Edition 2/2018 | March & April 2018
www.elektormagazine.com

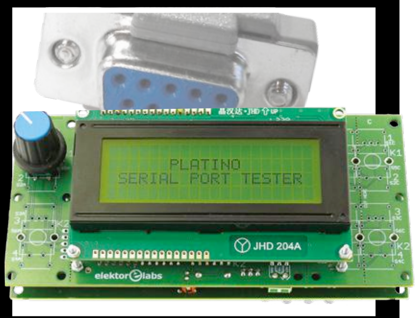
elektor

LEARN DESIGN SHARE

ESP32 Design Contest 2018

€2500* in prizes
* or equivalent in £ / US$

submit your application and win!



Platino "Fits-All" Serial Bus Tester
Debug serial comms issues with ease

Electromagnetic Interference from LED Lamps
Fourteen examples compared

RF Power Meter
For measurements up to 10 GHz

BASIC and the Embedded World • Chimes for the Elektor Sand Clock • **A Cloud in the Shape of a Raspberry** • Current Transformer for Oscilloscopes • **DCF77 Emulator with ESP8266** • Electromagnetic Interference from LED Lamps • **Elektor Store** • Elektor Video Olympics • **Engraving Machines under Test** • Err-lectronics • **ESP32 Design Contest 2018** • ESP32 Low Power • **From Russia with Love** • Hexadoku • **Homelab Helicopter** • Internet Censorship during the Catalan Referendum • **KiCAD** • Measurement Data Acquisition via USB • **Multi-Timer** • Platino "Fits-All" Serial Bus Tester • **Powerbank Surprise** • Q & A: Nixie tubes • **Remote Water Level Meter** • Retronics: HP650A Test Oscillator (ca. 1948) • **RF Power Meter** • A Simple Digital Audio Amplifier • **Soft Start for PSU** • Supercaps • **Tektronix Readout System** • Temperature Logger Hack • **Timers for the Wi-Fi Desktop Thermostat** • Universal USB Charge Controller Chip • **... and more**

Elektor Magazine

Edition 2/2018
Volume 44, No. 488
March & April 2018

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

www.elektor.com
www.elektormagazine.com

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

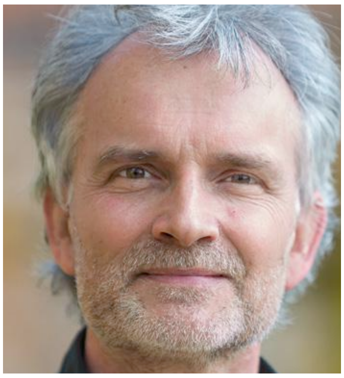
Elektor International Media
78 York Street
London W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344

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

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



Solid advice

Trade shows like *embedded world* and *electronica* are rare opportunities for me to have a chat with an Elektor reader. Provided the talk is technical, such conversations tend to attract other participants and it is gratifying to see engineers at ease with exchanging tech information as well as respecting mutual viewpoints and preferences. By contrast, especially younger electronics enthusiasts appear to be in danger of becoming socially isolated because so many things can be done they believe with just a laptop and a pizza as companions in their quest for quick & cheap circuit design and YouTube glory.

Now although the laptop is a great tool and an absolutely essential investment, in terms of conversation it will only beep at you occasionally, or produce blue screens and arcane error codes. Electronics is learned from manuals, publications, components trays and people. Give it a try, with some exceptions you will find people friendlier, more forgiving and inspiring than the average laptop on WiFi and with an array of CAD tools installed.

Another great place to learn about electronics in a very practical way seems to have disappeared completely, at least in the Western world: the electronic parts retail shop, where free classes were taught in compotalk, like *trannies*, *O/P* and *I/P*, *duds*, *caps*, *HT* (high tension), *DOA* (dead on arrival), *toasted*, *unobtanium, geranium,* and *the plans*. Ironically some of these terms still pop up in newsgroup messages where they add decorum to the writer. In the past, in the local electronics store, it was not exceptional for shoppers waiting to be served to exchange tech advice or plain gossip like projects from a certain magazine that were “never any good”, or “sure that’s a direct equivalent — just a bit larger”.

Despite the much larger volume of tech talk on electronics going on these days in newsgroups and forums, from the tone of some of the messages there I get the impression that the contributors lack the spirit of those that enabled electronics as a pastime to coexist cheerfully with all things professional.

Jan Buiting, Editor-in-Chief

This Edition

Volume 44 – Edition 2/2018

No. 488 March & April 2018

Regulars

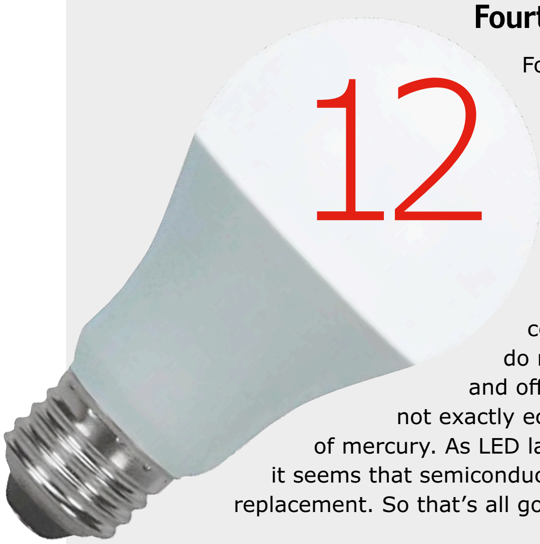
Features

A Cloud in the Shape of a Raspberry

install a cloud in your living room

Electromagnetic Interference from LED Lamps

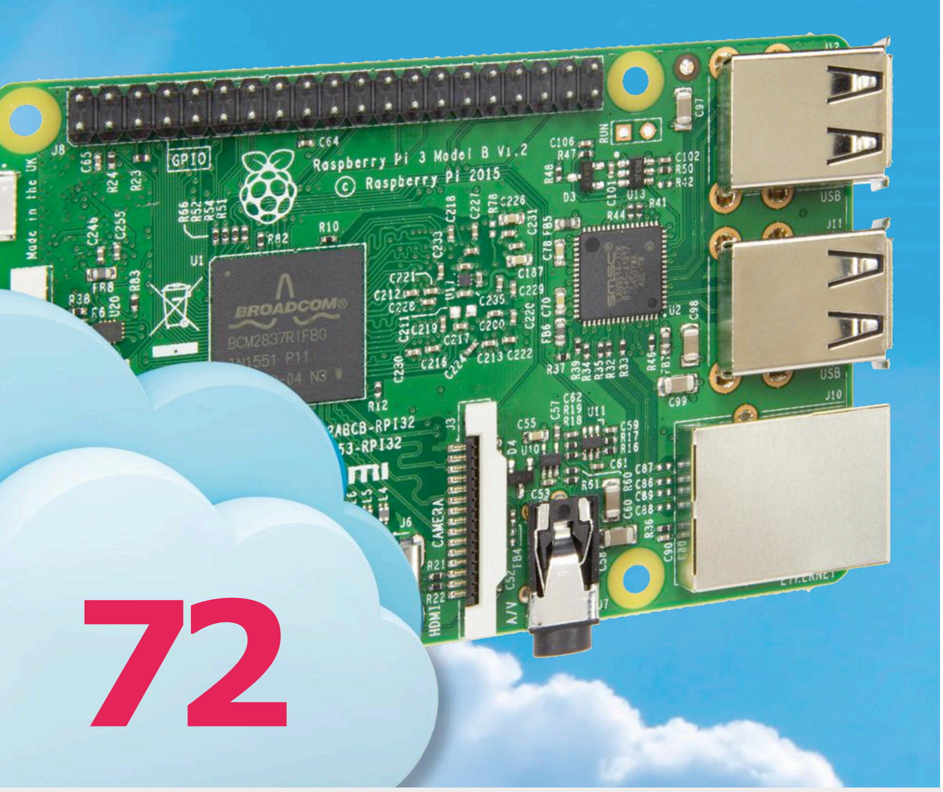
Fourteen examples compared



12

For more than a hundred years 'artificial light' has almost exclusively meant incandescent lamps. However, the relatively low output of incandescent lamps has been criticized and has ultimately led to their being banned in the EU. Low-energy fluorescent lamps are certainly more efficient, but they do not work well when turned on and off frequently, and they contain not exactly ecologically-friendly quantities of mercury. As LED lamps become cheaper to buy it seems that semiconductor-based lighting is the ideal replacement. So that's all good then. Or is it?

Projects



72

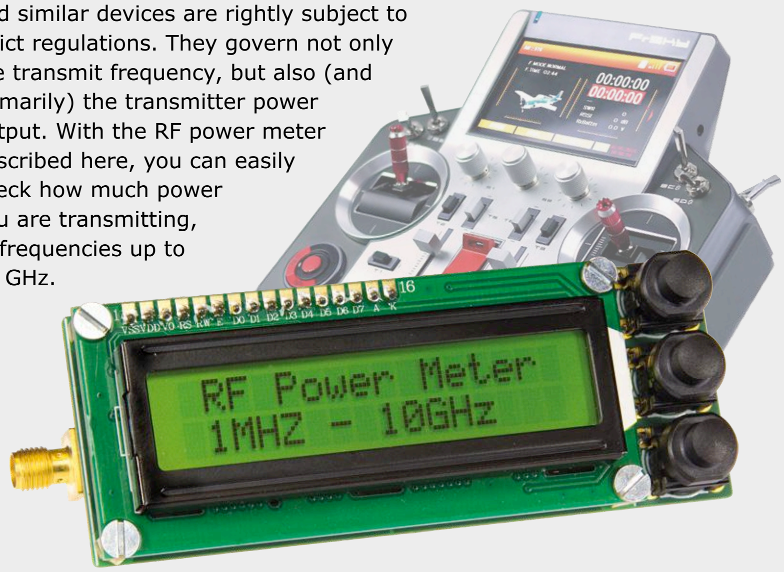
elektor magazine

RF Power Meter

42

For measurements up to 10 GHz

Radio remote controls for model aircraft and similar devices are rightly subject to strict regulations. They govern not only the transmit frequency, but also (and primarily) the transmitter power output. With the RF power meter described here, you can easily check how much power you are transmitting, at frequencies up to 10 GHz.

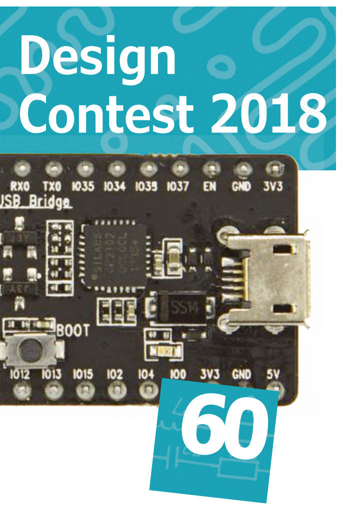


Next Editions

Elektor Magazine 3/2018

Cross-Platform Basic-Compilers • LoRa Telemetry Projects • Floranium • Interference from LED Strings • DAB/DAB+ Radio with RTL-SDR (RTL2832U) • Analogue alternating linear LED Fader • Universal I2C Bus Isolator and Level Adapter • Audio DAC Update • GPS-based 10MHz Reference • CPLD DIL Board • Remembering the SN76477.

Elektor Magazine edition 3 / 2018 covering May and June is published around 12 April 2018. Delivery of printed copies to Elektor Gold Members is subject to transport. Contents and article titles subject to change.



Elektor Business Edition 2/2018

Elektor Business Edition issue 2 / 2018 has a focus on **Industry 4.0 and Automation**. Among the contributors are companies and research institutions including Microchip, Mathworks, Atollic, AEG Power Solutions, Unitronic, LoRa Alliance, SecureRF, OSRAM, and Avnet Silica. Plus you'll find fresh instalments of all the EBM regulars like Infographics, Operation Marketing, Our Business, and Business Store.

Elektor Business Edition issue 2 / 2018 is published on 11 April 2018 to Elektor Gold members in print, and Elektor Green members as a pdf download. The edition is also available for purchase at www.elektormagazine.com.

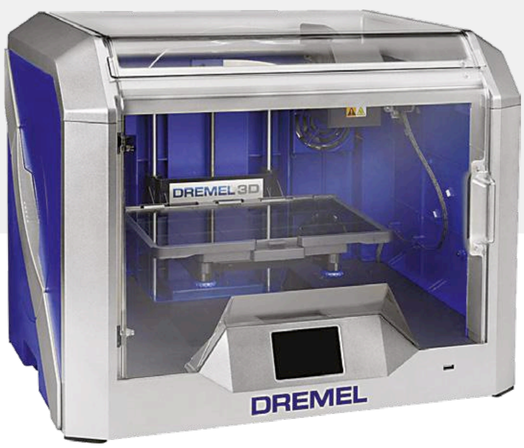
Elektor Video Olympics

In the September & October 2017 issue of Elektor we announced, with high expectations, the Elektor Video Olympics — and challenged all Elektorians to make a video about an Elektor project and send it in. To make it especially attractive for participants we also organised a number of spectacular prizes...

Now, were we surprised! We received a total of, count them, an unbelievable four (say, four!) entries, one of which we had to disqualify immediately (see below). Apparently Elektor readers rather spend time with a soldering iron than with a video camera.
Considering the low response rate it was decided to only award the first (and biggest) three from the available prizes – it was originally the intention of the jury to make their selection from those fifteen entries that received the most 'likes'. After considerable deliberation the Jury has decided to award the prizes as follows:

Grand prize

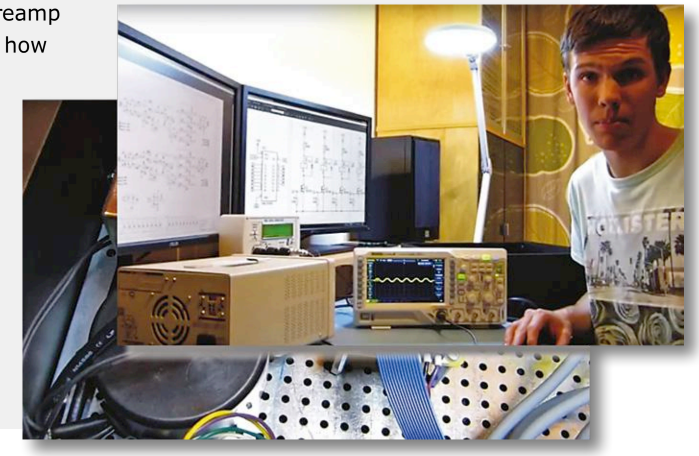
The entry 'Direct Current Traction — Elektor Teslameter on Traction' from **Mark van Helvoort** (Netherlands) has been viewed more than 1000 times. In this video Mark demonstrates that the Dutch trains, which run on direct current, can cause interference in other circuits and devices. He used the Elektor Teslameter for this, which was published in the January 1997 issue of Elektor. By the way, the Grand Prize is a Dremel 3D Idea Builder 3D40, a fancy 3D-printer with a retail value of €1449.



Grand prize: https://youtu.be/ZWnuH0qdly0

First prize

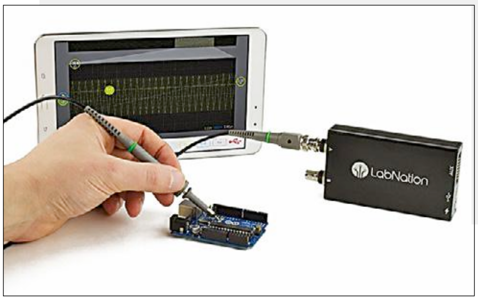
The entry from **Alexander Becker** (Austria) about his Elektor Preamp has been viewed more than 550 times. In this video Mark shows how he built his high-end Elektor preamplifier (of which the first instalment was published in the April 2012 issue of Elektor) and provided it with a few useful additions. The Jury was particularly impressed with the exemplary manner in which the preamplifier was built. The first prize is a Picoscope 2208B 100-MHz oscilloscope with a retail value of €1099.

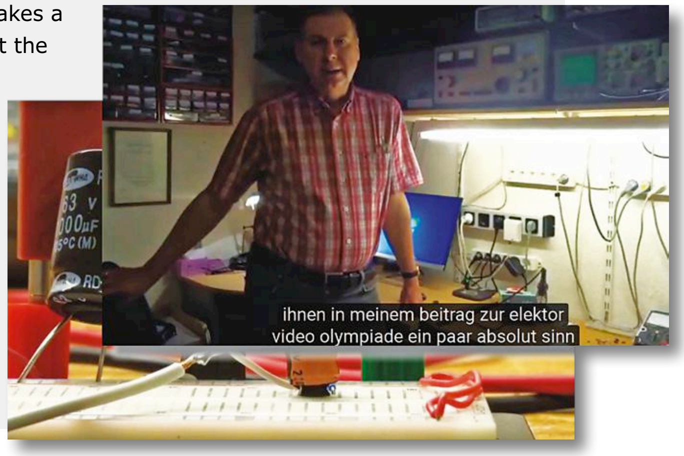


First prize: https://youtu.be/3TOsPorZKx4

Second prize

In his video 'Elektor Scherzartikel' **Rainer Schuster** (Germany) takes a closer look at a few of the famous (or infamous) 'joke circuits' that the Elektor editors used in the seventies and eighties of the previous century to liven-up the annual Summer Circuits editions. The entry from Rainer has been viewed more than 400 times. Especially our *Retronics* editor Jan Buiting was enamoured by this entry. And for those that do not recall: the second prize is a LabNation SmartScope Maker Kit with a retail value of €300.





Second prize: https://youtu.be/2f75lK12Wjo

Ineligible

Employees of Elektor and their families are obviously excluded from participating. Nevertheless our lab guru **Clemens Valens** made a video which was mainly intended as inspiration for potential participants, and in which Darth Vader (*of all people*) praises the virtues of the MicroTesla Music Synthesizer (from the January/February 2018 issue) – complete with music and titles à la *Star Wars*.

This (otherwise very good) Darth Vader impersonation was disqualified — https://youtu.be/I2Lc8gIPMWk

All employees from Elektor wish the winners much enjoyment with their prizes!
No correspondence will be entered into about the outcome. ◄

(160562)

HOMELAB PROJECT

Universal USB Charge Controller Chip

No microcontroller, just 6 pins

By **Tam Hanna** (Slovakia)

When designing systems with microcontrollers, SoCs, LCDs and touch screens it's often useful to incorporate a cheap Tablet or smartphone. In this sort of setup you need to think about power provision to keep the smart device charged up. Anyone starting out to make their own charger soon learns that it's not so simple. Just connecting the charger output to ground and +5 V pins of the USB socket isn't enough. There are at least half a dozen different charging specifications. When the charger doesn't provide the necessary states on the USB data lines the charging process fails. Help is at hand in the form of a tiny chip with 6 legs.

Building your own 5 V charger for mobile devices is not quite as easy as you may think. Mobile devices are really fussy about how the charger responds to them; they expect more than just regulated power at 5 V. It's a triumph of common sense that today almost all mobile devices can be charged from a standard 5 V USB charger (see the collection in **Figure 1**), but nowadays, in addition to power, the USB data lines must also be correctly terminated by the charger or the mobile device will refuse to charge. As can be seen in **Figure 2** some manufacturers have their own ideas about how the data lines must be terminated which is not especially helpful and just adds to the confusion.

Specification chaos

When designing a charger it is not such a sensible approach to build one that only conforms to the appropriate charging specification for the smart device you happen to be using at the moment. The device can quickly become obsolete; you replace it with a new device and find that the old charger is incompatible. It's much better when you can design an (almost) universal charger, able to provide the correct electrical signatures and charge any mobile device.

There are different ways in which such an intelligent USB charger can be built without using a dedicated microcontroller.

Figure 1. A collection of USB chargers. Clockwise from upper left: Apple 10-W iPad charger, Apple 5-W iPhone charger, Logitech 5-W charger, Noname 5-W charger.

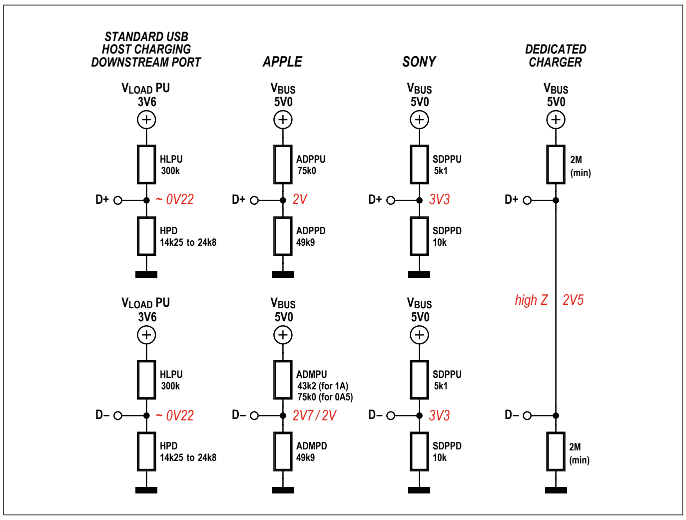


Figure 2. Take your pick: Manufacturers have their own ideas how the USB data lines should be terminated.

Technical Data

- Charge current: 1 A (5 W) and 2 A (10 W)
- Voltage: 5 V
- Specification: DCP Version 1.2
- IC outline: SMD SOT23, 6-pin.

