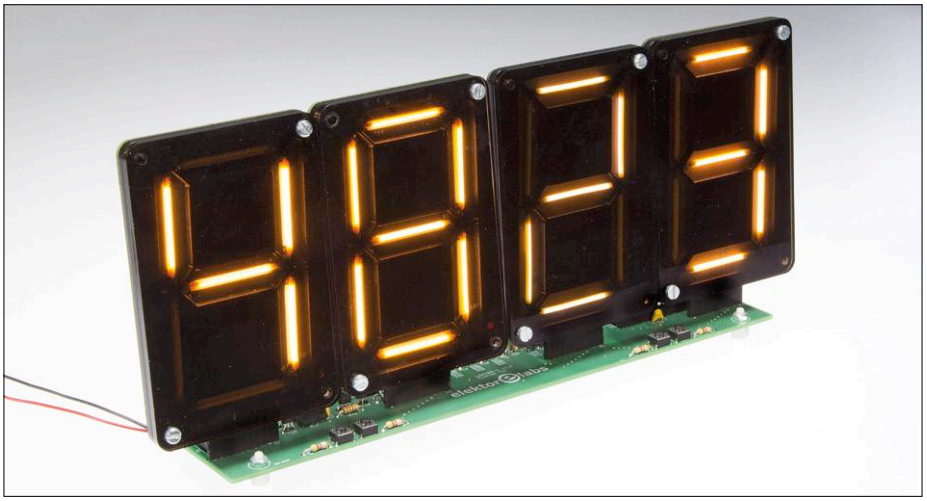


Figure 6. Once the display boards are plugged in, the scoreboard is ready to use.

### Final remarks

Although the scoreboard as described here works well and is certainly usable, we would like to point out that this project is primarily intended to serve as a source of inspiration. We encourage you to modify the hardware and software as you see fit, and we cordially invite you to show us the results of your efforts. ⏮

(160205-I)

### Web Links


[1] www.elektormagazine.com/150448

[2] www.elektor.com/tl866a-universal-programmer

[3] www.elektormagazine.com/160205


### Warning

Although the input voltage of this circuit is approximately 12 V, the voltage is raised by the step-up converter to a maximum of 100 V, which is a potentially hazardous voltage. The circuit must therefore be fitted in an enclosure providing suitable protection against accidental contact.


### FROM THE STORE

→150448-1
LEDitron display PCB (4 required)

→150448-71
LEDitron kit including acrylic parts, without filaments (4 required)

→160205-1
Scoreboard motherboard PCB

→160205-41
Programmed microcontroller

# The Wang 320SE:
## a Time-sharing Calculator (ca. 1970)

### Discrete transistors, core memory, and log-based math... far out!

When a friend at work said, "I found something in the Chemistry department's dungeon and I know you like Nixie tubes..." I had no idea where it would lead. Here's the story, specially adapted for Retronics, my favorite section in Elektor Magazine.




Figure 1: A single Wang 320K Keyboard/Display unit.

By **Professor Brian White** (USA)

In the dungeon, when he showed me what he had (**Figure 1**), it was like being handed a single T-Rex tooth — the rest of that beast had to be nearby. I hadn't seen a calculator like that since I visited the Boston Museum of Science when I was a boy. I remembered four keyboard/display units sharing a central processor (**Figure 2**) that had real core memory. If he'd found one keyboard, the other three and the processor just had to be there too. To my delight, they were all there; dusty, rusty and neglected but complete with a manufacturing date of May 19, 1971.

**One of many, or not?**

The Wang 320 has many features that would be familiar to modern calculator users: 14-digit calculations with 10-digit display and rounding, add, subtract, multiply, divide, ln(x), $e^x$, $x^2$, and $\sqrt{x}$. Addition and subtraction use a familiar key sequence: to add 12 and 15 to the left accumulator (there is also a "right accumulator"), you would press "12", then "Left +", then "15" and "Left +" again to get a running total of 27. It does have some quirks though. Multiplication and division use reverse Polish notation (RPN) — to multiply 22 × 14, you press "22", "enter", "14" and "X=" to get 308. There is no scientific notation, so you run out of digits with numbers higher than 9,999,999,999 and you only have

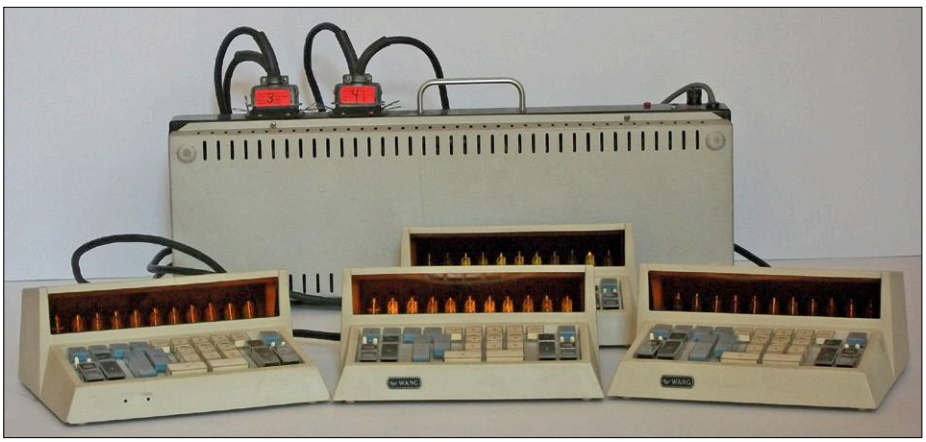


Figure 2: The Wang 320 System: four keyboard/display units and one 320SE Electronics Package.

one visible significant digit with numbers like 0.000000002. The 320 does its math in BCD (binary coded decimal) with four binary bits for each digit. This was common at the time, even though it used more bits than straight binary, as it was easier to interface BCD with a base-10 keyboard and display.

But it is "under the hood" where things get really interesting. The 320 is made entirely of discrete components; by my count, 832 transistors and a larger number of diodes, resistors, capacitors, etc. on a set of 32 circuit boards that Wang called "logic blocks" (**Figure 3**). The only calculating element is a single 4-bit BCD adder which is fine for addition and subtraction; however, more complex operations like multiplication cannot be executed in a single step with this simple hardware. Multiplication can be implemented as repeated addition; to multiply 45 × 12, just add 45 twelve times. This is simple and was used in some mechanical calculators of that period but it is extremely slow, especially when you want to multiply large numbers. It is possible to use a fully binary shift-and-add algorithm [1] that requires one addition per bit but this would incur the overhead of converting from BCD to binary and then back to BCD. Some calculators of that era used a combination of these two where "Multiplication is successive addition of the multiplicand as determined by each multiplier digit" going BCD digit by BCD digit rather than bit-by-bit [2].

The engineers at Wang took a less obvious but much more versatile approach. The 320 calculates the natural logarithm of the multiplier and then of the multiplicand, adds the logs together and takes the anti-log of the result — the digital equivalent of a slide rule! In the days before calculators were widely available, calculating with logarithms was a common way to multiply, divide, and more but it required a very large printed table of logarithms. In digital form, this would correspond to an enormous diode-matrix ROM. Modern computer and calculators either use a series expansion (that requires multiplication) or the CORDIC algorithm (which only requires shifting and adding). The Wang engineers managed to do this with a lookup table of the 15-digit natural logarithms of only 12 different numbers (10, 2, 0.9, 1.01, 0.999, 1.0001, 0.99999, 1.000001, 0.9999999, 1.00000001, 0.999999999 and 1.0000000001) and an extremely

ESTD 2004

www.elektor.tv

Retronics is a regular section covering vintage electronics including legendary Elektor designs.
Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

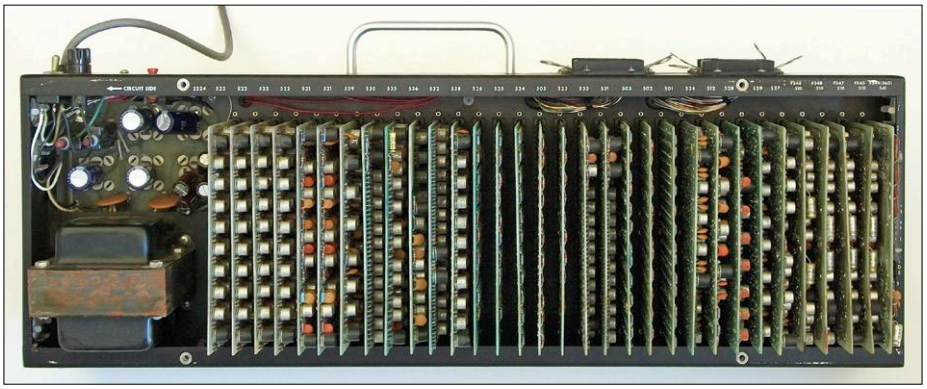


Figure 3: Inside the Electronics Package: 32 circuit boards, 832 transistors, and 1024 bits of core memory.

## Calculating a natural logarithm the Wang 320 way

The process requires two 16-digit BCD registers, a working register (W) for factoring $x$, and a log register (L) for accumulating ln($x$). To calculate ln($x$), you begin with $x$ in the W register. You then repeatedly multiply W by 0.9, each time subtracting ln(0.9) from L until W is just below 1.0; the original value of $x$ is lost but the process yields a rough approximation of ln($x$) in W. To get closer, you multiply what's left in W by 1.01 while subtracting ln(1.01) from the L until W goes just above 1.0. Next, you repeatedly multiply W by 0.999, etc. until you reduce W to as close to 1.000000 as you'd like and have accumulated ln($x$) in L.

But wait, why use strange numbers like 0.9, 1.01, etc. and how can you multiply a number by 0.9 if you can only add or subtract? It turns out that the answer to both of these is related and this is the cleverest part of the algorithm.

Since $0.9x = x - 0.1x$ and $0.1x$ is just $x$ shifted one digit to the right, you can calculate $0.9x$, $1.01x$ etc. by shifting one or more BCD digits and adding or subtracting — no multiplication required!

Suppose you wanted to find ln(1.2), which is 0.182.

| | **Working (W) register** | | | **Log (L) register** | |
|---|---|---|---|---|---|
| **Step** | **Action** | **Result** | | **Action** | **Result** |
| **1** | Start value | 1.2 | | start value | 0.0 |
| **2** | × 0.9 * | 1.08 | | subtract ln(0.9) | 0.105 |
| **3** | × 0.9 * | 0.972 | | subtract ln(0.9) | 0.211 |
| **4** | × 1.01 ** | 0.9817 | | subtract ln(1.01) | 0.201 |
| **5** | × 1.01 ** | 0.9915 | | subtract ln(1.01) | 0.191 |
| **6** | × 1.01 ** | 1.0015 | | subtract ln(1.01) | 0.181 |
| **7** | × 0.999 *** | 1.0004 | | subtract ln(0.999) | 0.182 |

As you can see, as you use constants that get ever closer to 1, W gets ever closer to 1, and L gets ever closer to ln(x).

* Shift one place to the right and subtract.
** Shift two places to the right and add.
*** Shift 3 places to the right and subtract.