According to the DCP specification — here in the **D**edicated **C**harging **P**ort 1.2 version — it is necessary to terminate the two USB differential data lines in the dedicated charging port according to **Figure 3**. This illustrates the interaction between the mobile device on the left side and the DCP-standard compatible charger on the right.
What's frustrating is not all manufacturers stick to this specification, some choose their own standard: Apple for example use a special voltage divider network (Figure 2). Without the specified voltage levels on the data lines the charging process does not even begin. The official reason for Apple's approach is to protect the end user and their equipment from any damage which may be caused by the use of uncertified chargers.

It only takes 6 pins

Texas Instruments already have an impressively wide range of devices and chips useful for power supply control applications. It shouldn't have been a surprise to find that TI also has a solution to this particular problem: Tiny chips of the TPS251X series. These are specifically designed for USB applications and connect to the two data lines. At power-up the chip uses an auto-detect feature to monitor the USB data line voltage and automatically provides the correct electrical signatures on the data lines to charge compliant mobile devices.
From the technical point of view there really isn't any design work required to incorporate this chip into a USB charger design. If you plan to use it in your own design it's sufficient to know that it supplies signals on the data lines to tell the mobile device whether the charger is capable of supplying 5 watts (standard charge) or 10 watts (power charge) to the device (**Figure 4**). The TPS251xx series of devices only connect to the USB data lines so the data sheet does not go into any detail about charging voltage or current limiting. TI also produces the TPS2561A dual-channel power distribution switch which may also be a device worth considering for any engineer working on a USB charger design. This connects to the USB power pin and switches the charging current. Further details of its application are outside the scope of this article. In our case the general rule is: The charger must be capable of supplying the corresponding power level (as indicated by the TPS251x) to charge the mobile device.

Testing

The chip has just 6 legs and is only available from TI in an SMD SOT23 package. Its tiny size makes it difficult to work with in a prototyping environment. The best solution is to fix the chip to a small break-out board so you can plug it in and out without worrying about re-soldering, solder splashes and short circuits. The author used a standard adapter board from Aries (**Figure 5**).
The next step is to mount the module onto perf board using two 3-way female header strips then fit the USB socket. Note that most USB sockets have mounting tabs requiring larger

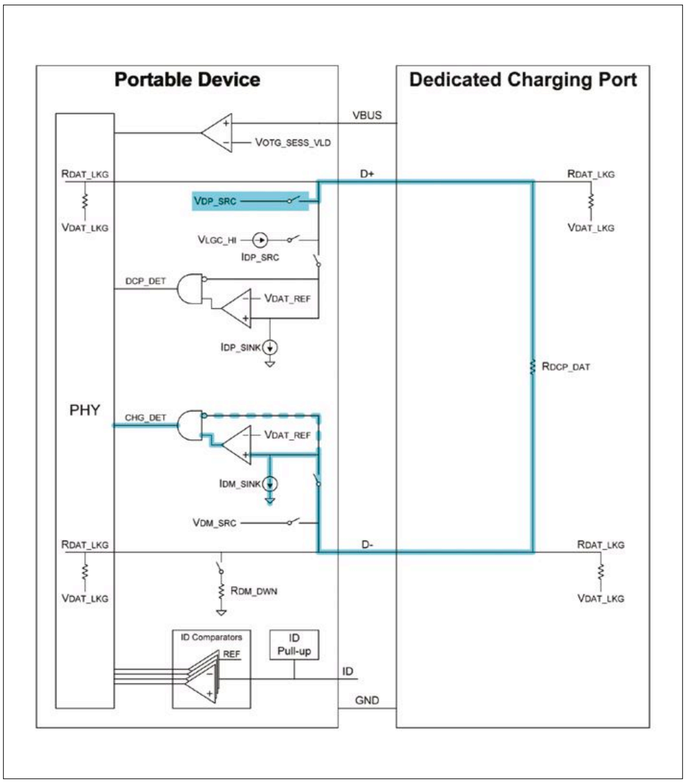


Figure 3. A mobile device (left) hooked up to a DCP compliant charge port (right). image: USB Implementers Forum [1].

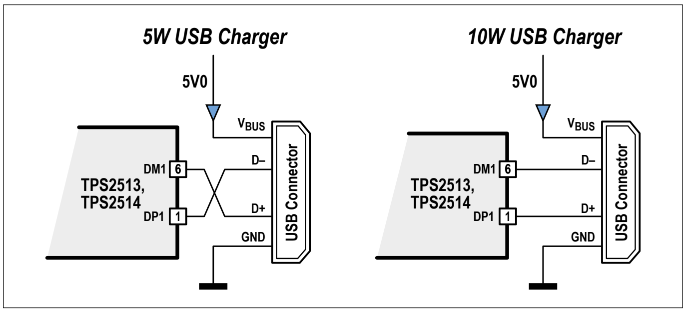


Figure 4. Depending on the connection the IC indicates to the mobile device if the charger can supply 5 or 10 watts.

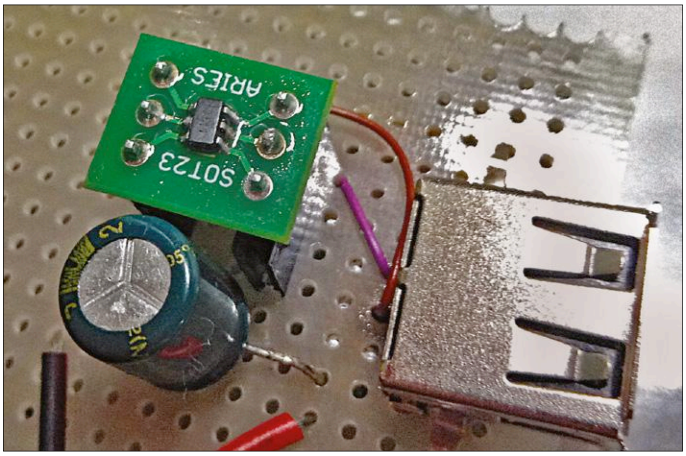


Figure 5. Break-out-board for the SOT23 SMD-package from Aries fitted with the IC from TI.

holes so that they fit through the PCB. The tabs are usually part of the connector shielding and can be soldered to a ground connection. Using these tabs makes the socket mounting more secure and less likely to be pulled out; just for prototyping the author decided not to bother with them.
And so to the electronics: The use of a ceramic capacitor between the ‘D+’ and ‘D–’ pins is recommended in the TI data sheet for this device. This helps reduce any voltage spikes that may occur when the charger is plugged into the device.
Now you need a good quality power supply; the author is using an HP6624A here. The output voltage is set to 5.0 V with a current limit of 3 A. In view of the levels of current the cables will be carrying it is important not to use long lengths of thin cable otherwise the losses generated in the cable will be too big. When necessary you may need to up the voltage slightly to 5.1 V but don’t go any higher! Now it’s time to try it out with some mobile devices.
First off the author plugged in a BlackBerry Q10: this device has a reputation for being fussy about the charger. At plug-in it displayed a message that the charging cable was unsuitable for the BlackBerry Q10 and therefore it could only be slow-charged. This message however disappeared after one or two seconds and the current meter on the power supply registered 1 A. Despite the message, it had definitely switched to fast charge mode.
Next the author tried a Kindle Fire, the supply current reading between 1.6 and 1.8 A indicated that the Kindle was also happy with the power offered by the charger.

Troubleshooting

Circuits that include USB port connections can be a source of trouble. Often the problem can be traced back to the relatively tight specification regarding USB voltage levels. Charging in power mode passes 2 A which can produce unacceptable voltage drops through flaky connectors, thin PCB tracks and skinny cables.

Kick out the terminal blocks!

Use a terminal block in the USB wiring hookup and you will introduce a voltage drop of a few hundred millivolts when the charge current ramps up to fast charge mode. The device on charge will recognize the voltage drop and switch to slow-charge mode or stop charging altogether (see **Figure 6**).

Use good quality cables

One of my niggles is the poor quality of many MicroUSB cables. Long thin cables introduce too much voltage drop which causes the device to switch to slow-charge mode.

Conclusion

When designing a USB charger just for one specific piece of equipment you may not consider it worthwhile to incorporate a dedicated charging port controller such as the TI TPS251xx. As was pointed out in the beginning, the extra work required to incorporate the chip into the design is not too onerous and these chips are priced at around €2, which is not excessive. What you gain is greater flexibility and a charger with more universal application. For the author it proved a worthwhile investment in the system design he is currently developing. The setup uses a process computer to transfer information to a standard tablet via Wi-Fi.

IC options

Texas Instruments offers four different ICs in its TPS251xx family of devices. The TPS2513x version can support two USB ports, while the TPS2514x version handles one port. Versions with an A suffix or no suffix support different voltage divider options; some smartphones and tablets can accept a 12-watt charge which is supported by the A series. A more detailed description of the chip’s properties can be found in the corresponding data sheet [2].

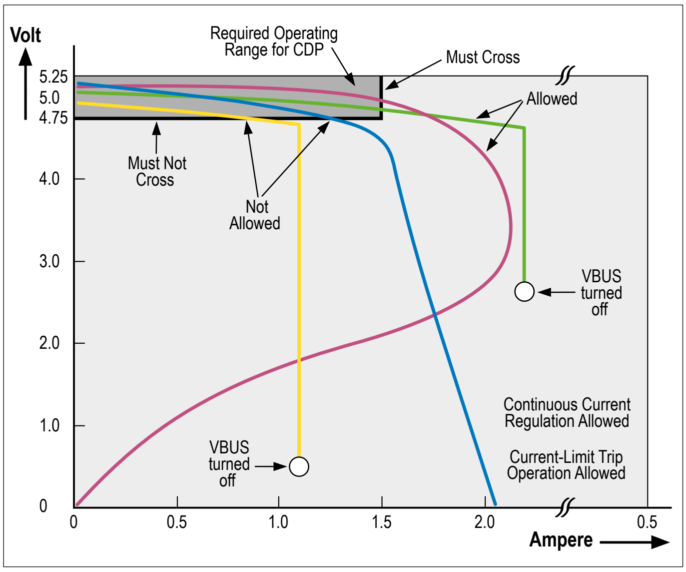


Figure 6. Template showing specified charging parameters for a USB charger.

For his application it was important from the point-of-view of system future-proofing and flexibility that the charger design was compatible with the widest possible range of connected hardware. Incorporating a TI TRS-251xx series of charger port controllers into a USB charger design ensures it will be compatible with all the most important industry standards.

(160360)

Web Links

[1] http://composter.com.ua/documents/BC1.2_FINAL.pdf

[2] http://goo.gl/wxSyJV

reichelt elektronik
Technology connects.

- More than 45 years of experience
- High availability and reliable delivery
- More than 85,000 products

OPTIMAL RESULTS – ALSO IN HARSH CONDITIONS:

NEW DIGITAL MULTIMETERS

PeakTech®

Robust digital multimeter with LED light

Ideal for daily field use in the service, construction and industrial sectors, as well as for the ambitious home user.

- Automatic lighting of the function keys and rotary selection switch
- Strong LED light for illumination of the measuring position
- 3 5/6-digit measurement display (6000 counts)
- LCD display with relative value measurement and 42-segment bar diagram
- Continuity check with buzzer and diode test

IP67

EN 61010-1 CAT IV 600 V

EN 61010-1 CAT III 1000 V

Order no.: PEAKTECH 3443

NEW £60.23

CAN ALSO BE USED IN TOTAL DARKNESS!

Illuminated keypad + integr. work light:

PEAKTECH 3443 and PEAKTECH 3445

PeakTech®

Robust digital multimeter with Bluetooth

Such as PeakTech 3443 with these additional functions:

- True RMS measurement
- Bluetooth interface for connecting to your Android smartphone

Order no.: PEAKTECH 3445

£82.82 NEW

Smartphone not included

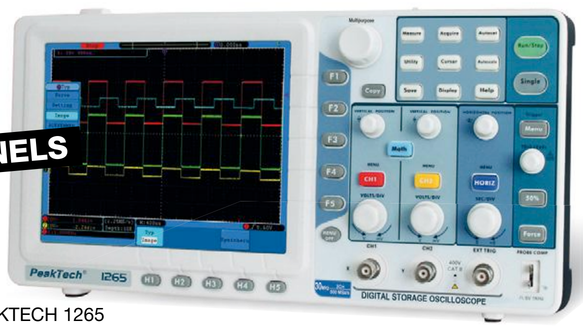
Digital storage oscilloscope

PeakTech®

Digital storage oscilloscope of the latest generation with high-resolution colour display, backlighting, high bandwidth and measuring rate, large internal data storage and USB port.

- VGA output for connecting an external monitor
- LAN-connection for remote query via the network
- 20 cm (8”) TFT colour display

30 MHz 2 CHANNELS



Bestell-Nr.: PEAKTECH 1265

£238.55 BEST SELLER

Daily prices! Price as of: 5. 2. 2018

Onlineshop languages:

reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Germany), Phone: +44 203 808 95 25

www.reichelt.co.uk

ORDER HOTLINE: +44 203 808 95 25

Electromagnetic Interference from LED Lamps

Fourteen examples compared

By **Alfred Rosenkränzer** (Germany)

For more than a hundred years 'artificial light' has almost exclusively meant incandescent lamps. However, the relatively low output of incandescent lamps has been criticized and has ultimately led to their being banned in the EU. Low-energy fluorescent lamps are certainly more efficient, but they do not work well when turned on and off frequently, and they contain not exactly ecologically-friendly quantities of mercury. As LED lamps become cheaper to buy it seems that semiconductor-based lighting is the ideal replacement. So that's all good then. Or is it?

Well, it depends: there have been reports of LED lamps causing electromagnetic interference. Reason enough for Elektor to try to get to the bottom of things.

As suggested above, low-energy fluorescent lamps enjoyed only a few brief moments of popularity. Their toxic chemical content is indeed a problem; but so is the quality of the light produced, and more particularly their behaviour when turned on: it takes some time after power is applied before the bulb reaches full brightness. It is also far from ideal that the service life of the bulbs is considerably reduced if they are turned on and off frequently. Low-energy fluorescents have almost disappeared from the market now, as LED lamps have increased in efficiency and light output and fallen significantly in price. LED lamps are also available in a very wide range of shapes, socket configurations and colour temperatures. They usually reach full brightness immediately they are switched on, they are hardly, if at all, affected by being turned on and off frequently, and (apart from a few dodgy ultra-low-cost examples) their service life is pleasingly long. LED lamps are thus both ecologically and economically sound. So it would seem on the face of it that all our problems are solved and we can turn our attention to other matters.

Table 1. List of lamps tested.

#	Manufacturer	Type	Power	Output (lm)	Efficiency (lm/W)	Weight	Flicker	f1	VHF	SW
0	Osram	Incandescent bulb	60.0 W	710	11.8	23 g	yes			
1	Osram	AB48940	9.5 W	806	84.8	43 g	–			
2	Blaupunkt	ILUA70-16WB B P	16.0 W	1521	95.1	72 g	–			
3	Philips	A60 7W12430	7.0 W	806	115.1	41 g	slight			
4	Müller Licht	400006	5.5 W	470	85.5	34 g	yes			
5	Müller Licht	Retro LED 400223	4.0 W	470	117.5	17 g	–			
6	Osram	LED Star Classic A60	7.0 W	806	115.1	29 g	–			
7	Toom	G60	10.0 W	806	80.6	63 g	–			
8	Opple	LED-EG50-E27	6.5 W	470	72.3	82 g	–			
9	Sygonix	TLK-A60-806PM	9.5 W	806	84.8	94 g	–			
10	IKEA	Ryet 1000	11.5 W	1000	87.0	72 g	–			
11	IKEA	LEDARE 600	8.6 W	600	69.8	78 g	yes			
12	IKEA	Ryet 400	5.0 W	400	80.0	40 g	–			
13	IKEA	LEDARE 1000	13.0 W	1000	76.9	122 g	yes			
14	B1	LED lamp	8.8 W	806	91.6	52 g	–			

Signs of trouble?

At the beginning of September 2017 Elektor received a report from a member of the German amateur radio club DARC that suggested that significantly increased levels of electromagnetic interference affecting broadcast radio, amateur radio and WLAN signals could be traced to LED lamps. We took up the issue in an article online [1]. The German-language version of this article alone attracted over 100 000 clicks.

At the end of 2016 the club published on its website a list of various manufacturers' LED lamps along with spectral plots [2], and the October 2017 issue of the club magazine 'CQ DL' included an article describing the measurements. The German 'Stiftung Warentest' consumer product testing organization also looked at LED lamps, but not with regard to electromagnetic interference. And so it falls to Elektor to look at the question in more detail: in this article we will take a closer look at 14 LED lamps, and give a brief guide to how you can find out for yourself to what extent your own LED lamps are polluting the ether, without the use of large banks of test equipment.

Procedure

First we visited IKEA, our local DIY shed, and the Internet, purchasing a total of 13 different LED lamps with E27 fittings. To them we added a further lamp supplied by an interested colleague. These 14 lamps provide a sample of the market, and include products from well-known companies, 'no name' brands and the DIY shed's own-brand products. The sample is not fully representative but nevertheless provides a good cross-section of what it is possible to buy today. Table 1 gives a list of the lamps we examined, the main characteristics of each, and a qualitative indication of the results of our testing as a colour (green=good, yellow=OK, orange=fair, red=poor).

Of course there is still a lot more to be said. Because anyone interested in electronics will (rightly enough) be keen to know more about the whys and wherefores of our testing, the following pages give more background regarding the measurement techniques we used, including some sample oscilloscope traces.

Interference

First we should look at how various kinds of interference can arise. The power networks in Europe supply approximately sinusoidal AC at a voltage of 230 Veff and a frequency of 50 Hz: in other words, we should expect a low-frequency signal without any significant harmonics to cause interference. In reality, however, there are dimmers, electric motors, switching power supplies in PCs and other devices to contend with, which, in conjunction with the resistive and inductive components of the household wiring, conspire to distort the pure sinewave to a greater or lesser extent. The result is a spectrum containing harmonics extending into the high-frequency RF bands. In turn this affects other devices that depend on interference-free radio signals to transmit and receive information. This includes broadcast radio receivers and amateur radio transceivers, to name just two categories.

In lower-tech days there were fewer sources of interference. The good old-fashioned incandescent bulb is basically just an ohmic piece of resistance wire, passing a sinusoidal current without distorting it. The humble incandescent bulb would also never be party to such dirty tricks as phase shifting or the like at any point in its brief existence.

However, the situation with LED lamps is completely different. It is well known that LEDs should be operated at a constant current because, like all semiconductor diodes, they exhibit a rather low differential resistance in the neighbourhood of the nominal forward operating current, and their forward voltage has a negative temperature coefficient. In the simplest designs both these effects can be smoothed out by the use of a series resistor in conjunction with a higher supply voltage than is strictly necessary. At its nominal operating current the forward voltage of a white high-power LED is around 3.3 V. For example, it is possible to use a series resistor to operate such an LED simply and safely from a 5-V voltage source. However, in this simple configuration around one third of the power consumed by the circuit is dissipated in the series resistor as heat, and so it is only suitable for driving smaller LEDs, for example those used as indicators. Beware, however, that the LED strips sold for DIY applications generally consist of groups of three LEDs plus one protection resistor in series, the groups being wired in parallel across the 12-V supply. These strips dissipate about 20 % of the power consumed as heat in the resistors.

On the other hand it should be said that the direct operation on LEDs at constant voltage in this way, although suboptimally efficient, has impeccable credentials from the viewpoint of electromagnetic interference. The arrangement causes no distortions to current waveforms and therefore generates no RF. However, LED lamp manufacturers are keen to make the most efficient devices possible, and to do this they have to eschew the simple series resistor in favor of switching power supply techniques. And that is where the problems start.

AC line operation

First we should mention an exception to the discussion below. In the article 'Let There Be LED' in the January & February 2016 issue of Elektor [3] Thomas Scherer looked at the LED shown in Figure 1, which was advertised as a 10-watt bulb. Had it not fallen to pieces inside its box during shipping, he would not have noticed what a strange product he had acquired: not only had the switching power supply been dispensed with and replaced with a series capacitor, but it also became apparent that the lamp was in fact only a 7 W model. The advantage of these economies was that the lamp produced hardly any electromagnetic interference. The same approach is used in modern imitation 'filament' lamps, which are in fact made from long rows of LEDs arranged to look like retro-style incandescent bulbs.

Some 90 % of all LED lamps, however, contain a full switching power supply that delivers at its output a constant current suitable for the series string of LEDs, typically either a small number of power LEDs or a large number of smaller LEDs. The circuit works as follows. As in any ordinary switching supply, the AC line input is first rectified and then smoothed using a high-voltage electrolytic capacitor. The resulting DC voltage is then electronically 'chopped up' and fed to a small transformer. The current flowing from the output of the transformer is measured and used as feedback to the electronics that controls the switching transistor. Typical switching frequencies are in the range from 30 kHz to 100 kHz: the fundamental and second harmonic thus have fairly low frequency in RF terms. The signal driving the transformer, however, is far from a sinewave: it is more like a squarewave with very fast edges. The fast switching edges mean that the RF spectrum includes many higher harmonics of the switching frequency at considerable power. Fast switching is good for the efficiency of the circuit, and a higher switching frequency is good for the mechanical design, as it allows a smaller transformer to be used. Unfortunately both these pressures act to increase the level of RF interference. Filtering to try to attenuate the interference adds to circuit complexity and hence cost, and negates the advantage of a physically smaller circuit design; for this reason, many designs only feature limited filtering.