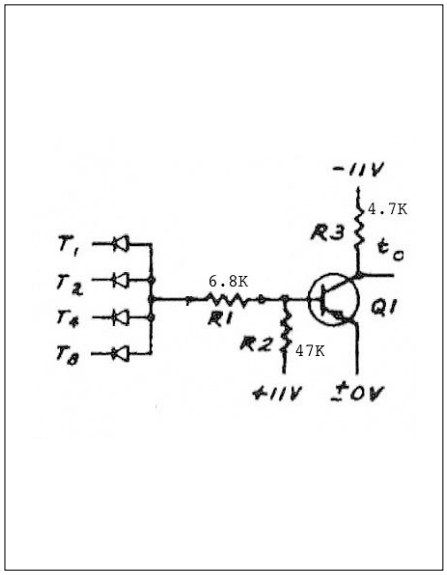



Figure 4: One of the 3-input NOR (inverting-OR) gates in the Electronics Package. The output at the right — "t0" — is logic 1 when T1, T2, T4, and T8 are all logic 0. Image scanned from equipment service manual.

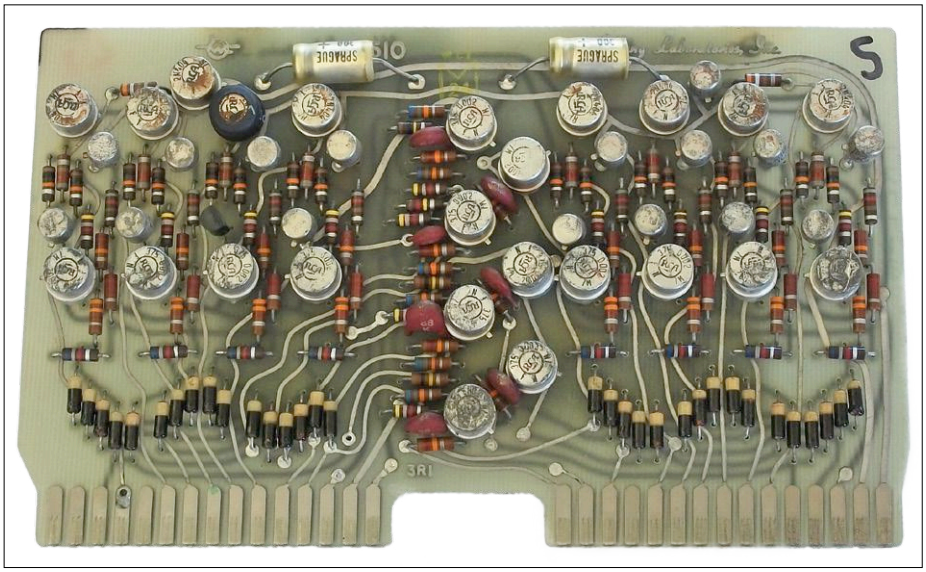


Figure 5: The culprit — the defective core driver board. Note how close the transistors are to each other. The transistor that restored the function of the 6th digit is easily identified by its plastic enclosure.

clever algorithm (see **sidebar** for details). Not only does this get you multiplication and division, but taking the log, exponentiation, squares and square roots are then trivial to build in.

It still takes a while: with a master clock rate of 20 kHz, an addition takes about 23 ms and calculating a logarithm about 200 ms. As a result, a full multiplication after you have entered the first number and its log has been taken — calculate logarithm, add logarithms, calculate anti-log — takes 425 ms, enough for a perceptible delay. At a price of $1,282.50 in 1968 (~$9,200 in 2017 dollars), it wasn't cheap, but, remember, it was four fully-functional calculators for that price.

## Bringing it back to life

Sadly, back in my home lab, it didn't work. The nixies lit up and the numbers displayed changed when I pressed various keys, but it clearly wasn't calculating properly. Figuring out why took some investigation. I have made a website that describes the diagnosis and repair that followed in detail including links to schematics and other information for the curious as well as those wanting to work with calculators of this series [4]; here, I will describe some of the highlights.

The first step was to find out what the logic levels were. Since the logic was implemented as discrete components, the Wang calculator was not constrained to follow the emerging 5 V logic standard of TTL ICs. A typical gate circuit from Wang's schematic is shown in **Figure 4**. Since transistors in logic circuits are typically either fully off or fully on, this showed that the logic levels would be roughly –11 V and 0 V — certainly oddball levels for 2017! Why on earth did the Wang engineers use negative voltages and PNP transistors rather than more intuitive positive voltages and NPN transistors? The details of this choice are lost to the mists of history, but Wang used germanium transistors instead of silicon in those days and I've read some suggestions that, back then, germanium PNP transistors were cheaper than germanium NPNs (if anyone knows more, please let me know). However, knowing the voltage levels does not show which is logic 1 and which is logic 0 and it turns out that you can't tell this from a schematic alone (that's a logic puzzle for another day).

Fortunately, there were clues in the schematic like the part shown in Figure 4. This circuit is found on a board with the 4-bit T-counter; the patent for which [5] showed how particular values of the T-counter corresponded to particular steps in the calculation process. The output of this circuit is called "t0", so it is reasonable to suppose that it is logic 1 if and only if the T-counter's count is 0; that is, when the T-counter bits are all 0. As a result, this must be a NOR gate. If you look carefully at the circuit or simulate it, it will only acts as a NOR gate if –11 V is "logic 1" and 0 V is "logic 0". Try it and see; with the voltages reversed, it's an AND gate, and the output should have been labeled "t15".