Figure 1. Innards of a low-cost LED lamp: dropper capacitor, diode, filter capacitor and discharge resistor. The silk screen legend gives it away: 7 W, not the claimed 10 W!

Test and measurement

To make sure that we exercised the various lamps under identical conditions we set up a test environment. The first stage, as is apparent from Figure 2, is to isolate the set-up from the mains supply using a transformer: this helps protect both the test equipment and its operator. The four taps on the secondary side of the transformer, at 15 V intervals, let us test whether the lamp brightness varies significantly with changes in the supply voltage.

The ground of the circuit is connected to mains earth so that it can in turn be connected to the ground of an oscilloscope. The current through the LED lamp is mea-

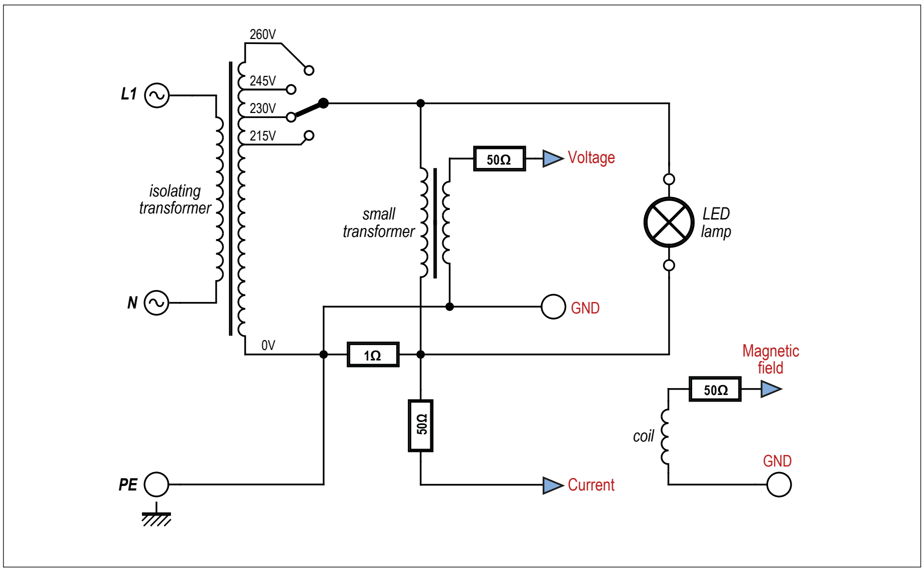


Figure 2. Test set-up for measuring LED lamps.

sured using a 1 Ω shunt resistor, with a further 50 Ω resistor providing impedance matching to the connected BNC cable. A small mains transformer in parallel with the LED lamp provides a stepped-down voltage on its secondary side proportional to the supply voltage. This allows us to measure interference on the voltage signal, and in particular provides a convenient way to trigger the oscilloscope.

Our arrangement allows us to measure current and voltage waveforms safely and accurately. But we want to go further than that: we add a suitable receiver coil to measure the stray magnetic field and an antenna for the electromagnetic field. We will look at these latter two measurements in more detail below.
We are not just interested in the time-domain behaviour of the current draw of the lamp: we also want to examine its spectrum.
In order to clearly identify our 14 LED lamps (plus the incandescent bulb we are using as a reference) we have numbered them as listed in Table 1 and these numbers are used to label our figures and screenshots. The photos in the page borders show the sample lamps we tested.

Current waveforms

As an experimental control we measured the waveform of the current drawn by a 60 W incandescent bulb. We do not have a photo of the lamp itself, but the plots show us what to expect from an interference-free lamp. Even when the mains voltage waveform is slightly distorted, we can see from the voltage (purple) and current (yellow) traces in Figure 3 that there is no interference. The trace colour scheme is the same in the plots for the LED lamps in Figure 4 to Figure 17. The current waveforms immediately provide us with our first surprise. The first lamp we tested (see Figure 4) exhibits sharp current spikes in its current waveform near the extremes of the voltage waveform, rather than a smooth sinusoidal or trapezoidal shape. The needle-like spikes correspond to the points in time

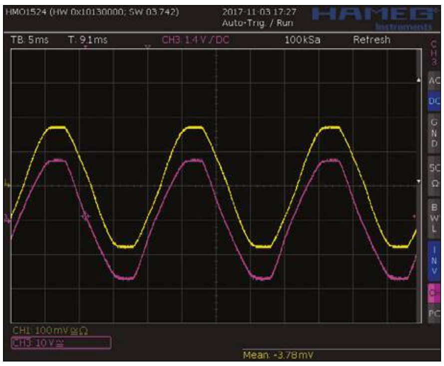


Figure 3. Current and voltage traces from the incandescent bulb.

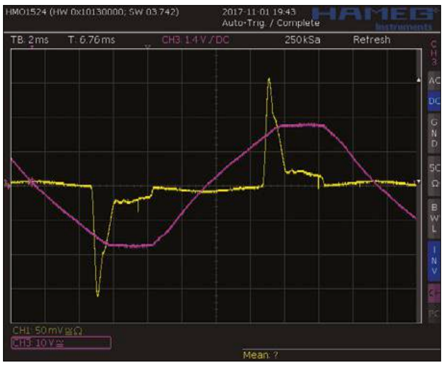


Figure 4. Current and voltage traces from lamp 1.

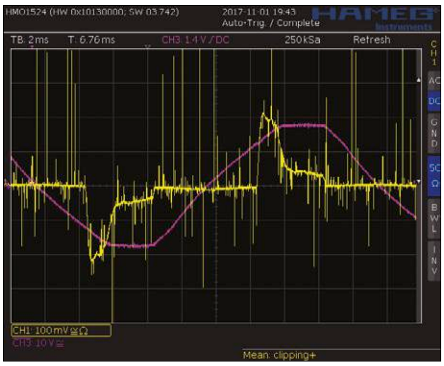


Figure 5. Current and voltage traces from lamp 2. Oh dear!

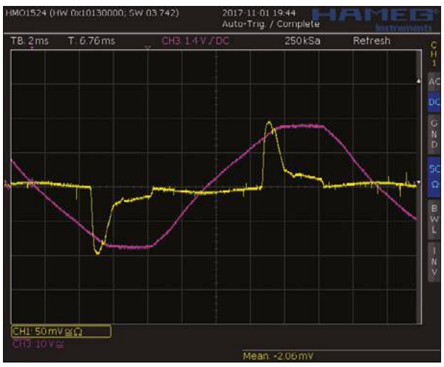


Figure 6. Current and voltage traces from lamp 3.

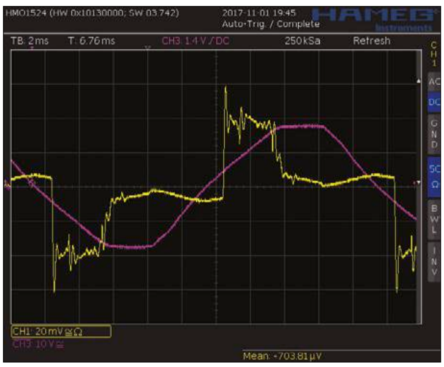


Figure 7. Current and voltage traces from lamp 4.

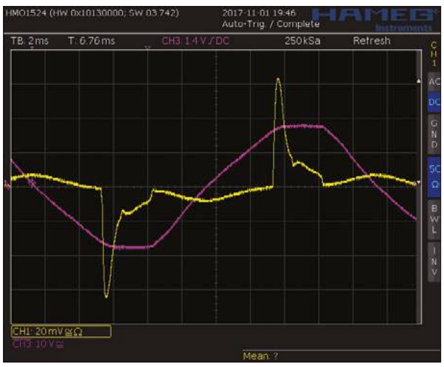


Figure 8. Current and voltage traces from lamp 5.

where the smoothing capacitor is charged back to its peak value. The following relatively flat region of current consumption, which looks like the seat of a chair, corresponds to the current drawn by the LEDs while the smoothed voltage is relatively constant. Once the supply voltage falls sufficiently the circuit is powered only from the smoothing capacitor, and no current flows from the mains until the mains voltage again exceeds the capacitor voltage, during the next half-cycle. A glance at Table 1 shows that this lamp does generate some interference in the VHF band, even though the current waveform looks relatively smooth.

However, if you think the current waveform for lamp 1 looks bad, you will find Figure 5 a sight for sore eyes. In addition to the charging curve that we saw for lamp 1, there is also a large number of very sharp spikes with an amplitude easily double that of the charging spikes. It doesn't look great. And indeed, as Table 1 reveals, this is among the worst of the lamps we tested.

The general shape of the current waveform for lamp 1 is also seen in the traces for lamps 3, 4, 5, 6, 8, 9, 10, 12 and 14. However, take a look at Figure 14 and Figure 16 (lamps 11 and 13 respectively). The current waveform has no charging peak: it is just a relatively smooth, roughly trapezoidal curve with a slight phase shift. The electronics in these devices must be charging the smoothing capacitor at a reasonably constant (and therefore low) current, beginning when the supply voltage reaches just a few tens of volts and lasting for a correspondingly long fraction of the half-cycle. The period where no current flows is very brief. In RF terms the performance of lamp 13 is excellent: practically no interference. Lamp 11 is nearly as good, and IKEA could claim two top spots if the nasty little spikes visible in Figure 14 were eliminated. It would seem that IKEA is purchases devices from more than one supplier.

Figure 10 paints a more ambiguous pic-

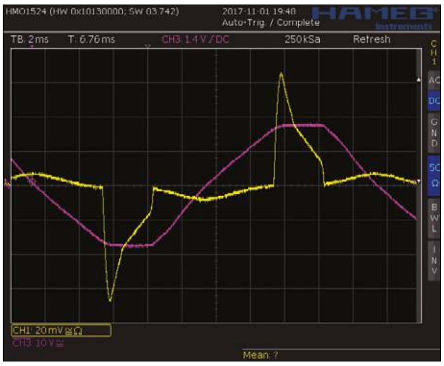


Figure 9. Current and voltage traces from lamp 6.

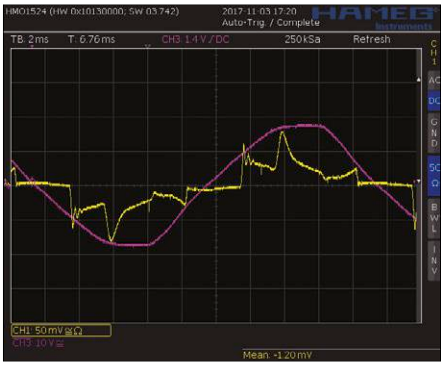


Figure 10. Current and voltage traces from lamp 7.

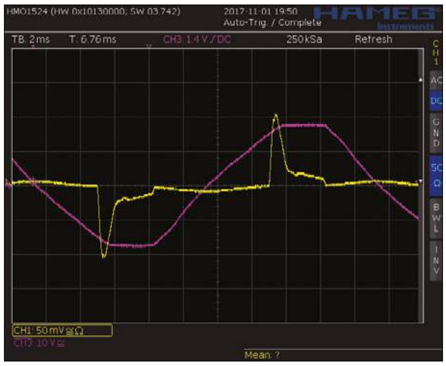


Figure 11. Current and voltage traces from lamp 8.

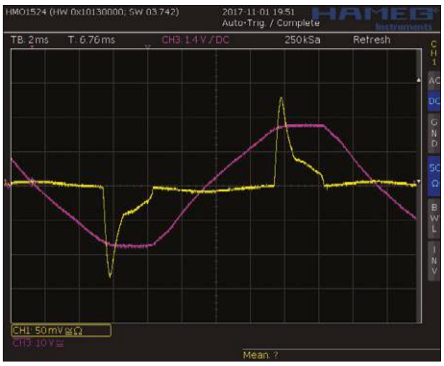


Figure 12. Current and voltage traces from lamp 9.

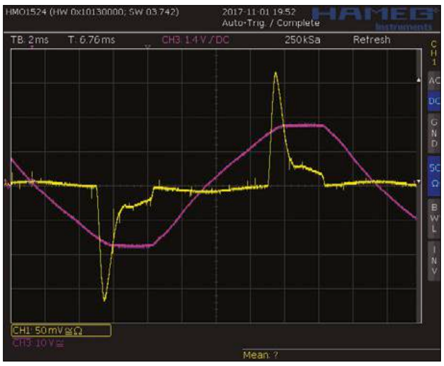


Figure 13. Current and voltage traces from lamp 10.

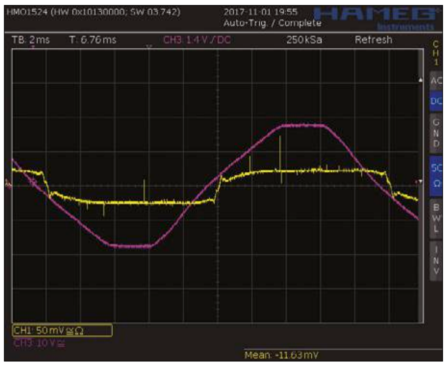


Figure 14. Current and voltage traces from lamp 11.

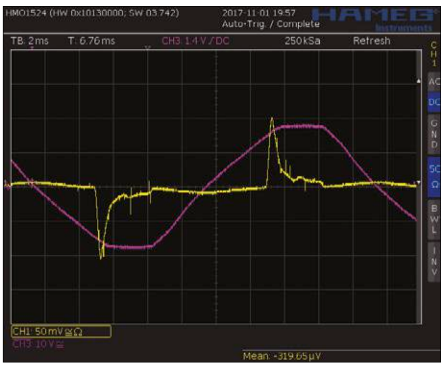


Figure 15. Current and voltage traces from lamp 12.

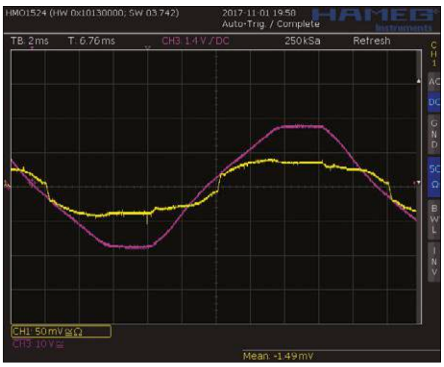


Figure 16. Current and voltage traces from lamp 13.

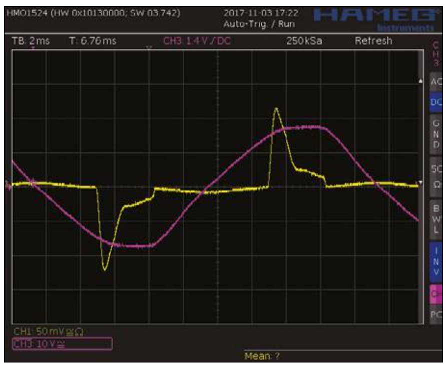


Figure 17. Current and voltage traces from lamp 14.

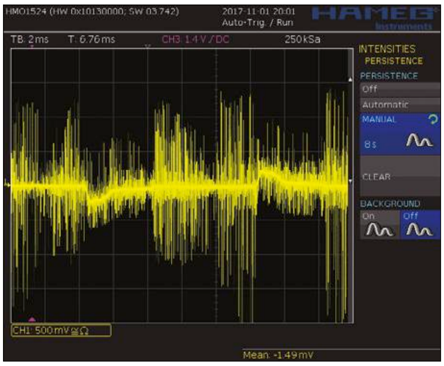


Figure 18. Current trace with persistence enabled: lamp 11.

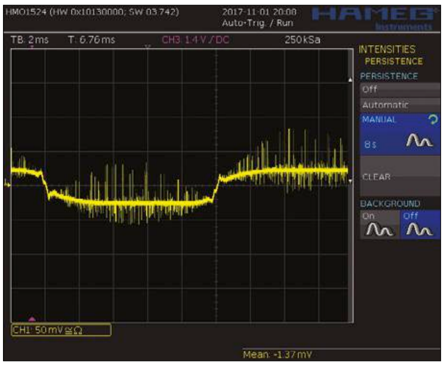


Figure 19. Current trace with persistence enabled: lamp 2.

ture. It appears that the lamp charges its smoothing capacitor when the voltage reaches about half its peak value, both starting and stopping suddenly; in between there is a further charging spike as we saw in many of the other

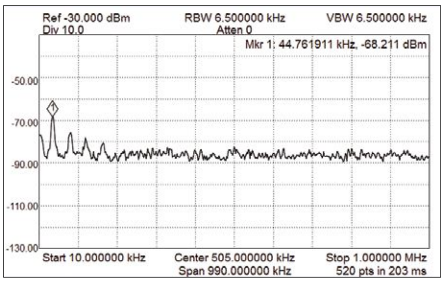


Figure 20. Current spectrum for lamp 1.

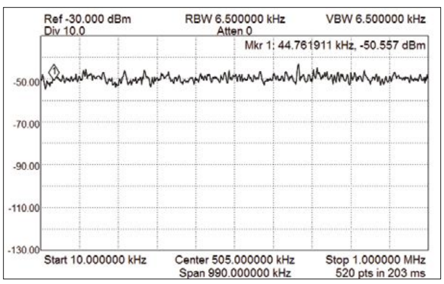


Figure 21. Current spectrum for lamp 2. No surprises there!

lamps. Perhaps a good idea with a poor implementation? The sharp edges to the waveforms suggest that we should expect some RF interference, and indeed this lamp from IKEA receives an orange warning in Table 1, the second worst interferer in our tests.

Current with 8-s persistence

Because the spikes are understandably responsible for the majority of the interference, it is helpful to look at the current waveform with persistence enabled on the oscilloscope. This makes the narrow spikes much bolder and so easier to see. The comparatively innocent-looking spikes in Figure 14 are thus transformed into the forest shown in Figure 18. It is now clearer why lamp 11 generates so much interference.

Just as dramatic an effect is seen with lamp 2. Figure 19 shows an impressively uniform series of spikes at all points along the current waveform.

Current spectrum

Our next step was to measure the spectrum of the current draw using a Signal Hound type USB-SA44B analyzer. A 220-nF capacitor was used to block the DC component of the signal, and in combination with the 50 Ω termination impedance forms a high-pass filter with a cutoff frequency of around 15 kHz. This is a convenient way to achieve considerable attenuation of the 50 Hz and 100 Hz components, as well as their first few harmonics. The addition of a Mini Circuits FTB-1-6 balun, with an operating frequency range of 10 kHz to 125 MHz, galvanically isolates the instrument from the test rig.

When making measurements with the analyzer it is always a good idea to drive it from a laptop running off its battery. The power supply in a desktop PC that we tried using created a large amount of interference, both radiated and conducted, that upset the measurements. We examined the spectrum from 10 kHz to 1 MHz: at higher frequencies we did not observe any significant power.

In the spectral plots the green curve shows for reference the background noise measured by the test rig with the LED not supplied with power. The spectral plot for lamp 1, shown in Figure 20, is interesting: there is a clear peak at 44.76 kHz (the lamp's internal switching frequency) and its harmonics. Above the fourth harmonic the background noise level starts to rise, but it remains below –80 dBm. Spectral plots obtained from other lamps are similar in appearance and do not provide much in the way of further insight, although lamp 12 distinguishes itself by having a noise floor some 8 dBm higher than its companions.

You may be wondering: what about the spectrum from lamp 2? Figure 21 reveals the awful truth: interference at powers of up to –43 dBm. Even the internal switching frequency is drowned out in the sea of broadband RF noise. It's hardly surprising that a lamp like this makes radio reception nearby practically impossible.

Low-frequency magnetic fields

Figure 22 shows the test set-up to measure magnetic fields at frequencies from 10 kHz to 1 MHz. The magnetic field sensor is an off-the-shelf 68-µH ferrite-core coil mounted 5 cm from the lamp under test. As always lamp 1 behaves in an exemplary fashion, and Figure 23 shows the clean magnetic spectrum measured from lamp 10. The fundamental of the switching frequency is hardly visible, and at higher frequencies the level of interference is pleasingly low. As before, the green curve shows the measured background signal with the lamp switched off as a reference.

Turning to the bad boys: the results from lamp 12 shown in Figure 24 reveal an array of harmonics across the entire measured bandwidth. And, as might be expected, lamp 2 (results in Figure 25) steals the show from a magnetic point of view.

Incidentally, lamps 5 and 6 show almost no stray magnetic field, which is probably because, as mentioned earlier, they do not contain switching power supplies.

High-frequency magnetic fields

Our investigations into magnetic fields did not stop at 1 MHz: we also examined the spectrum from 1 MHz all the way out to 110 MHz. For this experiment we replaced the ferrite-core coil with a more suitable air-core coil of five turns on a 3-cm diameter. With the lamp turned off we observed a spike at 24 MHz: bear that in mind when looking at the spectral plots.

The results from lamp 10 (Figure 26) and lamp 13 (Figure 27) are particularly noteworthy. From Figure 26 we see that lamp 10 produces broadband magnetic interference but at a low enough level not to disturb other equipment. More worrying is lamp 13, which shows a considerable level of interference between 14 MHz and 17 MHz. And, as ever, lamp 2 is pretty bad: even the background spike at 24 MHz does not make it through the noise it produces, as shown in Figure 28.

Electromagnetic interference

It is also important to find out what we can pick up in the way of electromagnetic interference using an antenna. For these experiments we used a 50 cm whip antenna mounted 50 cm from the lamp under test as a sensor, and we extended the frequency range we examined to 220 MHz. Although the experiments were carried out in a basement room that provided good electromagnetic attenuation, there were still background signals at several frequencies, as shown by the green reference line in our spectral plots. To get useful measurements for the better specimens among our test set we had to enable the preamplifier on the spectrum analyzer. We then saw some interference at 10 MHz from the laptop's mains power supply; unfortunately by this point in the series of tests the machine's battery was running low. Two of the lamps produced striking results. Figure 29 shows the result from lamp 11: a slightly increased level of interference above 100 MHz. Again, lamp 2 (Figure 30) puts in the poorest performance. It is practically impossible to receive broadcast stations on a radio placed near to this lamp.

Miscellanea

When changing lamps in the test rig it became apparent that the lamps differed considerably from one another in weight. Since the fitting and envelope presumably weigh much the same in each case, and the electronics and LED (apart from any heatsink) are fairly light, it must be the case that the difference is down to the transformers, coils and chokes used in the device, which in turn correlates with the effort put into the design. We have listed the weight of each lamp in Table 1: observe that lamp 5 is actually lighter than the incandescent equivalent. We used a photodiode to detect whether the rectified mains frequency (100 Hz) or the switching frequency was present in the light output of each device: the results are listed in the column headed 'Flicker' in Table 1. Even the incandescent bulb exhibited a little flickering at 100 Hz. The light output and power consumption data in Table 1 are as given by the manufacturers, and from these we have calculated the efficiency in lm/W. The best performers here are lamp 3 and the two retro-style lamps, numbers 5 and 6. In no case was the LED's current waveform sinusoidal. In order to verify that the power figures correspond to reality, we tested lamp 1 with a power meter

Figure 22. A small coil mounted 5 cm from the lamp is used as a magnetic field sensor.

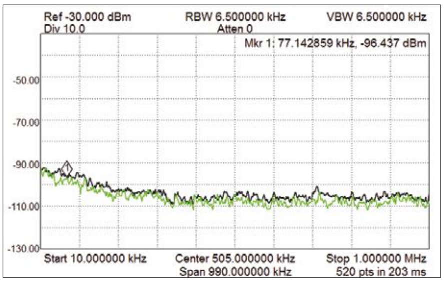


Figure 23. Low-frequency magnetic spectrum from lamp 10: looks good.

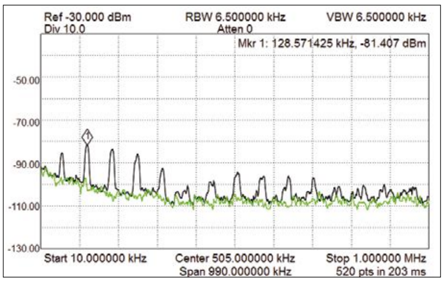


Figure 24. Low-frequency magnetic spectrum from lamp 12: a series of harmonics.

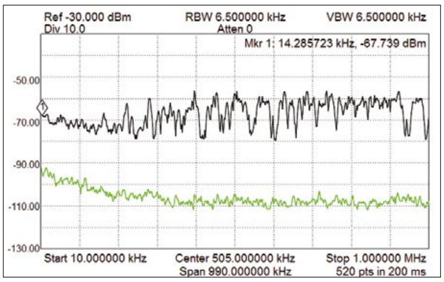


Figure 25. Low-frequency magnetic spectrum from lamp 2: not exactly great.

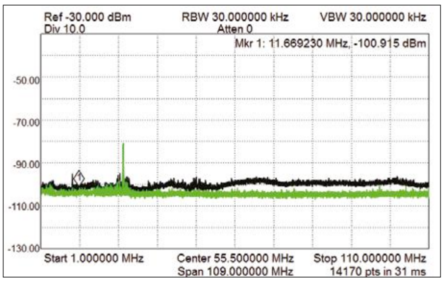


Figure 26. High-frequency magnetic spectrum from lamp 10: some noise at higher frequencies.

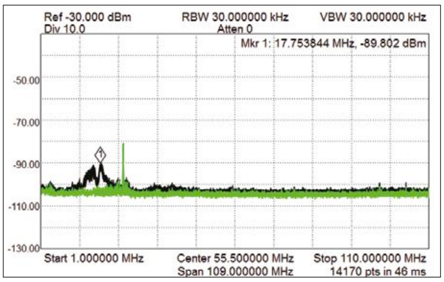


Figure 27. High-frequency magnetic spectrum from lamp 13: peaks at 14 MHz and 17 MHz.

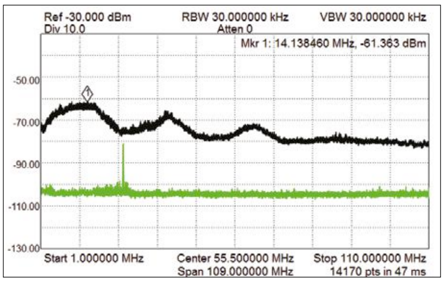


Figure 28. High-frequency magnetic spectrum from lamp 2: it doesn't look good.

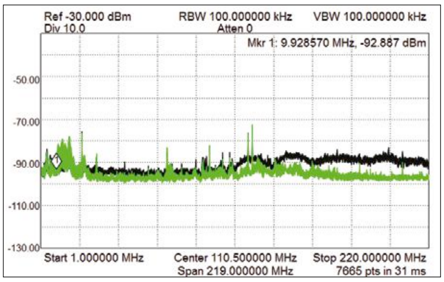


Figure 29. Antenna signal near to lamp 11.

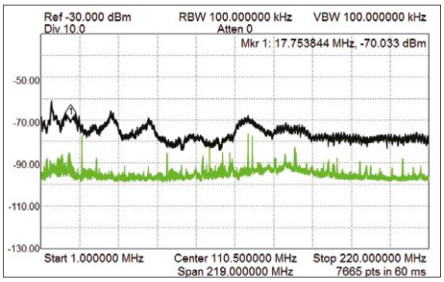


Figure 30. Antenna signal near to lamp 2: an excellent generator of interference.

EMI testing using a radio

In a room with modest VHF reception (the author's basement) we placed a radio about 20 cm from a LED lamp. The antenna was adjusted until the transmitter could be heard without interference, and then the LED was turned on (see image). A short video of this procedure can be seen on the Elektor web pages accompanying this article [4].

As expected, lamp 2 caused a great deal of interference. Lamp 10 caused a small amount of interference and lamps 1, 7 and 11 very little. All the other devices generated no interference at all. In order to hear the effect, it is necessary to listen close to the loudspeaker.

This simple test is surprisingly sensitive, and it is easy to explain to the man in the street how to carry it out and so determine whether an LED lamp is causing radio interference. If a portable radio is not available, the FM radio present in many smartphones will do instead. An ordinary table lamp or similar can be used in place of the mounted socket shown.

Figure 31. EMI testing using an FM radio.

However, FM reception is not the end of the story: with a shortwave receiver we were able to repeat the test at 8 MHz. Here lamp 2 was the only offender; the other lamps were fine. So, although this test does detect the interference from lamp 2, the simple FM radio test is in fact more sensitive.

Modern times demand a modern approach, and so for our third experiment the author's car radio, a DAB+ receiver, was pressed into service. The result of this test was that even lamp 2 did not disturb reception, which illustrates how robust the transmissions are. In FM mode, however, the car radio was not able to pick up a signal in the presence of lamp 2. The other lamps were not tested with this radio.

(a Voltcraft 4500 PRO DE). We measured the power at 10 W rather than the specified 9.5 W, which is acceptably close. Direct calculations from the current and voltage waveforms that we obtained, which should certainly be accurate enough, also indicate that there are no great deviations from the manufacturers' claimed figures.

Conclusion

Happily almost all the LED lamps we purchased for this test exhibit either no interference or so little interference that it can hardly be detected.

Lamp number 2, however, is an excellent generator of interference and should never have graced the supermarket shelf. We cannot be sure, of course, whether this is faulty design or whether our device is an outlier or defective. Things could be worse, however: see the 'Electronic transformers' text box.

If you suspect that a LED lamp is causing radio interference, you can follow the procedure outlined in the 'EMI testing using a radio' text box. If you discover a rotten apple, please write to Elektor at editor@elektor.com with subject line 'LED lamp EMI'. If it is a device that has yet to be tested then we will pass the details on to the author for further investigation, and, if applicable, the results will appear in an update to this article covering a wider range of manufacturers. ⏮

(160610)

Web Links

[1] www.elektormagazine.com/news/led-rumpus

[2] www.darc.de/der-club/referate/emv/konformitaetsmessungen/#c201092 *(in German)*

[3] www.elektormagazine.com/150577

[4] www.elektormagazine.com/160610

[5] www.elektormagazine.com/080691

Electronic transformers

Every DIY store should really put up a sign saying 'Beware of electronic transformers'. And no, we are not talking about 'robots in disguise'! No, we mean the devices that are used to power 12-V lighting systems, using halogen spotlights and more recently spotlights and bulbs employing LEDs. Considering their power output they are small, lightweight and cheap. They contain a small switching power supply, but of a special design as you will see if you take a second look at the photo, which comes from an article by Thomas Scherer in the December 2008 issue of Elektor [5].

The circuit lacks a smoothing capacitor on its input and a filter capacitor on its output. The latter omission explains why the output is a rather brutal squarewave at 12 Veff (albeit at a frequency of 30 kHz). Connect this output to some spotlights via a few meters of unshielded cabling behind a suspended ceiling and you have optimal conditions for radiating interference.

Figure 32. The spartan interior of a 12-V electronic transformer.

The lack of input smoothing also results in an enormous 100 Hz component in the output signal: and so we have not just interference, but an RF carrier modulated six ways to Sunday by a 100-Hz tone. The result is a not-so-soothing hum at the output of any nearby radio receiver. Astonishingly there seems to be nothing illegal about this set-up, or at least nothing illegal about selling the components: electricians will continue to blithely install such systems while radio amateurs nearby pull their remaining hair out. There is certainly scope for government intervention to prevent these systems becoming more widespread.

Tektronix Readout System

Peculiar Parts, the series

By **Neil Gruending** (Canada)

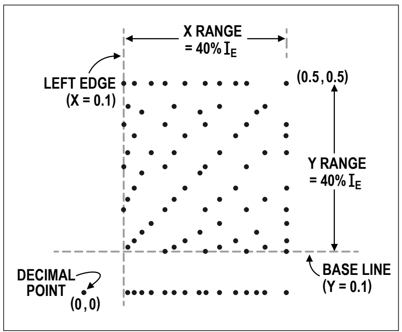


Figure 1. The readout is “drawn” using CRT coordinates.
(Redrawn from Tekwiki [1])

Tektronix needed a way to display data on the CRT for their 7000 series analogue oscilloscopes but it was a challenge because the initial oscilloscopes were analogue-only designs. Originally they had planned to use fibre optics to create a small display area next to the CRT but then a Tektronix engineer named Barrie Gilbert had a better idea: how about drawing the characters directly on the CRT? His prototype proved the idea so Barry went on to design the readout system using several custom designed integrated circuits.

Barrie’s solution allowed the oscilloscope plugins to use 2 resistors to select the row and column of the characters to be displayed from an 11 × 10 matrix of available characters. Up to 10 characters could be arranged into a word which were then displayed in the appropriate CRT area. Timing circuitry assigns a timeslot to each character which is then decoded to select the appropriate character generator IC to draw the character onto the CRT. The timing circuitry could also do things like pad words with zeros or skip words as well.

The character generator chips use a stroke font of sorts with 7 strokes per character. A triangle-wave scan signal was used to activate the CRT XY coordinates (**Figure 1**) [1] of each stroke which activates one of 8 groups of 3 transistors like in **Figure 2**, each biased with a constant current I_e. Two of the transistors provided the CRT’s X and Y current signals with the final transistor directing any unused current to the chip substrate so that

$$I_e - I_x - I_y - I_z = 0$$

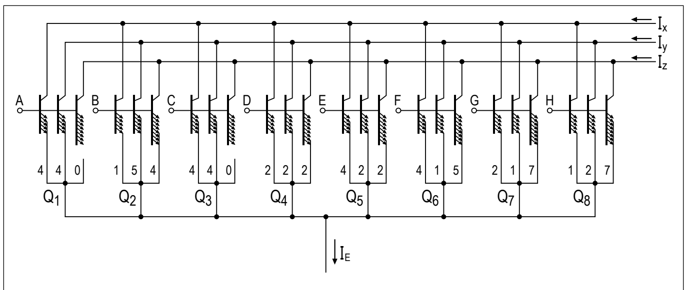
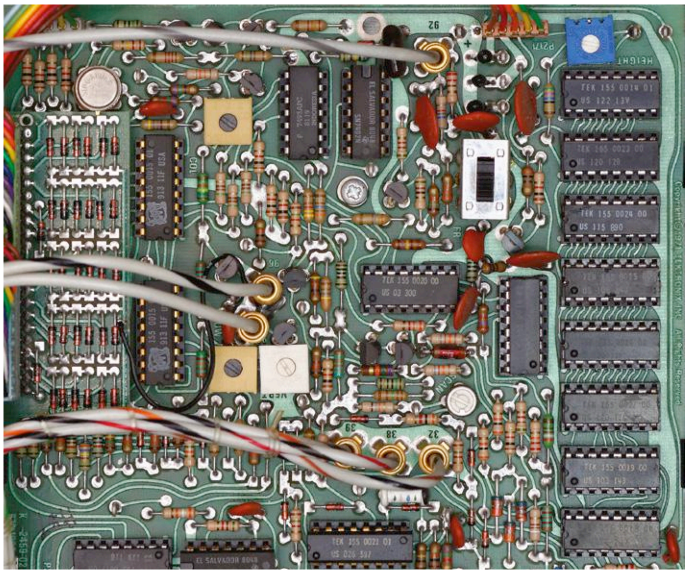


Figure 2. Example of a readout cell. (Redrawn from Tekwiki [1])



Tektronix 7904A readout board. The row and column decoders are on the upper left, the timing ICs in the middle and the character generators are the right side column. (Image courtesy TekWiki)

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

The triangle signal is important because it causes the CRT to draw lines between the points instead of them just appearing as dots.

The design of the character generator is also interesting because the emitter areas in the mask are the same for different variations of the chip. Each area contains a number of emitters of equal size that can be connected as needed by only changing the metallization layers. That way the same silicon layout could be used for all of the variations.

I’m impressed with what Barrie was able to implement with his design and fortunately there are online copies of Tektronix’ excellent service manuals available for their 7000 series oscilloscopes with super detail about how the circuitry works for your enjoyment. There’s also other excellent sources for information like the TekWiki [2] or the TekScopes mailing list [3]. Just remember that not all Tektronix 7000 series oscilloscopes use this design because Tektronix eventually transitioned to a digital readout circuit that was used on later models. ◄

(160528)

Web Links

[1] w140.com/tekwiki/wiki/7000_series_readout_system

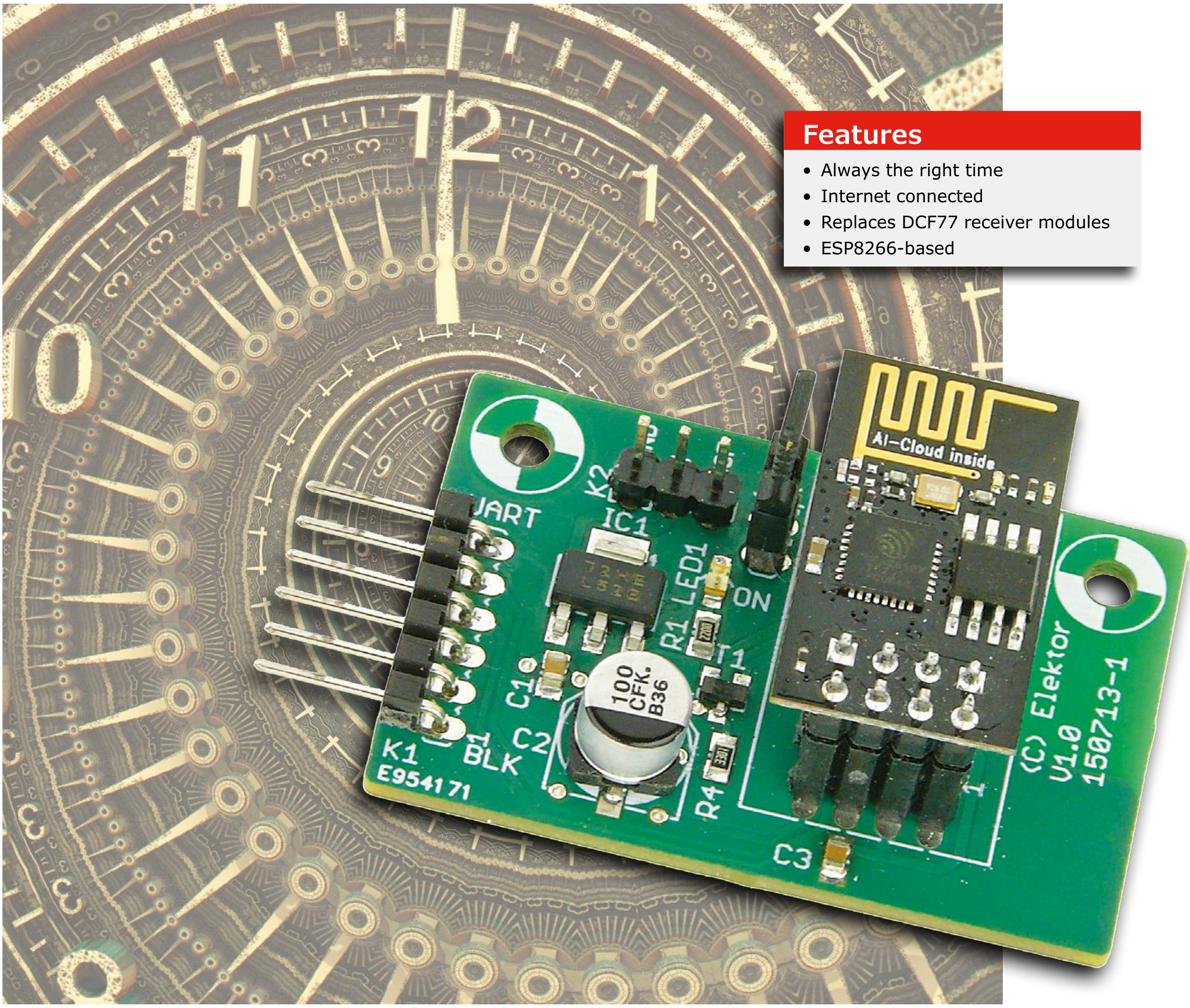
[2] w140.com/tekwiki/wiki/Main_Page

[3] https://groups.yahoo.com/neo/groups/TekScopes/info

LABS PROJECT

DCF77 Emulator with ESP8266

Replace over-air time by Internet time



Features

- Always the right time
- Internet connected
- Replaces DCF77 receiver modules
- ESP8266-based

By **Massimo Fusari** (Italy) & **Luc Lemmens** (Elektor Labs)

About twenty years ago I recycled and modernised a vintage clock with Nixie tubes built by my father in the nineteen seventies. I replaced the digital logic by a microcontroller and used a DCF77 receiver module instead of the original 50-Hz derived timebase.

Over many years the clock has worked fine, but recently DCF77 reception in my house has worsened due to electromagnetic interference created by modern switching power supplies, I suppose. So I decided to replace the DCF77 receiver by some form of Network Time Protocol (NTP) client.
An ESP-01 ESP8266-based module would be perfect for the job. It is cheap, powerful, and can be easily programmed in Arduino style with all the benefits of open source libraries. The result is a DCF77 receiver emulated by an ESP-01 module connected to my home Wi-Fi network. Only one output pin is required to drive the old Nixie clock.
Although the idea behind this project is quite straightforward, the implementation may be a bit more elaborate. The software may need a few modifications to make it compatible with your DCF77 clock, so Luc at Elektor Labs designed a small printed circuit board (PCB) that makes it very easy to (re)program the ESP-01 module if and when needed. The circuit presented here therefore is a DCF77 emulator and ESP-01 programmer rolled into one circuit.

The hardware is simple

The schematic of the emulator (**Figure 1)** isn't particularly complex. The ESP-01 module is represented here by MOD1, powered by a 3.3-V low-dropout (LDO) voltage regulator (IC1). Jumper JP1 is available for putting the ESP-01 into programming mode.
LED1 indicates that the supply voltage is present, however since the ESP-01 has its own power LED, LED1 (together with R1) may be omitted.
K1, which is pin-compatible with an FTDI cable, permits connecting a USB-to-serial converter **of the 3.3 V type**.
Transistor T1 will make interfacing the emulator to your clock slightly easier as it translates the 3.3-V level of the ESP-01's output to the logic voltage level of your DCF77 clock. Collector resistor R4 may be omitted if the clock at hand already has a pull-up resistor at the input of the DCF77 decoder.
The DCF77 output signal gets inverted by T1, but the software fixes this. You may need to check the documentation of the DCF77 receiver module in your clock to see if the output signal should be inverted or not. Of course, it is always possible to use the good-old 'trial and error' method to find out which polarity works for you, or use an oscilloscope to observe the output signal polarity from your original DCF77 receiver.