To be sure, this was only a "working hypothesis" and my experience as a biologist has taught me that nature likes to trick the over-confident; the best test of this hypothesis would be to use it to design an interface to test the keyboard/display units. To do this, I built a board with BJT level shifters and LEDs to monitor the keyboard outputs and a PIC16F1823 microcontroller driving optoisolators to supply the display control inputs. The interface worked, verifying the logic levels, and I was able to use it to diagnose and repair some broken key switches and a few failed transistors in the four keyboard/display units. Although I could not find equivalents for the RCA transistors Wang used, their specifications are not critical as they're functioning as on/off switches and I found that ordinary 2N3904 and 2N3906 general-purpose silicon transistors worked just fine for all but the high-voltage nixie drivers — for those, I used ZTX458's.

## One more step to glory

Unfortunately, even with all the keyboards working, the 320 still wasn't calculating right. To diagnose the Electronics Package, I built a 16-channel level-shifter to convert the 0 V/–11 V levels

to 0/+3.3 V levels that my logic analyzer could recognize. I started by monitoring the core memory to determine which locations were being accessed and what was being read or written. Fortunately, the patent on the 300-series calculators [5] gives detailed flowcharts of the calculator's activities, and I was able to follow along. I also noticed a pattern: something was wrong with the 6th digit from the left in all of the keyboard units. Although I could clear the 6th place to a "0" with all the other digits, if I tried to enter the number 123456, I would get to "12345" but no matter what I tried to type next, nothing would appear in the 6th place. If I then added the 12,345 to one of the accumulators, the first five digits added fine but there were random added digits at the 6th and higher places. Finally, if I tried any operations that involved taking a logarithm, for example squaring or multiplying, it would go into "la la land" with the display showing ever-changing random numbers until I pressed the "Prime" (reset?) button on the Electronics Package.

In the 320, these numbers are stored in the core memory, which is arranged as 16 registers, each consisting of 16 4-bit BCD numbers (14 digits, a sign digit and the decimal point position). Each keyboard/display unit gets its own set of four registers: working register, log register, and two accumulators. Physically, the core is four 16×16 arrays of little magnetic cores, with each of the four arrays corresponding to one of the four bits in the BCD digit. The 16×16 arrays are addressed by a grid of X- and Y- core driver lines. By passing just the right amount of current through one X and one Y line, the core (a tiny magnetizable ferrite donut) at the intersection of those lines will "see" the magnetic field of both wires and be magnetized — while the other cores on those X and Y lines will see a magnetic field only half as strong and will not be magnetized (for more details on core memory, see [6]).

The 16 X-selection lines choose which of the 16 registers is being accessed; the 16 Y-selection lines choose the digit within that register. The X and Y selection lines are shared across all four arrays so all four bits of the BCD digit are accessed simultaneously. Since the 6th digit was a problem in all registers of all four keyboard/display units, that suggested that there was something wrong with the Y-driver circuitry for that particular column in the core.

Because of the way the cores were wired, it was not possible to test the core driver boards *in situ*. I used another trusty PIC16F1823 and some BJT level shifters to build a test bed for the driver boards and tested them one-by-one. Lo and behold, one driver was bad — the one for the 6th digit. It turned out that two TO-5 style transistors were very close together and when their metal cases — connected to their collectors — touched, it fried the driving transistor. You can see how close the transistors are in **Figure 5**. A plain vanilla 2N3904 replaced the blown driver and eventually I put heat-shrink tubing around the other transistors as a precaution.

And…. that did the trick. It now works and calculates happily. The nixie tubes are refreshed very slowly so there's a significantly visible flicker but that warm orange glow is a pleasure to look at. You can see it "thinking" when it takes the logarithm of a number; as it reduces the working register to 1, you can see the digits change as it converges on the solution. You can sometimes catch the last step in a calculation as the decimal point bounces around to its final location. I've posted a video on my site [4] so you can see it in action. There's just one fly in the ointment — the Overflow light doesn't light when it should. But that's a task for another day…. ⏮

(160531)




The 1970s … small business accounting in community style! Seriously, Wang Laboratories under Dr An Wang's leadership was a bold company in many ways, constantly pestering and challenging IBM through adverts in the US press as well as on TV. Wang Labs pioneered word processing as early as in 1971, and their 2200 and 3300 office automation systems were hugely successful. Images: wang3300.org; AP/Townson.

**Web Links**


[1] https://en.wikipedia.org/wiki/Multiplication_algorithm#Peasant_or_binary_multiplication

[2] Hewlett Packard Journal September 1968 page 14.

[3] US Patent 3402285, https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US3402285.pdf

[4] https://brianwhite94.wixsite.com/electronics/wang-320-calculator

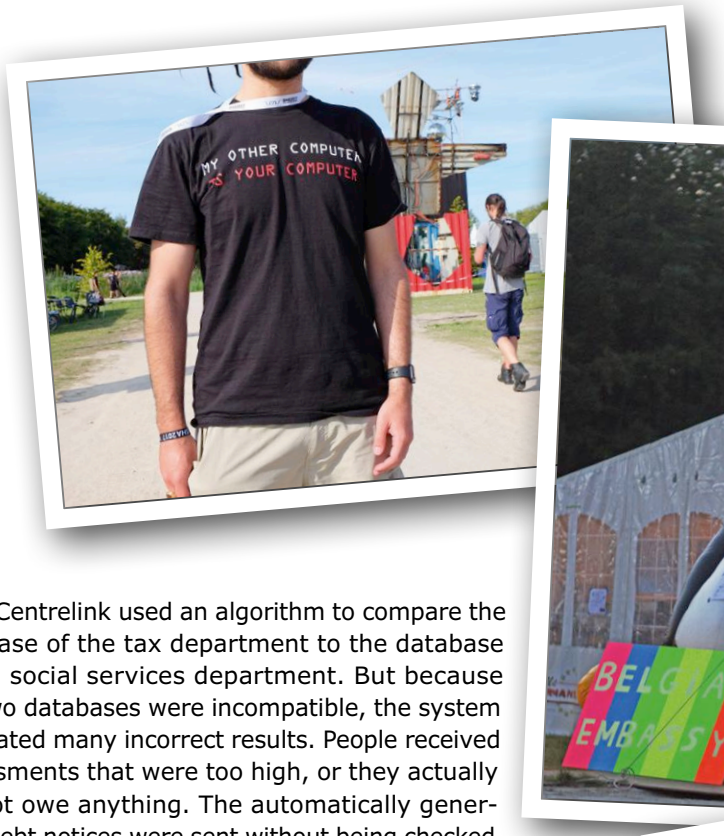
[5] US Patent 3509329, https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US3509329.pdf