All the hard work is done by ReadAndDecodeTime

The program is in the shape of an Arduino sketch. As is customary for these kinds of programs, they first execute a function called `setup` and then continuously repeat a function called `loop`.
The function `setup` first sets up the ESP-01's inputs and outputs and it initializes some global variables. It also starts a timer that fires every 100 milliseconds. This timer is used to produce the DCF77 encoded bit stream (refer to the Internet for the details about the DCF77 protocol). Finally, a connection to the Wi-Fi network is established.
The function `loop` is very simple and runs just once every minute. When it does, it either connects to the time server to obtain the current time or it tries to reconnect to the network because the connection was lost for some reason.
The reason for the function `loop` to idle is because it delegates all the hard work to the function `ReadAndDecodeTime`. This is the workhorse that really connects to the NTP server to request the time and then converts it into something that can be easily encoded in the DCF77 format.
The function `CalculateArray` is assigned with this task. It converts the different values that make up a valid frame into zeroes and ones at the right position in the frame.
The 100-ms timer mentioned above calls the function `DcfOut`. This function reads the bit array filled by `CalculateArray` and delivers the values in the shape of 100-ms (zeroes) and 200-ms (ones) pulses on pin GPIO2.
The conversion of NTP time to DCF77

PROJECT INFORMATION

DCF77 | ESP-01 | ESP8266 | emulator | NTP | Wi-Fi

entry level
→ intermediate level
expert level

2 hours approx.

SMD soldering iron, computer, Arduino IDE

£14 / €15 / $16 approx.

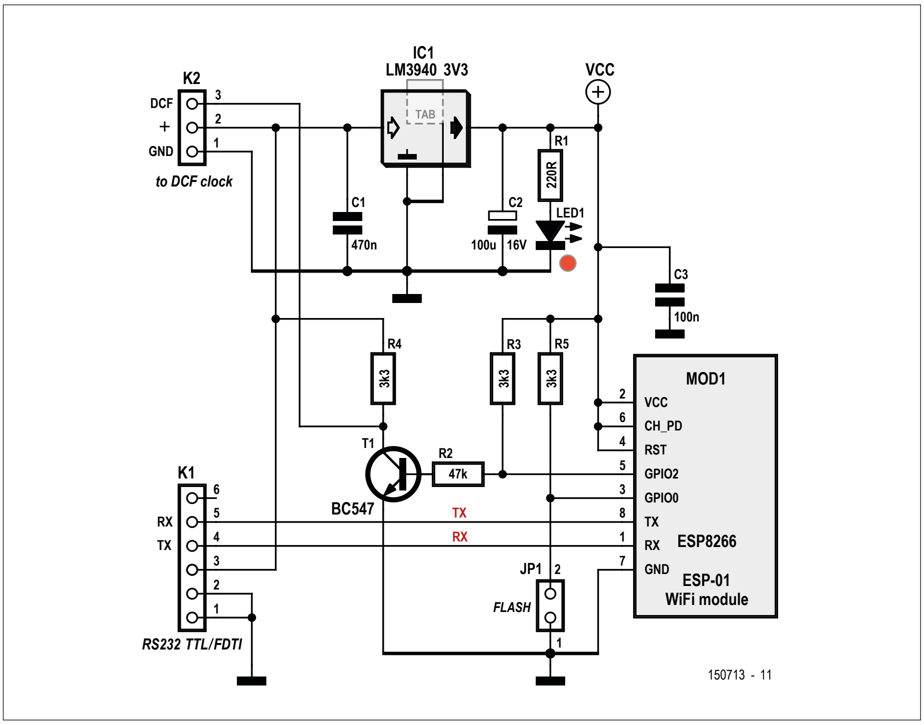


Figure 1. The intelligence of the DCF77 emulator resides in the ESP-01 Wi-Fi module.

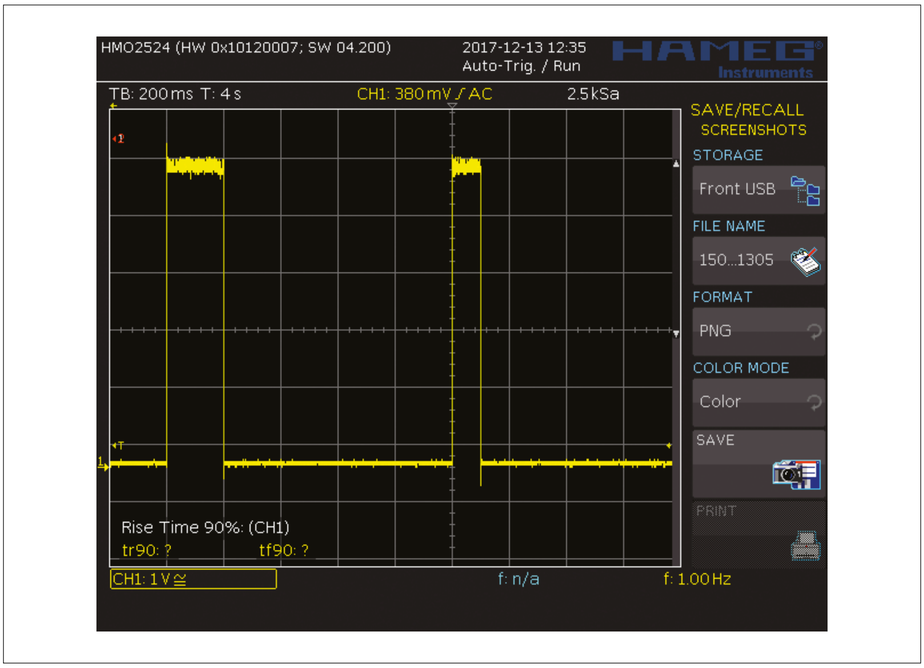


Figure 2. The pulses produced by the emulator are timed perfectly. A logic one (left pulse) is 200 ms, a zero is 100 ms and the time between each pulse is one second.

time might seem a trivial task but it's likely more complicated than you think. First of all, the time received from the NTP server is the number of seconds elapsed since 1900. Second, the library used for time calculations uses Unix time — the number of seconds elapsed since 1970 — so NTP time must first be converted to Unix time. Third, the DCF77 protocol transmits the time of the next minute ("At the third stroke, the time will be…"), not the actual time. Finally, the NTP time received is probably not on a minute boundary. To correct for this we subtract two minutes from the received time and send them to the clock, followed by a third complete minute. This way we ensure that the clock receives enough data to be able to synchronize to it and extract the correct time.

Construction

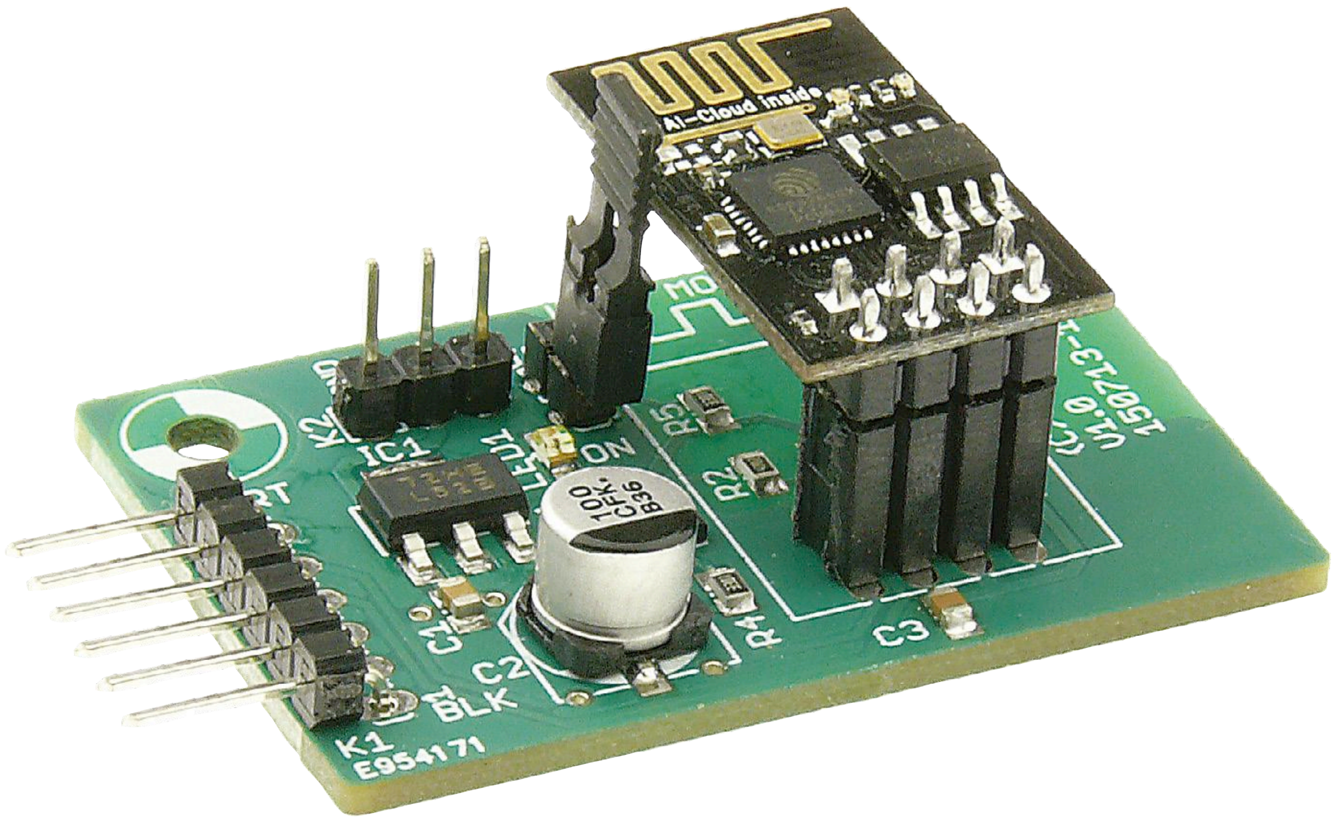
Although many parts are SMT (surface mount technology) types, soldering and assembling will not be too difficult. Even though the BOM includes an 8-way (2×4) socket strip for mounting the ESP-01 module, to save space (height) and improve mechanical stability it is preferable to solder the ESP-01 directly on the main PCB without this socket. You can also temporarily mount a socket and remove it once you are sure that everything is working fine.

The sketch has to be adapted to your setup

After assembling the board, it is time to program the ESP-01 module with the Arduino sketch ESP8266_NTPtoDCF available as a free download from [1]. If you have never programmed an ESP8266-based module before with the Arduino IDE, it is necessary to install the ESP8266 Boards Pack age first. More on this subject can be found on [2] and many other websites, as well as in articles.

Three things need to be set up to make the sketch work with your DCF77 clock:

1. the credentials for your Wi-Fi network;
2. the URI of the time server used for synchronization;
3. the polarity of the DCF77 emulator's output.



The ESP-01 needs to connect to your Wi-Fi network to enable it to collect time information from an NTP server. To make this work the credentials for the Wi-Fi network must be entered at the top of the sketch:

```
char ssid[] = "your_network_name";
   // your network SSID (name)
char pass[] = "network_password";
   // your network password
```

The URI of the NTP time server must be defined too:

```
const char* ntpServerName = "0.nl.
   pool.ntp.org";
```

In this case a server for the Dutch (NL) time zone is specified, but you may prefer another one.
The polarity of the DCF77 emulator output signal is defined in the function `DcfOut`:

```
case 0:
  if (PulseArray[PulseCount]!=0)
    digitalWrite(LedPin,0);
  break;
case 1:
  if (PulseArray[PulseCount]==1)
    digitalWrite(LedPin,1);
  break;
case 2:
    digitalWrite(LedPin, 1);
  break;
```

Shown here is the active-high version (output normally logic low, pulses are logic high). For an active-low output signal (output normally logic high, pulses logic low) invert the '0' and the '1's of the three `digitalWrite` commands.

Programming the ESP-01

Before we continue, an important remark first: never connect power to K2 when a USB-to-serial converter is powering the circuit through K1 (or the other way around). Disrespecting this No-No will short two power supplies, and may damage your DCF77 clock, your computer or both. Don't come crying to us.
Close jumper JP1 and connect a 3.3-V FTDI-cable-compatible USB-to-serial converter between K1 and your computer. JP1 must be closed at power on to switch the ESP-01 module to flash (programming) mode. In the Arduino IDE, on 'Tools' menu, select 'Generic ESP8266 Module' as board type and the correct COM port number of your USB-to-serial interface. Compile and upload the sketch. When the upload is completed open the 'Serial Monitor' in the Arduino IDE (again in the 'Tools' menu). The ESP module will echo useful information to your computer screen to check if the DCF77 simulator is doing its job correctly. First it will show if the ESP-01 is able to connect to your network. If it isn't, verify that you entered the correct network SSID and password in the sketch.
Once a connection has been established, the time server is read and decoded time information is displayed. Please note that the time on your screen is two minutes fast — this lead is needed to synchronize the DCF77 clock in time!
If the connection to the time server is not made, you may have misspelled its URI. Correct it in the sketch.
Every time you make modifications to the sketch it must be recompiled and reprogrammed into the ESP-01 module. Remember to power cycle the circuit (unplug and plug K1) to put the module back in programming mode.

Install the emulator in the clock

Once the information in the Serial Monitor is correct, the circuit is ready to be installed in your DCF77 clock.
Remove the programming cable from K1 and open jumper JP1. Remove the old DCF77 receiver from your clock. In many cases it will be a separate module with three wires (power, 0 V and DCF77 signal). Connect K2 of our circuit to the now unconnected input of the clock.
If you want to be sure that the output of our DCF77 emulator produces a valid DCF77 encoded signal, there are numerous Arduino-based (test) projects on the Internet with DCF77 decoders. We tested our prototype with the sketch found at [3].

(150713)

Web Links

[1] www.elektormagazine.com/labs/dcf77-emulator-with-esp8266-elektor-labs-version-150713

[2] https://github.com/esp8266/Arduino

[3] https://arduino-hannover.de/2012/06/14/dcf77-empfanger-mit-arduino-betreiben/

[4] www.elektormagazine.com/150713

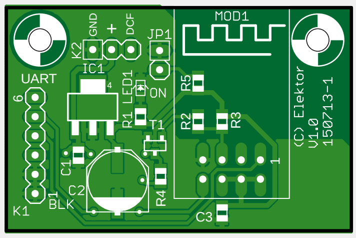
COMPONENT LIST

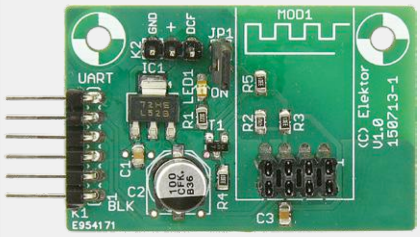
Resistors
All 0805, 5%, 0.1 W
R1 = 220Ω
R2 = 47kΩ
R3,R4,R5 = 3.3kΩ

Capacitors
All 0805
C1 = 470nF
C2 = 100µF, 16V, radial can SMD, 6.3mm diameter
C3 = 100nF

Semiconductors
IC1 = LM3940IMP-3.3/NOPB
LED1 = LED, red, 0805
T1 = BC847C

Miscellaneous
JP1 = 2-pin pinheader, 0.1" pitch
Jumper, 2-way, 0.1" pitch
K1 = 6-pin pinheader, 0.1" pitch
K2 = 3-pin pinheader, 0.1" pitch
MOD1 = ESP-01 Wi-Fi module
MOD2 = 8-way (2x4) pinheader socket, 0.1" pitch
PCB 150713-1 v1.0





FROM THE STORE

→150713-1
bare DCF77 emulator PCB

→150445-91
ESP-01 module

Multi-Timer

Switches up to 16 channels

By **Willem Tak** (The Netherlands)

Timers — you can't have enough of them. Turn on the aquarium lights, simulate occupancy while you're away, heat up a frozen pizza in time for when you get home, no matter what you come up with, you can probably do it with a timer.

A timer should be a piece of cake for the average electronics hobbyist. But once you've built a few of them, you'll be making the same circuit over and over again for friends and family. Not very enjoyable really, and it certainly isn't much of a challenge. The author therefore decided to design a timer "to end all timers" — with a predictable outcome: he is now busy building the fifth or sixth one, and orders keep coming in.

To start with: eight channels

There are now two versions of the multi-timer: one with eight channels and one with sixteen channels. We begin with the first one. Each of the eight timers drives an output relay. They can be configured independently to turn on within a 24-hour period, with a minimum on-time of 1 minute and a maximum of 23 hours and 59 minutes.

It is of course possible to turn on a load several times per day, simply by combining the outputs of several output relays.

The timer always knows the correct time and date thanks to the built-in GPS unit. Furthermore, each timer has a selectable mode whereby different switch times can be set up during the weekend.

A combined rotary encoder/switch (Alps EC11B152442D) is used for the operation and configuration of the timer. An OLED display completes the timer.

It is possible to connect a light sensor that influences the behavior of the timer.

Hardware

The circuit for the eight-channel timer is shown in **Figure 1**. This isn't overly complex because most of the intelligence is hidden inside the microcontroller. The author has chosen an 18F4685 for the microcontroller, mainly because of its large RAM capacity. The microcontroller runs at 22.1184 MHz in order to provide a straightforward interface to the GPS unit. This is an EM-408, which can be bought quite cheaply via the Internet; it is also possible to use a different receiver, such as the one from the Elektor Store [1]. In any case, the EM-408 requires a supply voltage of 3.3 V, which necessitates the use of a 74HCT00 level-shifter between the receiver and the microcontroller.

The OLED display uses a standard interface, and there is no need to explain this in further detail. The brightness of the display can be adjusted using a potentiometer that is connected to the analog input of the microcontroller.

A photocell (Sparkfun SEN-09088) is connected to another analog input; it is used to (amongst other things) turn on an LED when the ambient light exceeds a certain level, and turn it off when it is dark. The light level at which it switches can be set using a potentiometer.

A status LED has been mounted on the board, which comes to life when the circuit is turned on. It subsequently changes state every time that a valid GPS string has been received.
For the actual switching of the loads we've used relays (Finder 34.51.7.012.0010) that can switch a current of up to 6 A at 230 VAC. Since the outputs of the microcontroller cannot drive the relays directly, a ULN2803 (IC3) driver has been included.
And last, but not least, is the power supply. The timer uses a 12 V supply; this voltage is also used to power the relays. Two three-pin switch-mode regulators turn the 12 V into a voltage of 3.3 V for the GPS unit and a voltage of 5 V for the rest of the electronics.

Firmware

Because the author decided to put all functions into a single microcontroller, the firmware has become quite complex — it's certainly too large to print it in this magazine. If you're interested in having a closer look at the firmware, you can download it free of charge from the project page for this article [2]. The time supplied by the GPS unit is used to ensure that the internal clock is set correctly; from then on, the timer uses its internal clock. The software has several routines that check that the internal clock doesn't deviate (too much) from the GPS time.
The eight timer blocks are controlled using a windowRAM of 1440 bytes in RAM. Each of these bytes corresponds to a specific minute of the day. Each of the eight timers has one bit reserved in this byte; a 0 in that position means that the relay will be turned off, a 1 ensures that it will be turned on. The software has a routine that converts the time into the actual minute of the day, which is then used to access the correct windowRAM address.

The program has a large number of lookup tables — one example is a table that keeps track when Saturdays, Sundays and holidays occur.

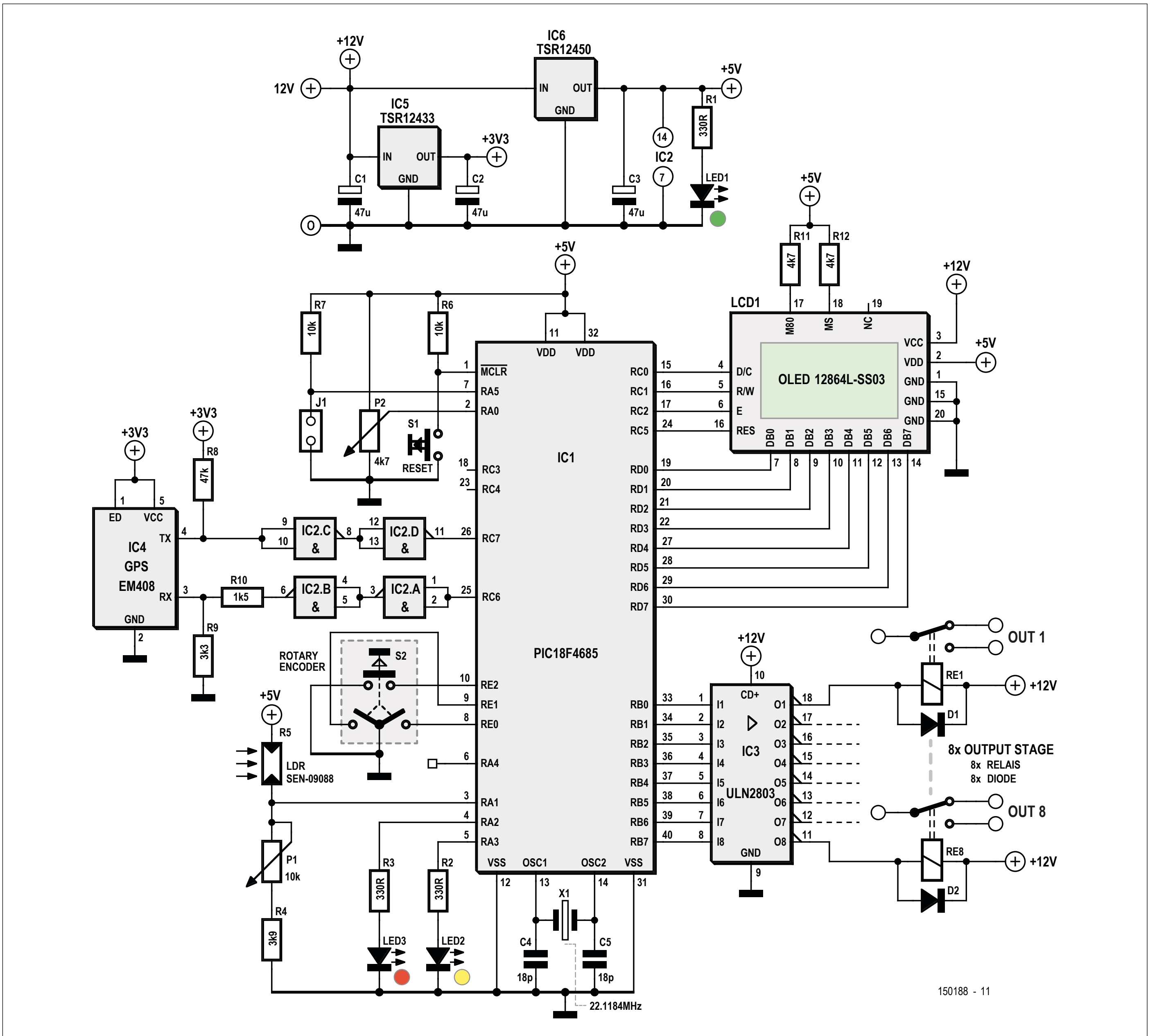


Figure 1. The complete hardware for the 8-channel timer.