(6) https://en.wikipedia.org/wiki/Magnetic-core_memory

ELEKTOR ETHICS

# Why We Need Hackers

By **Tessel Renzenbrink** (The Netherlands)




Over 3500 hackers recently descended on the countryside in the Dutch province of Flevoland for the five-day Still Hacking Away (SHA) 2017 outdoor hacker camp. Hacking has a negative image nowadays, with the term often being used as a synonym for cyber-criminality. However, the original meaning of hacking is "using innovation to create a new application for an existing system." This innovative spirit was visible everywhere during the SHA event. From the 4th to the 8th of August, the green acres of Zeewolde Scout Camp were converted into a tent village, which had a mystical appearance at night due to the many thousands of LEDS, fire-spewing towers, and motorized sofas that bowled you over.

The hackers naturally brought their own Internet along. Dark fibers (unused fiber optic cables) were temporarily lit up to give the tent village an impressive 100 Gbit/s uplink. An electrical system was also rolled out to power the computers, robots, telex machines, drones, LEDs, 3D printers, soldering irons, and the sauna of the Finnish hackers. In addition there was an alt-power field running on sustainable energy. That was an experiment intended to pave the way for energy autonomy in future camps.

**Made by angels**

SHA was completely organized and run by angels — hacker camp jargon for volunteers. The people who spent two years making the SHA a reality did all that for free. The bar angel who sold the drinks and the trash angel who emptied the wastebaskets also paid for their tickets. At SHA there are no visitors, only participants.

The current urgent need for the enthusiasm, commitment and inventiveness which characterize hackers was clearly demonstrated by several of the presentations held on the various podiums. They described technological systems that were rolled out without careful consideration of the consequences. In many cases the impact on people subjected to those systems was enormous. Take the story of Lyndsey Jackson, for example. She flew in from Australia to tell how the government there unleashed an algorithm on its citizens and left them to their fate.

**Governed by algorithms**

Shortly before Christmas in 2016, thousands of Australians received letters which informed them that they owed money to Centrelink, the Australian social services department. These letters were the result of an automated process. Centrelink used an algorithm to compare the database of the tax department to the database of the social services department. But because the two databases were incompatible, the system generated many incorrect results. People received assessments that were too high, or they actually did not owe anything. The automatically generated debt notices were sent without being checked by humans. The affected citizens were confronted with the onerous task of proving to the government that they did not owe anything.

Jackson launched the online campaign #notmydebt to create national attention for the problem. As a result of publicizing the issue, the affected persons realized that they were not alone. The campaign brought people out of their shells and forged a movement that resists being governed by algorithms.

**Gambling with the electricity supply**

Another story about a system fault that can have major consequences was told by Willem Westerhof, a security engineer at the security company ITsec. He was in the news in August when he claimed that security vulnerabilities in solar power systems could lead to pan-European blackouts. At SHA he told how he had hacked the inverters produced by market leader SMA and was able to remotely switch the systems on and off. If someone with malicious intent were to do that on a large scale, the stability of the power grid could be endangered. Revelation of Westerhof's findings sparked a public discussion with grid managers and politicians. Westerhof: "They ask questions like 'Can an attacker actually compromise so many devices?' and 'Would that really bring down the grid?' I think those are strange reactions, because the question we should be asking is 'Why are we connecting unsafe devices to the power



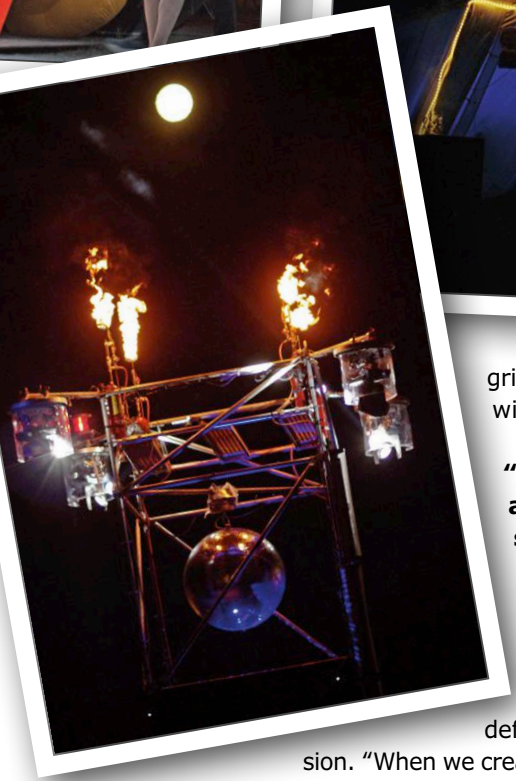












grid?'" That statement met with loud applause.

**"We shape our tools, and thereafter they shape us"**

"Technology is not neutral," says Niels ten Oever from Article 19, a British human rights organization that defends freedom of expression. "When we create technologies, we consciously and unconsciously embed our own thoughts, values and world view in them. That is not special for technology; we also do that in architecture. If we make broad streets with no sidewalks, only for cars, what will happen? More cars. If we create large spaces with benches and trees, people will be outside." He displayed a quote from John Culkin on the large plasma screen: "We shape our tools, and thereafter they shape us." That is why Ten Oever is working to incorporate human rights in the Intranet infrastructure. For example, the right to privacy can be better protected at the technical level by using a protocol that encrypts data in transit instead of sending data fully exposed over the Internet. "We have to take human rights into account as early as the design phase," says Ten Oever.

**Hacker ethos**

At a time when technology is rapidly changing our society, we urgently need the hacker ethos. So we can ask the right questions, like Westerhof. So we can incorporate human rights in technology, like Ten Oever. So we can resist "the computer says no," like Jackson. Hackers show us that we do not have to accept systems as externally imposed, rigid structures that determine how we act, but instead use our innovative skills to find other applications. To take ownership of the systems and shape them, rather than letting them shape us. ⏮

(160535-I)

## Web Links

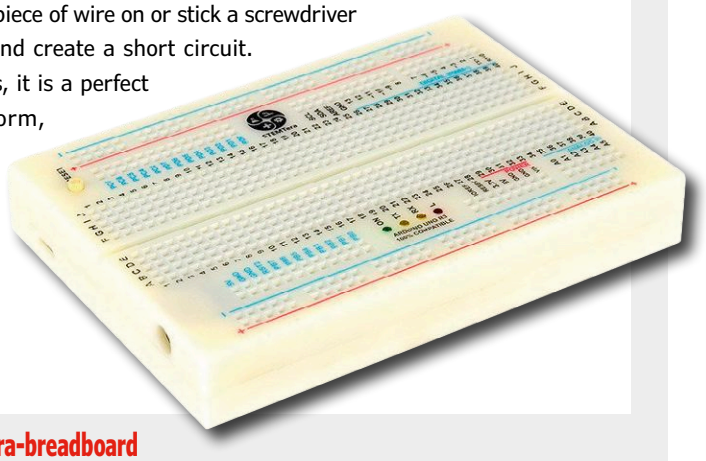
All presentations mentioned here were filmed by video angels and are available online at https://media.ccc.de/b/conferences/sha2017.

# welcome in your ONLINE STORE

## EDITOR'S CHOICE



When I was asked to review the STEMTera I had absolutely no idea what it was. But when I opened the box, the product's purpose became immediately clear and I loved the idea. In short, STEMTera is a 40-column breadboard with an Arduino Uno R3 built in. Put the other way around, it is an Arduino Uno R3 built into a strong enclosure that doubles as a breadboard. This clearly is the strong point of the STEMTera: it provides a robust way for tinkering with Arduino. It is sturdy and will not break easily when it drops on the floor, and it is impossible to accidentally drop a piece of wire on or stick a screwdriver in the running Arduino and create a short circuit. Available in several colors, it is a perfect Arduino teaching platform, but also great for people who (try to) use Arduino in Lego projects thanks to its Lego-compatible bottom side.



**Clemens Valens**
Elektor Labs

**www.elektor.com/stemtera-breadboard**

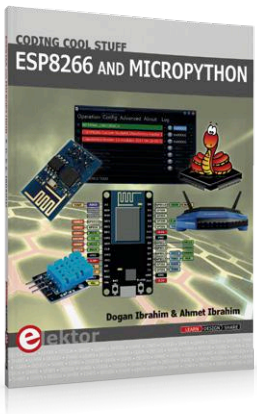
## Elektor Bestsellers

1. Python 3 Programming and GUIs
   www.elektor.com/python3




2. ESP8266 and MicroPython
   www.elektor.com/esp8266-book
3. HDMI Digital Microscope ADSM201
   www.elektor.com/adsm201
4. Raspberry Pi 3 (Model B)
   www.elektor.com/rpi3
5. TL866A Universal Programmer
   www.elektor.com/tl866a
6. Sand Clock
   www.elektor.com/sandclock
7. GSM/GPRS Projects
   www.elektor.com/gsm-gprs-projects

### ESP8266 and MicroPython

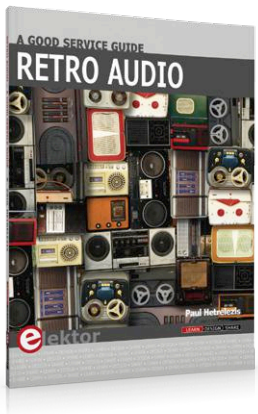



This book is an introduction to the ESP8266 chip and describes the features of this chip and shows how various firmware and programming languages such as the MicroPython can be uploaded to the chip. The main aim of the book is to teach the readers how to use the MicroPython programming language on ESP8266 based hardware, especially on the NodeMCU.

member price: £24.95 • €26.96 • US $33

**www.elektor.com/esp8266-book**

### Retro Audio

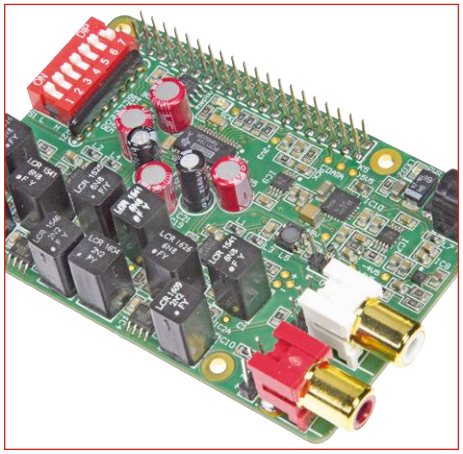



Today there is a re-emerging, nostalgic interest in vinyl records and associated music entertainment gear. With this interest, there is a paralleled market for the repair of this gear. The intention of this book is to offer the reader understandings, ideas and solutions from the perspective of a workbench technician and electronics hobbyist. It is a descriptive text with many tables of useful data, servicing tips and supplementary notes of not so common knowledge.