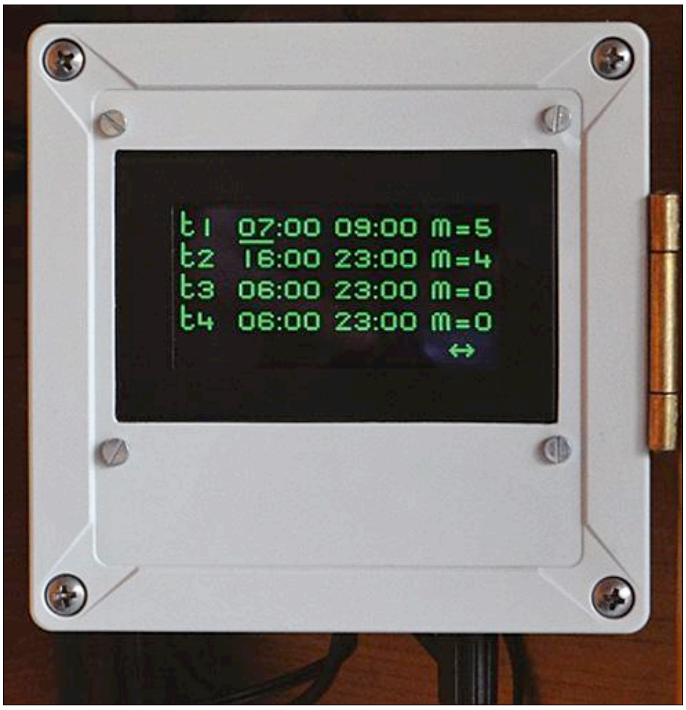


Figure 2. A smart enclosure gives the 8-channel timer a polished look.

Operating instructions

This timer has so many features that we could fill a large number of Elektor pages describing them. However, we decided against this. Instead, a comprehensive instruction manual forms part of the download [2].

An example of a timer completed by the author can be seen in **Figure 2**.

The 16-channel version

If the eight-channel version was *the timer to end all timers*, then we can safely call the sixteen-channel version *the timer to end the timer to end all timers*...

But enough of the jokes — the circuit diagram for this version is shown in **Figure 3**. The circuit is broadly similar to the smaller version, although there are sufficient differences to warrant a separate description.

As the name implies, this circuit has 16 timers that each drive a relay. Each timer can be configured independently to turn on within a 24-hour period, with a minimum on-time of 1 minute and a maximum of 23 hours and 59 minutes. Again, we can combine outputs so that we can program multiple switch times for a load. The time and date are also taken from a GPS unit. The time zone can now be configured. Each timer can be configured so that it behaves differently during a weekend.

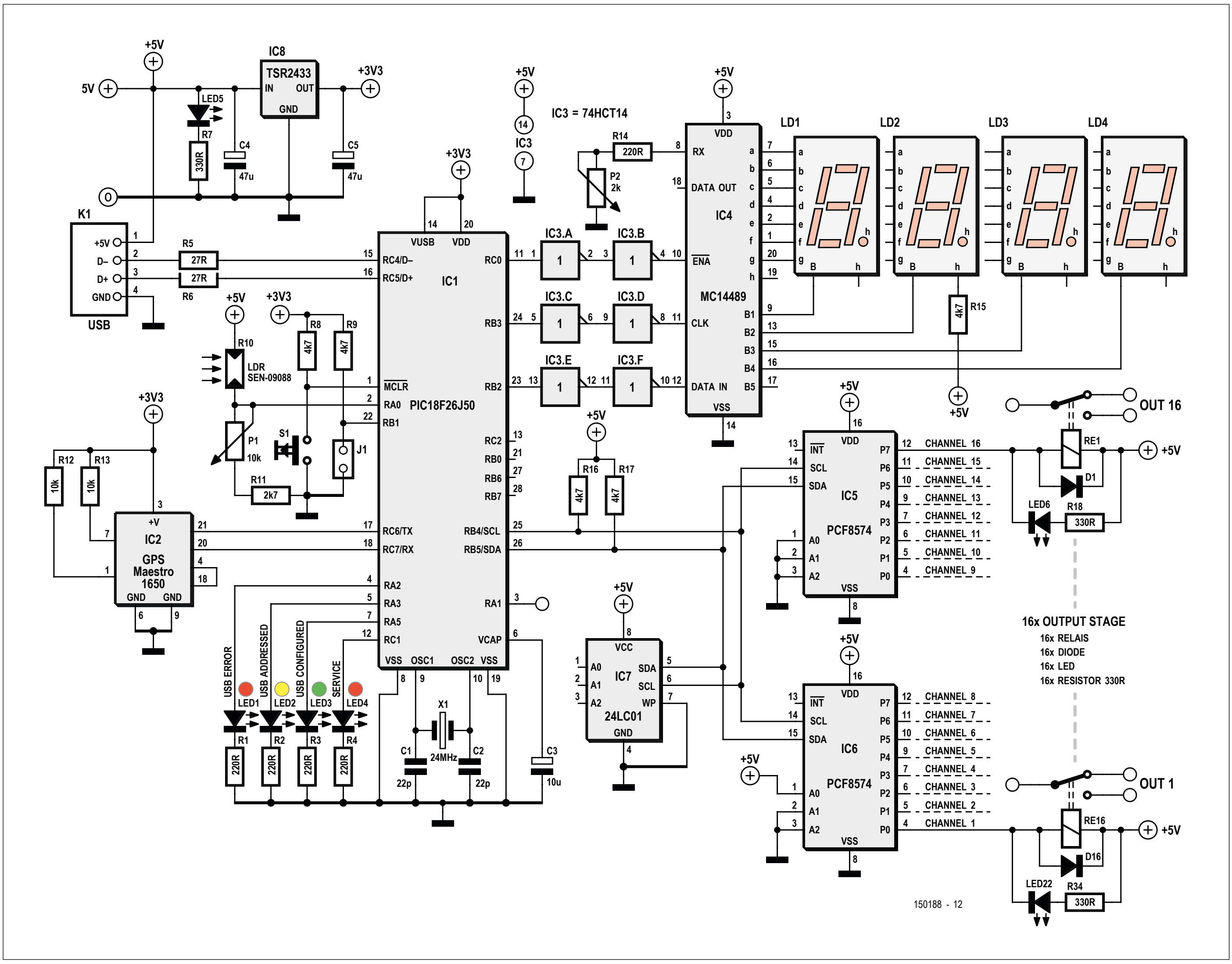


Figure 3. The circuit for the 16-channel version looks more complex than the 'smaller' version, but appearances can be deceptive.

The timers are programmed using a PC or laptop via a USB connection. The timer has a 4-digit 7-segment LED display that shows the time, as well as 16 individual LEDs that show the status of the outputs. You can add a photocell to influence the behavior of (certain) timers.

Hardware

At the heart of the circuit is another microcontroller, but this time it's a PIC 18F26J50. As with the smaller version, it was chosen for the large amount of RAM that it contains. The clock frequency is 24 MHz (because of the USB connection). The GPS unit is now a Maestro 2035, which requires a supply of 3.3 V. It can therefore be connected directly to the RS232 lines of the microcontroller. Regarding the RS232: a clock frequency of 24 MHz results in RS232 baud rates that deviate slightly from their proper values, although the differences remain within the permitted tolerances.
The photocell is connected to an analog input of the microcontroller; this is (also) used to drive a LED that turns on when there is enough ambient light. The switch point can be adjusted using a potentiometer.
A connector is provided for the USB connection to the PC/laptop. There is a separate connector for the 5 V power supply for the timer. Please note that you must remove the external 5 V supply before connecting the timer to the PC/laptop when you want to program it! The 3.3 V for the microcontroller and the GPS unit is derived from the 5 V supply by a TSR12433 (a three-pin switch-mode converter).
The programming data for the timer (created on the PC) is stored in an external EEPROM (a 24LC01). The PIC version used here doesn't have an on-chip EEPROM — instead, Microchip decided to use something called HEF (*High Endurance Flash*), where part of the program memory is reserved for permanent memory. This does result in a few disadvantages: a memory location needs to be wiped clear before new data can be written to it, and this must be done in blocks of at least 1024 bytes. In computer terms this process takes ages — it's much too long if you also need to keep a USB connection going.
Because the I²C lines of the microcontroller, which are connected to the external EEPROM, are 5 V tolerant, they can be connected without the need for any level converters. There are also two old favorites connected to the I²C bus: PCF8574 port expanders. These are able to drive the relays directly (from the same series as those used in the 8-channel version).
The display, consisting of four standard 7-segment LED displays with a common cathode, is driven by a MC144489 display driver IC. The brightness of the LED display can be adjusted using a potentiometer that's connected to the IC. The driver is controlled using three lines from the microcontroller. A 74HCT14 level shifter is used to increase the 3.3 V level from the microcontroller to the 5 V level required by the driver IC.

Firmware

The software also turned out to be complex in this version, since everything is taken care of by a single microcontroller. This time the software has to perform two tasks: one part is used for the USB connection (and hence the programming of the timer); the other part is used to take care of all the normal timer functions.
The operating mode of the timer is determined by how the circuit is powered. When a USB cable from a (running) PC or laptop is connected, the software will attempt to start the USB enumeration process. When the timer has been recognized as a HID, there will be an active connection to the PC/laptop. The program Timer_16 (also part of the de download) should then be started on the PC and you can then program the timer. The programming interface is shown in **Figure 4**.

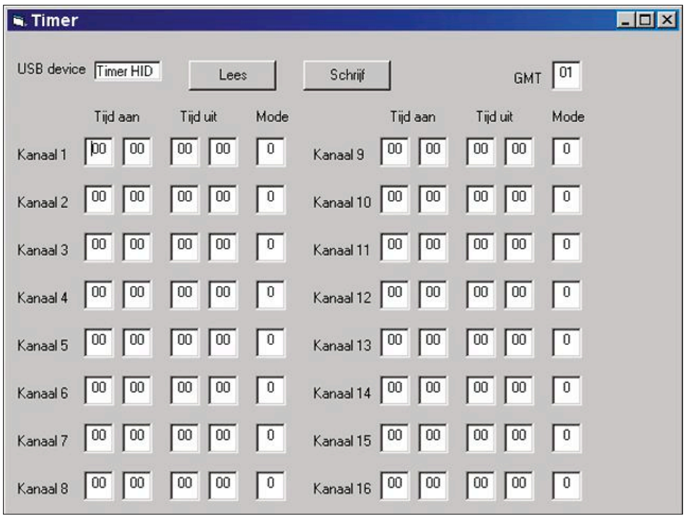


Figure 4. Interface for configuring the 16-channel timer.

When the normal 5 V supply is connected, the enumeration process will obviously fail and the program will run in the mode that deals with the normal timer functions.
It should be clear that the 5 V supply and the USB cable must not be connected at the same time!
The rest of the firmware operates the same way as in the 8-channel version, except that there are now two windowRAM blocks defined, each with 1440 bytes.

Operating instructions

The 16-channel version also has more pages in its operating manual than we can publish in this magazine. The extensive manual is therefore part of the (free) download for this article [2].

(150188)

Web Links

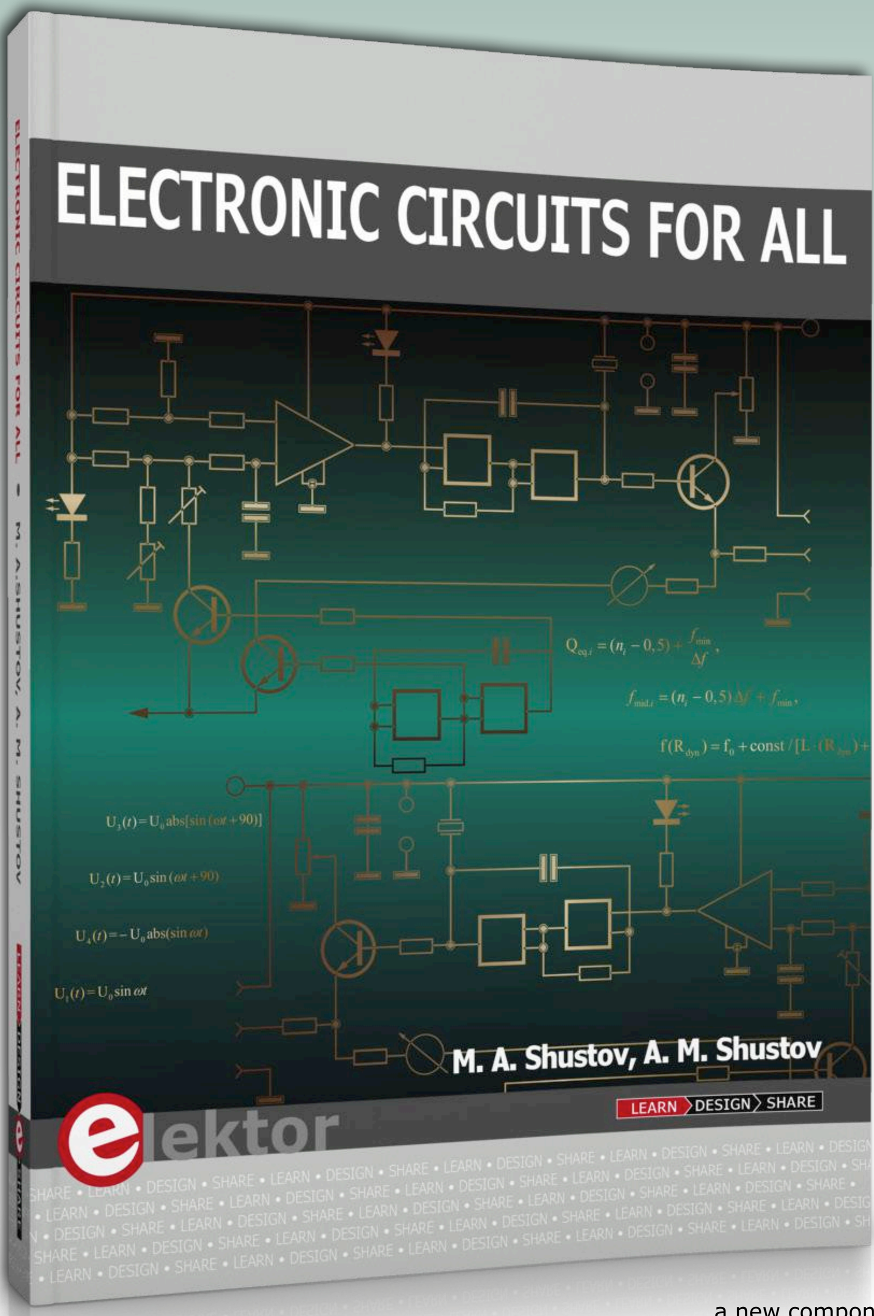
[1] www.elektor.com/gps-board-eb056

[2] www.elektormagazine.com/150188

STORE HIGHLIGHT

From Russia with Love...

400+ circuits on 400 pages



By **Eric Bogers** (Elektor Netherlands Editor)

In the present-day, microcontroller-driven age, it's easy to forget that there once was a time when electronic circuits consisted of discrete components that were easy to handle and could be soldered without specialised equipment...

Probably driven by nostalgic feelings, Michael Shustov and his son Andrej have put together the book *Electronic Circuits for All*, a collection of more than four hundred small and simple circuits that use conventional ('old-fashioned') components — actually somewhat similar to the style of the *3xx Circuits* series (where *xx* = 00 through 11) from Elektor and with an equally varied contents: there is something for everyone, including power supplies, test equipment, generators, filters, (tele)communications, amplifiers and security. And it is therefore not much of a surprise that Elektor published this (English-language) book for the European market in 2017.

Baristor

In 2016 father and son Shustov, during the *Fast Forward Award* event at the *electronica* trade show in Munich, received a prize for their *baristor* (a contraction of *barrier resistor*), a new component, comprising conventional parts, that behaves as an open-circuit below a certain voltage and as a normal resistor above that voltage (or the other way around) — see **Figure 1**. It will not be a surprise that this component receives extensive coverage in the book; **Figure 2** shows a typical example.

Minimalistic

The circuits in the book give a minimalistic impression. Decoupling capacitors, protection resistors or inputs of unused CMOS-gates that are connected to ground — these things are absent. The authors have apparently sufficient confidence in their readers that they will add these ‘automatically’. They are therefore not real DIY schematics — after all, when we quickly scribble something on a piece of paper we don’t always meticulously include all these refinements...

Language use

When reading, there forms the impression that the English-language version of the book was written or translated by someone who uses Russian, not English, as their native language. Perhaps the authors have provided their own English-language version or a translator, who goes unmentioned on the colophon page, has worked on it — whatever the case may be, language purists will be rubbed the wrong way from time to time, but the mistakes (for example ‘capacity’ instead of ‘capacitance’) and some awkward phrasings fortunately do not stand in the way of an understanding of what has been written.

The return of the TUP and TUN

We have already mentioned that this book does remind us of the 300-series from Elektor – and this is also true for the selection of the components. As an example, for the transistors, the authors invariably choose the BC547 and BC557 (NPN and PNP respectively). These were once the workhorses of electronic circuits, everyone used these for just about anything imaginable. The use of these *Transistor Universal PNP* and *Transistor Universal NPN* does, in any case, evoke nostalgic feelings, and if you do not happen to have a box of these around, then you can, in good conscience, substitute a modern version. The same is true for all the other components that have been used: nothing exotic, only standard parts that everyone has on hand. This facilitates the building of the circuits in the book without first having to make a trip to the electronics store around the corner — sorry, that was still true during the seventies of the previous century; speaking of nostalgia — so: without first having to order the correct parts from some Internet merchant.

Esoteric

In addition to a large number of extremely practical circuits, the authors also devote some attention to more esoteric subjects: Kirlian photography, thunderstorm prediction, electrotherapy, aerion detection (whatever that may be) and “psycho-emotional correction” (**Figure 3**) — subjects from the flower-power age that also used to grace the pages of Elektor once. The utility of those contributions can be debated, but in any case they are a good excuse for spending a rainy Sunday afternoon with a soldering iron.

Conclusion

Despite a few minor criticisms, we may thank the authors that they have made this collection of circuits ‘from the past’ available to 21st-century electronics enthusiasts, as proof that a µC or an RPi is not always an absolute necessity. Ladies and gentlemen soldering devotees: dig out the box with ‘old stuff’ and heat up that soldering iron!

(160394)

Web Link

www.elektor.com/electronic-circuits-for-all

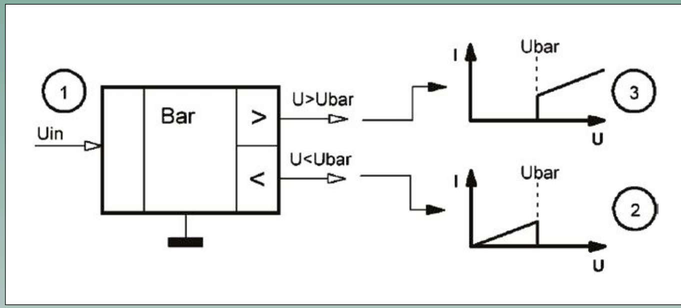


Figure 1. The schematic symbol of the ‘baristor’.