member price: £24.95 • €26.95 • US $33

**www.elektor.com/retro-audio**

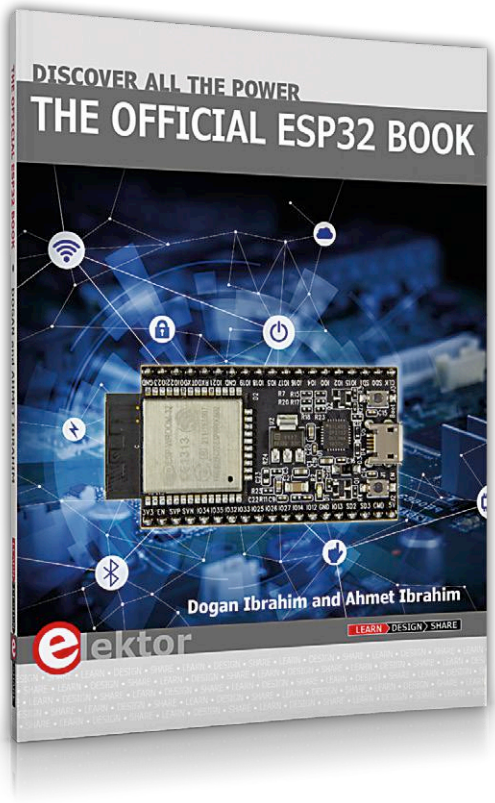
### Audio DAC for Raspberry Pi



Small single-board computers like the Raspberry Pi are ideal for use as small stand-alone network audio players. With Volumio and other specialized Linux distributions, putting together an audio player is remarkably easy. What's more, you can also us this approach to build a high-end audio player! The module with LCD touchscreen for RPi is now available in our store (RPi not included)!

member price: £141.95 • €152.96 • US $182

**www.elektor.com/rpi-audio-dac**




NEW

# The Official ESP32 Book

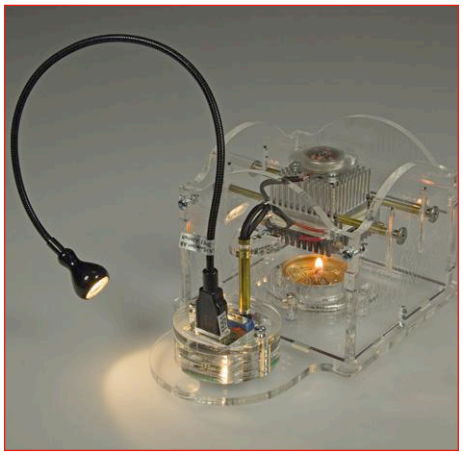
This book is an introduction to the ESP32 processor and describes the main hardware and software features of this chip. The aim of the book is to teach the reader how to use the ESP32 hardware and software in practical projects, especially using the highly popular ESP32 development board. Many basic, simple, and intermediate level projects are given in the book based on the ESP32 DevKitC development board, using the highly popular Arduino IDE and also the MicroPython programming language.

*A ESP32 hardware kit is available from Elektor, especially prepared for the book. The kit contains all the components used in the projects in this book.*

**MEMBER PRICE: £26.95 • €29.95 • US $36**

**www.elektor.com/esp32-book**

## Peltier Lamp



This lamp uses a Peltier thermoelectric module to convert heat from a candle flame into electricity. A Peltier module is a sort of semiconductor heat pump which uses electricity to transports heat from one side to the other, or generates electricity from heat transport through the module. If you heat a Peltier module on one side (for example, with a candle) and cool it on the other side (for example, with a fan), it will generate electricity.

member price: £99.95 • €107.96 • US $129

**www.elektor.com/peltier-lamp-1**

## Home Automation Projects with Arduino




If you really want to build some innovative projects it's often necessary to get down to component level. This book explains how a wide variety of practical projects can be built using items supplied in a single kit together with the Arduino board. This kit, called the 'RFID Starter Kit for Arduino UNO' is not just limited to RFID applications but contains more than 30 components, devices and modules covering all areas of modern electronics.

member price: £27.95 • €29.66 • US $36

**www.elektor.com/arduino-rfid-book**

## 10" Touch-Display for Raspberry Pi



This 10" Touch-Display is housed in a sleek metal case making it the perfect choice for a wide range of applications. Included with the display is an adapter plate which can be used to mount any of the standard single board computers on the rear of the unit. The USB socket can power the display when it's used with an SBC. The display can also be used as a monitor for standard PC systems via the HDMI and VGA connectors. A BNC and AV connector provide analog inputs for CCTV etc.

member price: £120.95 • €130.45 • US $156

**www.elektor.com/touch-display-for-rpi**

# Hexadoku The Original Elektorized Sudoku

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 book 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 Book Vouchers worth **$70.00 / £40.00 / €50.00 each**, which should encourage all Elektor readers to participate.

## Participate!

**Ultimately November 24, 2017**, 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 5/2017 (September & October) is: **C3E61**.