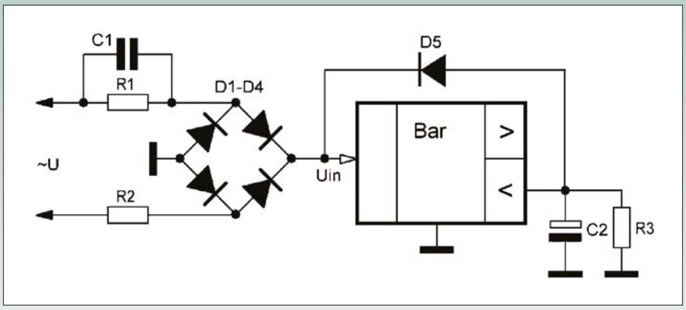


Figure 2. Transformer-less power supply using a baristor.

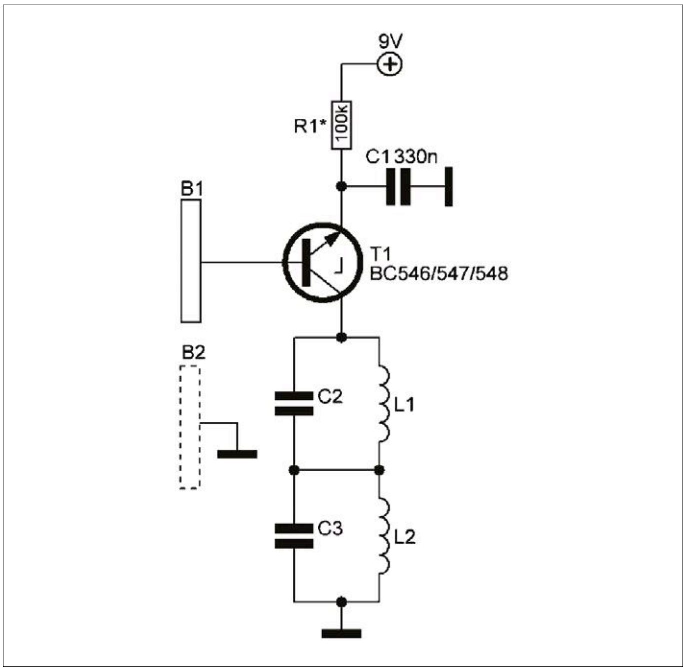


Figure 3. Circuit for psycho-emotional correction. Whoever knows how it works may speak up...

ESP32 Low Power

Programming the ULP coprocessor

By **Tam Hanna** (Slovakia)

Many developers who work with the ESP8266 are unhappy with its high current consumption. The ESP32 offers a remedy for this — it is equipped with a coprocessor designated ULP, which has considerably less functional scope but needs less power. However, the ULP processor requires programming in assembly language, and here we show you how.

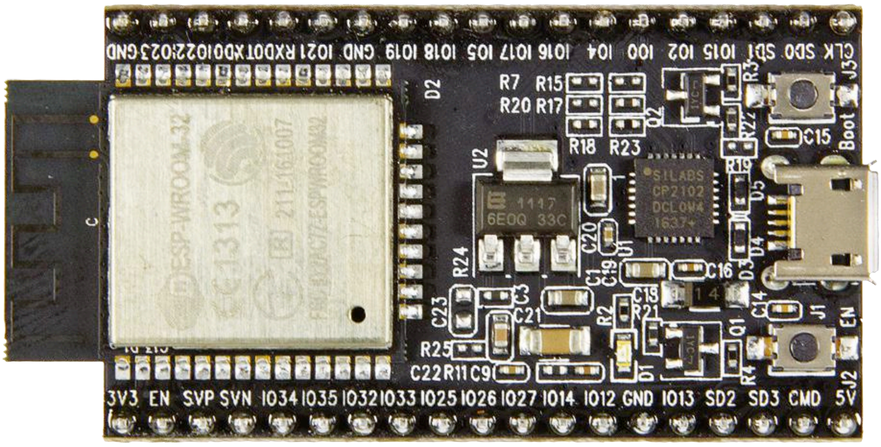


Figure 1. The ESP32 DevKitC is a handy development board.

In this article we use the ESP32 DevKitC development board (**Figure 1**), which comes with a micro USB connector and is conveniently available in the Elektor Store [1].
In a previous Elektor article [2] we described how to get started with the ESP32. In the present article we provide an introduction to using the ULP processor.

Preparations
Like it or not, the manufacturer Espressif regularly updates its programming framework, which goes by the name IDF. As a lot of water has passed under the bridge since the publication of the last ESP32 article in Elektor Magazine, we recommend that you get the current version of the framework before proceeding further. More information about this is available at [3].

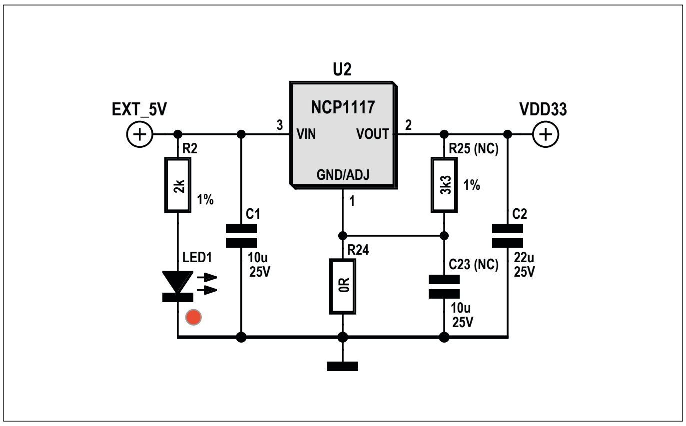


Figure 2. Part of the power supply circuit.

As usual, you should bear in mind that the ESP toolchain is allergic to space characters. For example, the 'Documents and Settings' folders repeatedly found in older versions of Windows always cause problems, and unfortunately the manufacturer's forums do not show much appreciation of this.
For users of the SparkFun board discussed in the previous article [2], there is a small change here: instead of the previously used FTDI converter, the DevKitC board employs the CP210x, which is considerably cheaper and provides comparable performance. The driver should be present already, but if for some reason your PC has never heard of it, you can find archives at [4] where you can obtain the driver.
When you connect the board to your computer through a cable with a micro USB connector, the red LED will light up immediately. Under Ubuntu you can see the path of the virtual serial port in the system log. On the author's computer, the DevKitC logs in as */ttyUSB0*, just like its predecessor:

```
tamhan@TAMHAN14:~$ dmesg
[19280.368821] usb 1-1.7: Product: CP2102 USB to UART
    Bridge Controller

. . .

[19280.399519] cp210x 1-1.7:1.0: cp210x converter
    detected
[19280.399840] usb 1-1.7: cp210x converter now
    attached to ttyUSB0
```

As the user name and other details vary from one PC to the next, the ESP programming environment expects to see environment variables. If you work under Unix, you can define the two variables as follows:

```
tamhan@TAMHAN14:~/esp$ export IDF_PATH=~/esp/esp-idf
tamhan@TAMHAN14:~/esp$ export PATH="$PATH:$HOME/esp/
   xtensa-esp32-elf/bin"
```

These two commands must be entered manually in a (newly opened) terminal window, unless you want to change the overall settings of your PC.

Demo program without ULP

For our next task, let's see if we can measure the current consumption. The Hello World demo program, which is kindly provided by Espressif as part of the development environment package and copied to your computer, is suitable for this. Under Unix you can access it by entering the following pair of commands. The first command changes to the home directory, and the second command copies the project code from the *examples* folder to the folder of the ESP32 programming environment:

```
tamhan@TAMHAN14:~$ cd ~/esp
tamhan@TAMHAN14:~/esp$ cp -r $IDF_PATH/examples/get-
   started/hello_world .
```

Please note that the dot at the end the `copy` command line is not a mistake by the Elektor layout department. In Unix an isolated dot stands for the current directory, which means that this command tells the operating system to copy the *examples* folder to the current working folder.

Next you have to define some settings so the compiler will be able to do its job properly. To make the necessary settings, you can use the `menuconfig` command, which you invoke as follows:

```
tamhan@TAMHAN14:~/esp$ cd hello_world/
tamhan@TAMHAN14:~/esp/hello_world$ make menuconfig
```

After the automatic compilation of several auxiliary files, you should navigate to *Serial flasher config → Default serial port* to specify the port to be used. Under Unix the default setting is often correct.
Then save your settings and close `menuconfig`. Next, use the `make` command to compile the ESP32 firmware. A nice aspect of this for developers is that `make` is generally very good at parallel processing:

```
tamhan@TAMHAN14:~/esp/hello_world$ make flash -j8
```

The parameter `-j8` passed with the command here tells the tool to use eight threads. With the author's eight-core workstation, this results in virtually ideal system utilisation. The `make flash` command automatically downloads the compiled code to the ESP32.

If you look at the code of the Hello World demo program, you will see a lot of instances of the `printf` command:

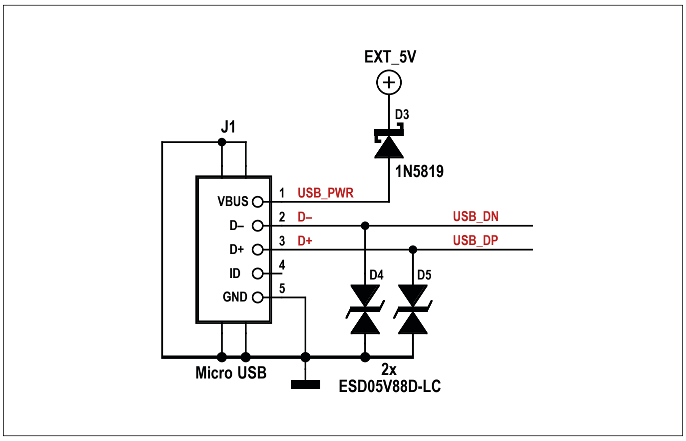


Figure 3. You can block the supply voltage from the USB port by applying a higher voltage to diode D3.

```
void app_main()
{
    printf("Hello world!\n");
    esp_chip_info_t chip_info;
    esp_chip_info(&chip_info);
    printf("This is ESP32 chip with %d CPU cores,
   WiFi%s%s, ",
    . . .
```

You can group the output with an existing USB connection by entering "`make monitor`". To exit the work environment, press the key combination *Ctrl+]*.

Current consumption

Now we're ready to measure the current consumption. The schematic diagram of the DevKitC can be viewed at [5]. The portion of the diagram in **Figure 2** shows that the required 3.3-V supply voltage is generated by an LDO voltage regulator.
In theory you should disconnect the USB cable from the computer at this point to break the supply connection, but then you would also lose the debugger connection. Fortunately, there's an alternative. **Figure 3** shows how the USB connector is wired on the board. If you do not need an especially accurate measurement, you can use D3 to block the supply voltage from the PC. Connect a lab power supply and an ammeter to the board as shown in **Figure 4**.

Kaboom!

If (like the author) you use an HP 6624A lab power supply in your test setup, you must remember to physically switch off the power supply. When you press the Output Disable button, a significant displacement current (which can be as high as 5 A) flows from the USB port to the power supply — which is more than a lot of instruments can handle. The Solartron 7150 has a 2-A fuse, which is usually more than adequate, but if you are using a small multimeter with a maximum range of 200 mA, you will be looking for a new fuse.

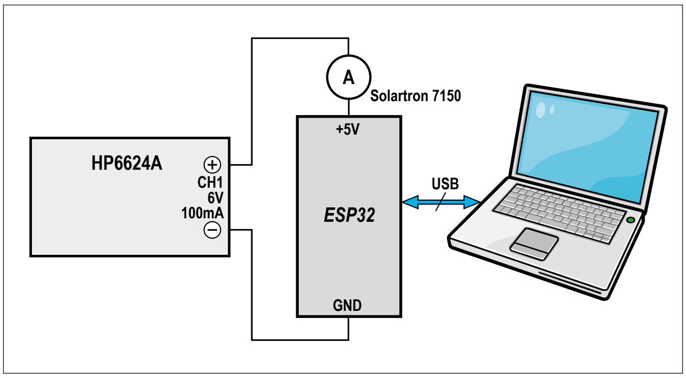


Figure 4. This arrangement can be used to measure the current consumption.

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11b, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11g, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

Figure 5. WLAN is not especially low-power (source: Espressif ESP32 data sheet [6]).

With the right setup, the author's Solartron 7150 ammeter measures a current consumption of about 48 mA. As the DevKitC board has an LED and some other bells and whistles, if you design your own board you should be able to manage with less. According to the data sheet, the board draws between 30 and 50 mA in this state, which is designated "Modem Sleep". If you reduce the CPU clock to 80 MHz, you only need 20 to 25 mA, and at 2 MHz the current consumption drops to 2–4 mA.
Of course, these figures are only true as long as the WLAN transmitter is disabled. When the radio module is active, the current consumption depends on the operating mode of the ESP32. Measurement is difficult in that situation, so **Figure 5** shows some information from the datasheet [6].

ULP joins the fray

The Arduino Yun popularised the idea of combinatorial process computers. This idea is both simple and clever: the microcontroller consists of various modules that are activated as necessary.
In the case of the ESP32, along with the main CPU (designated Xtensa) there is an auxiliary module designated RTC that contains a special processor called ULP, which stands for "ultra low power". That is a sort of lower-tier device equipped with its own SRAM working memory, which can operate independently of the main processor and can even access peripheral functions. The underlying idea is that developers can shut down the main processor if necessary and delegate monitoring to the very low-power RTC module. If it determines that something has happened that needs attention, it starts up the high-performance processor. Thanks to the segmented memory space, it is also possible to share data between the RTC and Xtensa sides.

If you want to learn more about the RTC module, you won't get much help from the data sheet. In Espressif's view, the detailed technical description belongs in the reference manual, which (at the time of printing of this article) is available for download at [7]. You can also find more information about the structure at [8], and you can use that document as a skeleton for your own projects.
There you can also learn that the ULP processor is a finite state machine equipped with one full-width working register and three half-width working registers. It also has a very unusual instruction set, so it can only be programmed in assembly language (C programming is not possible).
Now let's create a small demo program that shows the interaction between the ULP processor and the main processor. For this we use a shared counter variable that is used similarly by both parts of the ESP32.
To do this, we extend the Hello World demo program in five steps. In the first step we create a subdirectory *ulp* in the directory *main*, which is initially left empty. Then we open the file *main/component.mk* and adjust its content as follows:

```
# . . .
# (Uses default behaviour of compiling all source
   files in directory, adding 'include' to include
   path.)
ULP_APP_NAME ?= ulp_$(COMPONENT_NAME)
ULP_S_SOURCES = $(COMPONENT_PATH)/ulp/
   ulp_source_file.S
ULP_EXP_DEP_OBJECTS := main.o
include $(IDF_PATH)/components/ulp/component_ulp_
   common.mk
```

If you are working with a ULP program, you can simply copy the *make* snippet provided by Espressif to your own project. It uses the file *ulp_source_file.S* located in the subfolder / *ulp* and includes it in the compilation process.
After generating the assembly language file, you have to include the header. That's because the assembler in the Espressif toolchain uses a preprocessor, which would otherwise spit out various error messages:

```
#include "soc/rtc_cntl_reg.h"
#include "soc/rtc_io_reg.h"
#include "soc/soc_ulp.h"
```

The rest of the first assembly language file looks like this:

```
.bss

   .global acti_count
acti_count:
   .long 0
```

Here there are two significant passages. The first is the *.bss* passage (here empty). If it is missing, the ULP processor will find the file but will simply refuse to execute it, without telling you why. That is followed by a sequence that creates a global variable, which means a variable shared between the Xtensa processor and the ULP processor. Now you can turn your attention to the actual code:

```
    /* Code goes into .text section */
    .text
    .global entry
entry:
    move r3, acti_count
    ld r0, r3, 0
    add r0, r0, 1
    st r0, r3, 0
```

The ULP processor is a sixteen-bit device. For our purposes that is only relevant when part of a register is ignored when it is read. In the first step we push the address of `acti_count` onto the half-width working register R3 and then use it to load the value. That is incremented by `add` and then written back to `acti_count`.
Finally, we wake up the Xtensa processor. This confronts us with a small problem: `wake` commands can only be accepted after the core processor has been shut down. Our code therefore issues `wake` only if the SOC has confirmed via the appropriate register that it is in sleep mode.

```
    /* Check if the SoC has said INRI already */
    READ_RTC_REG(RTC_CNTL_DIAG0_REG, 19, 1)
    and r0, r0, 1
    jump exit, eq
    wake
    WRITE_RTC_FIELD(RTC_CNTL_STATE0_REG, RTC_CNTL_
   ULP_CP_SLP_TIMER_EN, 0)
    halt

exit:
    halt
halt
```

On the C side

Now we go back to *hello_world_main.c*, where a number of headers are included:

```
#include "esp_sleep.h"
#include "nvs.h"
#include "nvs_flash.h"
#include "soc/rtc_cntl_reg.h"
#include "soc/rtc_io_reg.h"
#include "soc/sens_reg.h"
#include "soc/soc.h"
#include "driver/gpio.h"
#include "driver/rtc_io.h"
#include "esp32/ulp.h"
#include "ulp_main.h"
```

ulp_main is generated by the *make* file. That file provides content that is necessary for execution of the ULP program or for calling it from the C side.
A number of basic considerations are necessary here. The Xtensa part of the microcontroller has extensive working memory, and it takes a lot of power to keep it alive. In the interest of maximum energy savings, Espressif cuts off power to the working memory in sleep mode, with the result that its content is lost. The first official action of our program is therefore to check the wakeup cause:

```
void app_main() {
    //Init check
    esp_sleep_wakeup_cause_t cause =
   esp_sleep_get_wakeup_cause();
    if (cause != ESP_SLEEP_WAKEUP_ULP) {
        printf("Clean boot\n");
        init_ulp_program();
    } else {
        printf("Start caused by ULP\n");
        printf("Counter %u \n", ulp_acti_count  &
   UINT16_MAX);
    }
```

Here it is important to mask the value contained in `ulp_acti_count`. This register, like all registers of the Xtensa, is 32 bits wide. As the ULP processor can only write the first sixteen of these bits, the 'undefined' remainder must be masked.
Entering sleep mode is a two-step process. In the first step we instruct the power management logic to accept start

Too complex?

If you want to avoid working at the *make* level or you are working with ancient demo code, an alternative is to use the macro function (which has since been discontinued by Espressif). In that case the ULP program consists of an array of instructions:

```
const ulp_insn_t program[] = {
    I_MOVI(R3, 16),
    I_LD(R0, R3, 0),
    I_LD(R1, R3, 1),
    I_ADDR(R2, R0, R1),
    I_ST(R2, R3, 2),
    I_HALT()
};
```

That is followed by a block of more or less standard C code, which calculates the number of instructions in the program and then releases it for execution. Make sure you always perform the `size` calculation as shown here, because some macros are expanded to two separate instructions:

```
size_t load_addr = 0;
size_t size = sizeof(program)/sizeof(ulp_insn_t);
ulp_process_macros_and_load(load_addr, program,
   &size);
ulp_run(load_addr);
```

Figure 6. A module is missing here.

RTC GPIO Num	GPIO Num	Pad Name	Analog Function		
			1	2	3
0	36	SENSOR_VP	ADC_H	ADC1_CH0	-
1	37	SENSOR_CAPP	ADC_H	ADC1_CH1	-
2	38	SENSOR_CAPN	ADC_H	ADC1_CH2	-
3	39	SENSOR_VN	ADC_H	ADC1_CH3	-
4	34	VDET_1	-	ADC1_CH6	-
5	35	VDET_2	-	ADC1_CH7	-
6	25	GPIO25	DAC_1	ADC2_CH8	-
7	26	GPIO26	DAC_2	ADC2_CH9	-
8	33	32K_XN	XTAL_32K_N	ADC1_CH5	TOUCH8
9	32	32K_XP	XTAL_32K_P	ADC1_CH4	TOUCH9
10	4	GPIO4	-	ADC2_CH0	TOUCH0
11	0	GPIO0	-	ADC2_CH1	TOUCH1
12	2	GPIO2	-	ADC2_CH2	TOUCH2
13	15	MTDO	-	ADC2_CH3	TOUCH3
14	13	MTCK	-	ADC2_CH4	TOUCH4
15	12	MTDI	-	ADC2_CH5	TOUCH5
16	14	MTMS	-	ADC2_CH6	TOUCH6
17	27	GPIO27	-	ADC2_CH7	TOUCH7

Figure 7. This table lists the pins that can be accessed from the ULP processor (source: Espressif, [7]).

commands from the ULP processor. Then we start the ULP program and put the main processor to sleep:

```
    //Good night
    ESP_ERROR_CHECK( esp_sleep_enable_ulp_wakeup() );
    ESP_ERROR_CHECK( ulp_run((&ulp_entry - RTC_SLOW_
   MEM) / sizeof(uint32_t)));
    esp_deep_sleep_start();
}
```

Initialising the ULP program also involves several steps. We start by including two constants that are generated by the *make* file. Then we call `ulp_load_binary` to load the resources:

```
extern const uint8_t ulp_main_bin_start[]
   asm("_binary_ulp_main_bin_start");
extern const uint8_t ulp_main_bin_end[]
   asm("_binary_ulp_main_bin_end");
void init_ulp_program()
{
    esp_err_t err = ulp_load_binary(0, ulp_main_bin_
   start, (ulp_main_bin_end - ulp_main_bin_start) /
   sizeof(uint32_t));
    ESP_ERROR_CHECK(err);

    . . .
```

If you try to compile the program in its present state, you will receive the error message shown in **Figure 6** because the normally downloaded components do not include the modules required by the ULP (at least not when this article was printed).