The €50 / £40 / $70 book vouchers have been awarded to: Eugene Stemple (USA); Han Welmer (Netherlands); Thomas Lienhard (Switzerland); Gregorio Ballini (Italy); Emil Cugini (Switzerland). .

**Congratulations everyone!**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | 4 | | | | | | | | | D | F | 3 | 7 | E |
| | | | 3 | | 7 | 1 | | | | | | | | | 9 |
| | | | E | | D | | 2 | 1 | F | 6 | 3 | | 4 | | 5 |
| | | | 2 | | 3 | E | 5 | | | | B | C | | | 6 |
| F | | | | C | | D | 6 | | | 5 | | 7 | 9 | | 4 |
| 9 | | A | | 5 | F | 0 | | B | C | | E | | 1 | | |
| | | 2 | | | | | | | | 3 | | | E | | |
| | 5 | 6 | | | 8 | B | E | | | 1 | | | A | | |
| | | 7 | | | | | 4 | 9 | | | A | E | 8 | | |
| D | A | | | | | | | 0 | | 2 | 5 | 1 | | 4 | |
| | E | | 9 | 8 | A | | | 6 | | B | | 2 | 5 | F | |
| | 4 | | | 6 | E | | | | | F | 8 | | | | |
| | 0 | 9 | | | 5 | | | | | | | D | F | 8 | |
| | | D | C | | | | F | 3 | 5 | 0 | | | | | 7 |
| 4 | | | F | B | 2 | C | | A | | | | | | | |
| 6 | 7 | | | | | 4 | | | | D | F | | | | 1 |

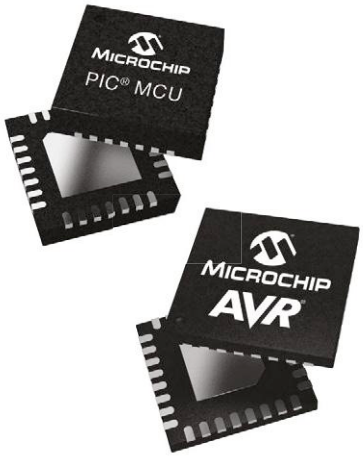
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | E | 2 | 8 | C | 4 | D | 9 | 0 | 1 | 5 | B | 3 | 6 | 7 | F |
| 7 | B | 6 | 3 | 5 | 8 | E | F | 9 | C | 4 | 2 | A | D | 0 | 1 |
| C | F | 4 | 5 | 0 | 1 | 7 | A | 3 | 6 | D | 8 | 9 | 2 | B | E |
| 0 | D | 9 | 1 | 3 | 6 | B | 2 | 7 | E | F | A | 5 | C | 4 | 8 |
| B | 0 | A | 7 | 9 | E | 2 | 1 | 4 | 5 | 3 | F | 6 | 8 | C | D |
| D | 8 | C | 2 | A | F | 3 | 0 | 6 | 7 | 9 | 1 | B | 5 | E | 4 |
| F | 5 | 1 | 9 | 6 | B | 4 | D | E | 8 | A | C | 7 | 3 | 2 | 0 |
| E | 4 | 3 | 6 | 7 | 5 | 8 | C | B | 2 | 0 | D | F | 1 | 9 | A |
| 9 | C | E | F | 8 | A | 5 | 7 | D | B | 6 | 3 | 0 | 4 | 1 | 2 |
| 2 | 7 | 5 | D | B | 0 | 6 | E | F | A | 1 | 4 | C | 9 | 8 | 3 |
| 3 | A | 8 | B | 1 | D | 9 | 4 | 2 | 0 | C | 7 | E | F | 5 | 6 |
| 6 | 1 | 0 | 4 | F | 2 | C | 3 | 5 | 9 | 8 | E | D | 7 | A | B |
| 1 | 3 | F | E | D | 7 | 0 | 5 | 8 | 4 | B | 9 | 2 | A | 6 | C |
| 8 | 9 | 7 | C | E | 3 | 1 | 6 | A | D | 2 | 0 | 4 | B | F | 5 |
| 5 | 2 | B | A | 4 | 9 | F | 8 | C | 3 | E | 6 | 1 | 0 | D | 7 |
| 4 | 6 | D | 0 | 2 | C | A | B | 1 | F | 7 | 5 | 8 | E | 3 | 9 |

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.

# PIC® & AVR® MCUs

## Together Your Possibilities are Unlimited

You have a desire to make technology smarter, more efficient and accessible to everyone. Microchip has a passion for developing products and tools that make it easier for you to solve your design problems and adapt to future needs. Microchip's portfolio of more than 1,200 8-bit PIC® and AVR® microcontrollers is not only the industry's largest—it incorporates the latest technologies to enhance system performance while reducing power consumption and development time. With 45 years of combined experience developing commercially available and cost-effective MCUs, Microchip is the supplier of choice due to its strong legacy and history in innovation.




**Key Features**

- Autonomous peripherals
- Low-power performance
- Industry-leading robustness
- Easy development

microchip DIRECT
www.microchipdirect.com

MICROCHIP

www.microchip.com/8bitEU

The Microchip name and logo, the Microchip logo, PIC and AVR are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2016 Microchip Technology Inc. All rights reserved. DS30010130A. MEC2116Eng09/16

# DO YOU WANT THE BEST ELECTRONICS DESIGN SOFTWARE




**FEATURES**

- Schematic Capture
- PCB Layout
- Gridless Autorouting
- 3D Visualization
- M-CAD Integration
- SPICE Simulation
- MCU Co-simulation
- Built in IDE
- Visual Programming

**NOW INCLUDES:**
Serpentine Routing, Layer Stackup Manager, and Assembly Variants.

Labcenter Electronics

www.labcenter.com

(+44) 01756 753440