You have to go to the portal at [9] and download the file corresponding to your platform. Then extract the archive to */esp* and add it to the path as follows:

```
tamhan@TAMHAN14:~/esp$ ls
esp32ulp-elf-binutils  esp-idf  hello_world
   xtensa-esp32-elf
tamhan@TAMHAN14:~/esp$ export PATH="$PATH:$HOME/esp/
   esp32ulp-elf-binutils/bin"
```

After this, go back to *make menuconfig* and activate the option *Component Config-Y ESP32-Specific → Enable Ultra Low Power (ULP) Coprocessor*. In response, *menuconfig* will include *(512) RTC slow memory reserved for coprocessor*, and this memory allocation will survive sleep mode.

Stop and go

If you run the program as it is at this point, you will notice that the counter is incremented by 3 each time. That is because the timer starts after `ulp_run` is called, and it takes a while for the main processor to shut down. Any wakeup calls received earlier will not wake it up; the ULP processor spends its time in a sort of endless loop that performs three increment operations.
To clear up this problem, all you have to do is to wait 'actively' by jumping back to the start if the main processor is still awake.

```
exit:
    /* Check if the SoC has said INRI already */
    READ_RTC_REG(RTC_CNTL_DIAG0_REG, 19, 1)
    and r0, r0, 1
    jump exit, eq
    wake
    WRITE_RTC_FIELD(RTC_CNTL_STATE0_REG, RTC_CNTL_
   ULP_CP_SLP_TIMER_EN, 0)
    halt
```

The author could not resist the temptation to see how long it takes for the main processor to shut down. Some of the GPIO pins of the ESP32 are implemented as multiplexed, which means they can be accessed by both the RTC module and the main CPU. The relevant pins are listed in the table shown in **Figure 7**.
These pins are initialised through a specific API designated by the prefix "RTC". Note that the pin IDs transferred here (on the ESP32 side) refer to ESP32 GPIO pins:

```
void init_ulp_program() { . . .
    ulp_acti_count=0;
    rtc_gpio_init(cpu_num);
    rtc_gpio_set_direction(cpu_num,
```

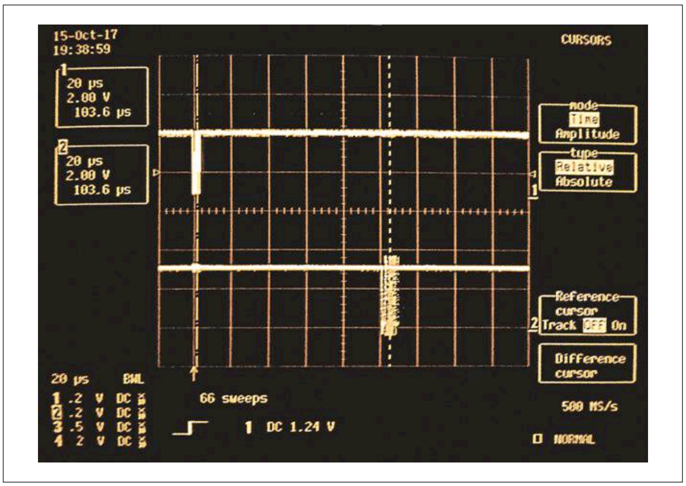


Figure 8. The elapsed time from the end of toggling on the Xtensa side (Ch. 1) to the start of program execution on the ULP side is about 100 µs.

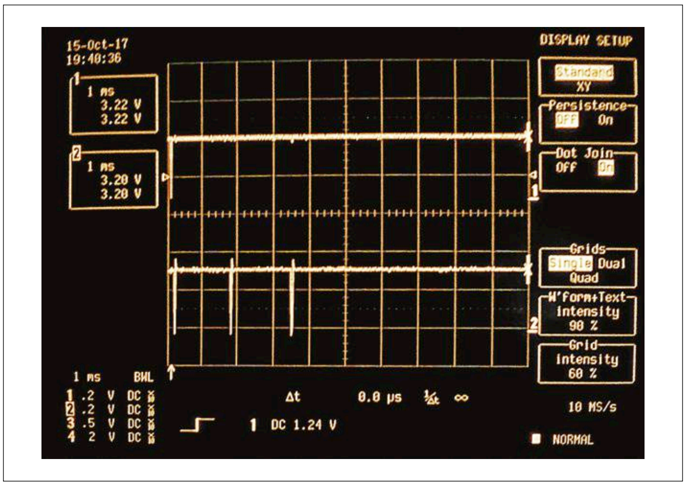


Figure 9. The three activations of the ULP routine are clearly visible on this screenshot.

```
  RTC_GPIO_MODE_OUTPUT_ONLY);
   rtc_gpio_set_level(cpu_num, 1);

   rtc_gpio_init(ulp_num);
   rtc_gpio_set_direction(ulp_num,
  RTC_GPIO_MODE_OUTPUT_ONLY);
   rtc_gpio_set_level(ulp_num, 1);
```

Now we place code between the `Init check` and `Good night` blocks that generates a characteristic waveform, which can easily be viewed on an oscilloscope:

```
//BlinkBlink
rtc_gpio_hold_dis(cpu_num);
char bctr=0;
while(bctr<4){
    rtc_gpio_set_level(cpu_num, 0);
    rtc_gpio_set_level(cpu_num, 1);
    bctr++;
}
rtc_gpio_hold_en(cpu_num);
```

Espressif equips the GPIO pins with a function called “Hold”. By setting the corresponding flag, you instruct the GPIO engine to hold the state of the pin concerned during the transition between ULP and ESP32.

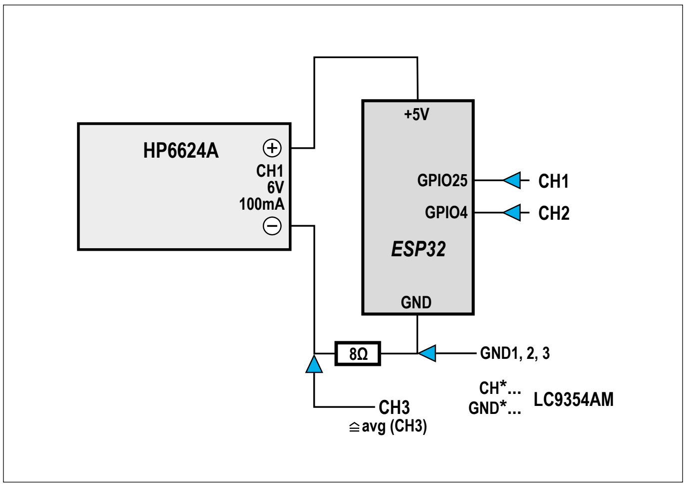
Next we have to make the ULP part of the ESP32 visible on the oscilloscope. For this purpose, in *ulp_source_file.S* we replace the incrementation code by the following code block that performs the startup and shutdown operations:

```
WRITE_RTC_REG(RTC_IO_TOUCH_PAD0_REG,RTC_IO_TOUCH_
   PAD0_HOLD_S,1,0)
WRITE_RTC_REG(RTC_GPIO_OUT_W1TS_REG,RTC_GPIO_OUT_
   DATA_W1TS_S+10,1,1)
WRITE_RTC_REG(RTC_GPIO_OUT_W1TC_REG,RTC_GPIO_OUT_
   DATA_W1TC_S+10,1,1)
WRITE_RTC_REG(RTC_GPIO_OUT_W1TS_REG,RTC_GPIO_OUT_
   DATA_W1TS_S+10,1,1)
```

After this program is downloaded to the ESP32, you can see on the oscilloscope how much time elapses between the sleep command and activation of the ULP routine (**Figure 8**). **Figure 9** shows this situation with an expanded time base.

Web Links

[1] www.elektor.com/esp32-devkitc
[2] www.elektormagazine.com/160454
[3] http://esp-idf.readthedocs.io/en/latest/get-started/index.html#get-started-connect
[4] www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers
[5] https://dl.espressif.com/dl/schematics/ESP32-Core-Board-V2_sch.pdf
[6] http://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
[7] http://espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
[8] https://github.com/krzychb/ulp-loop/blob/master/main/ulp/loop_blink.S
[9] https://github.com/espressif/binutils-esp32ulp/wiki#downloads
[10] www.elektor.tv



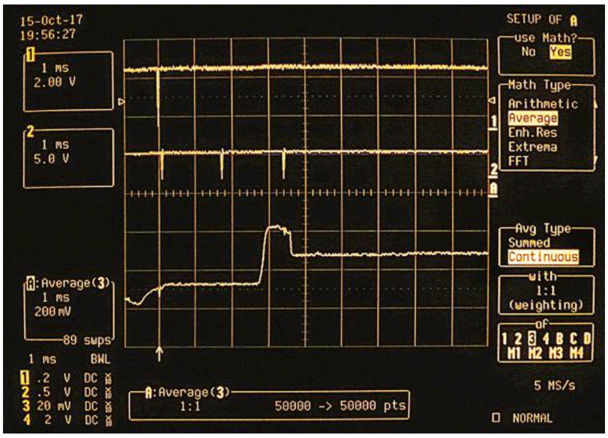


Figure 10. With this arrangement you can clearly see that the current consumption is lower when the Xtensa core is in sleep mode. Be careful: the grounds of the oscilloscope inputs are connected together.

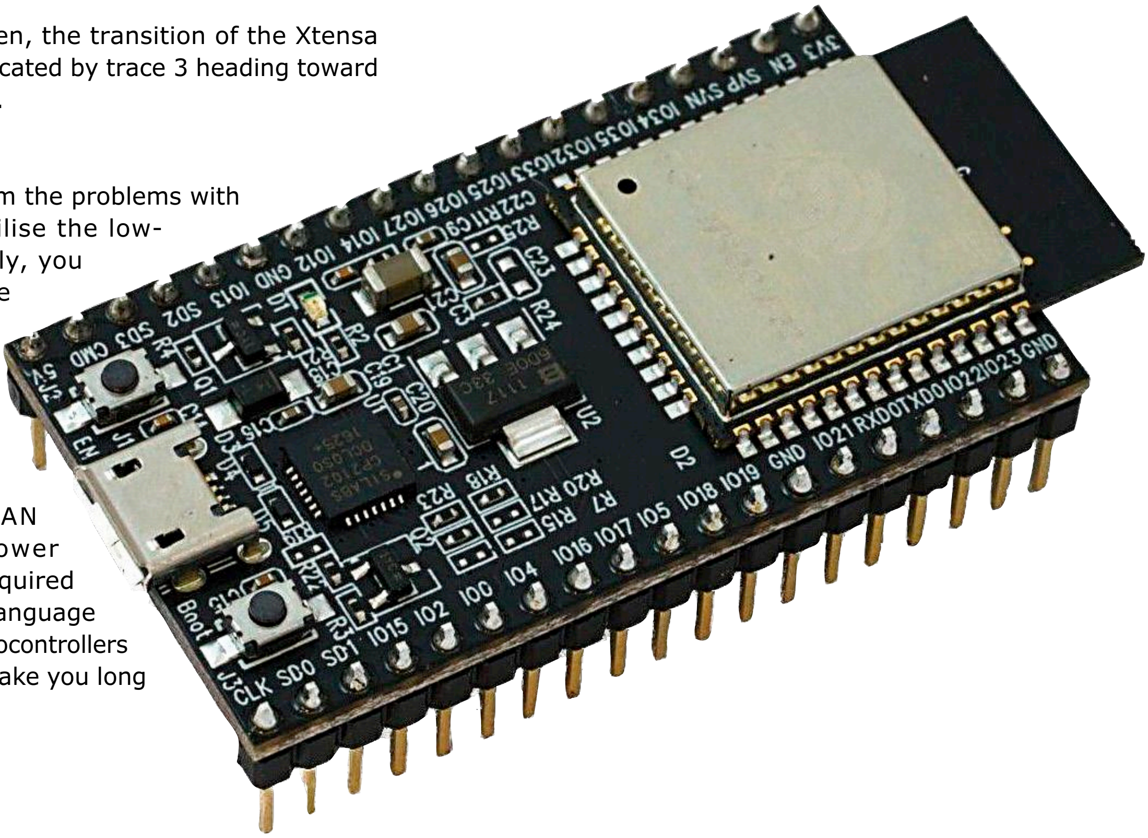
Figure 11. Shutdown of the Xtensa core can be recognised from trace 3 heading toward the baseline.

Here you can see that the ULP routine is activated three times in total, which explains the incrementing of the counter by 3 between the calls.

Now let's see whether all this actually reduces power consumption. For this, we connect an 8-ohm resistor in series with the GND lead and set up the measuring circuit shown in **Figure 10**.
On the oscilloscope screen, the transition of the Xtensa core to sleep mode is indicated by trace 3 heading toward the baseline (**Figure 11**).

Summary
Espressif has learned from the problems with the ESP8266. If you utilise the low-power processor properly, you can definitely reduce the power consumption of the ESP32. However, the very large scope of functions naturally means that it is no match for a true low-power microcontroller — WLAN is simply not a low-power interface. If you have acquired experience in assembly language programming for PIC microcontrollers and the like, it shouldn't take you long to feel at home with the ULP processor. For those (including the author) who think that assembly language programming on eight-bit processors is good preparation for working with high-level languages on MCUs, this is a sort of (belated) confirmation.

(160549-I)



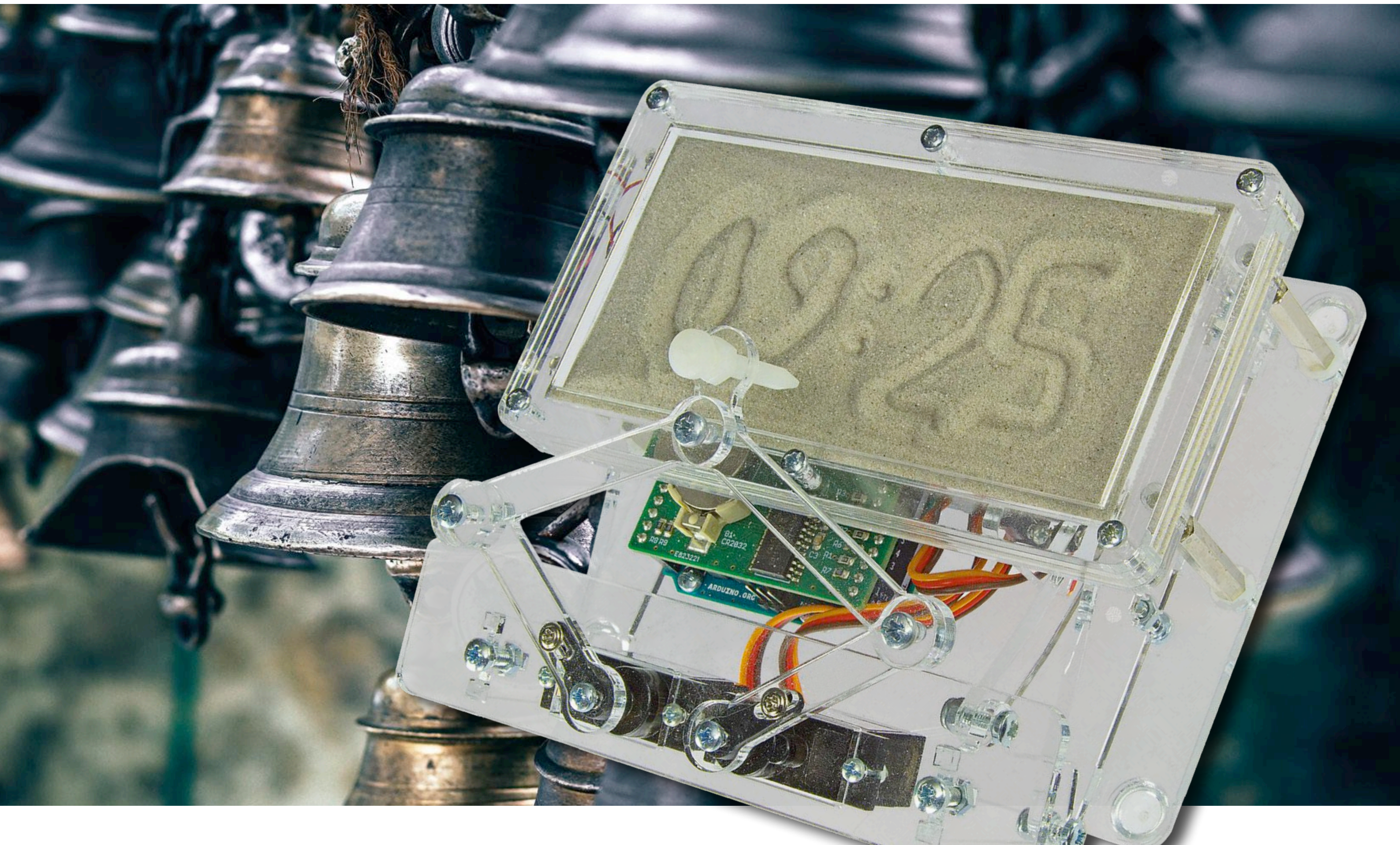
A video

If you find dealing with *make* a bit overwhelming, at [10] you can find a small video provided by the author that explains this in more detail.

HOMELAB PROJECT

Chimes for the Elektor Sand Clock

Shake, rattle and beep



By **Christopher R. Lee** (France)

I bought the Elektor Sand Clock as soon as it was announced. It made an unusually daft Christmas novelty, though the noise it produces renders it fairly useless in home or office environments. In this article we will make it even noisier by adding chimes.

Having assembled the Sand Clock [1], my first inclination was, as with any worthwhile gadget, to add some bells, whistles or — more obvious in this case — chimes! Because one should not be overambitious when launching a well-directed project, the present chimes are disappointingly… digital.

Selecting a sound system

An old-fashioned mechanical clock chimes every quarter-hour, the melody getting longer as the full hour is approached. Readers familiar with Big Ben (currently silent) will know the sequence of four-note motifs known as the *Westminster quarters* or *Westminster chimes*; I chose these but you can set up your own. The clockwork 'knows' when to start the chimes just before of each quarter hour, and first *boing* of the hour bell (or *cuckoo*) is supposed to be exactly on the hour.
I thought of programming the pantograph to strike a set of gongs (or play a xylophone or pluck some strings) at the

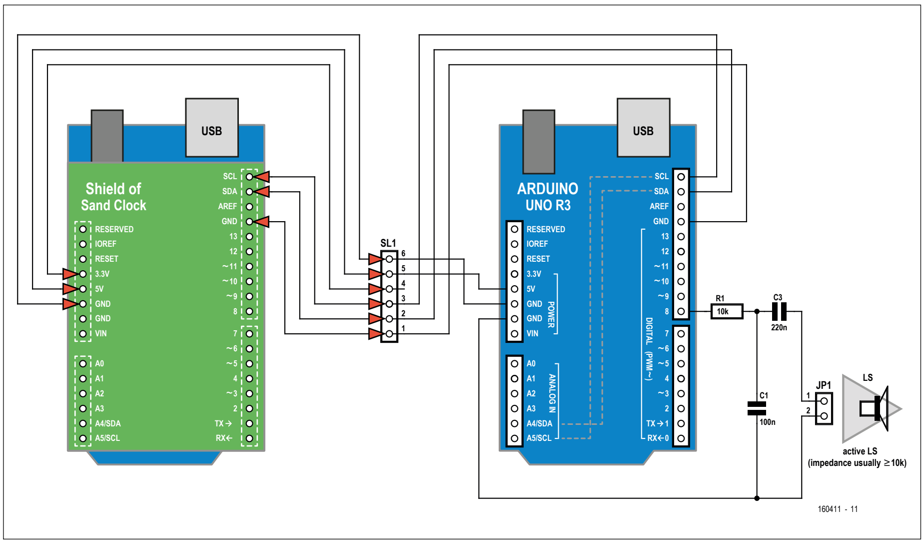


Figure 1. Here is how I connected a “slave” Arduino to the Sand Clock.

appropriate times. It could be done using the commands provided but I abandoned that idea for now, because you’d need some kind of escapement mechanism and sadly I don’t have the revered skills of ancient clockmakers, or harpsichord, piano makers for that matter.
While software will never replace genuine musical instruments, one feeble attempt is to make the Arduino family generate “acceptable” sounds using the Mozzi sound synthesis library [2]. For a preliminary tryout I decided instead to use Arduino’s built-in Tone function.

Interfacing to the Sand Clock

Following best practice I made the fewest possible adaptations to the Sand Clock’s hardware and software and sent the relevant data to a separate Arduino Uno. Any other Arduino should work with this simple application, and I’d suggest using the cheapest version for a permanent fixture.

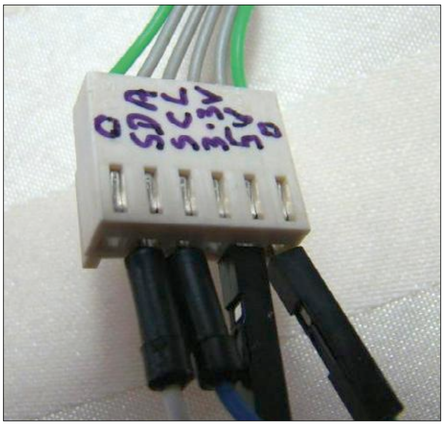


Figure 2. Clearly labelling connectors helps to avoid wiring mistakes.

However, with the arrangement shown (**Figure 1**) it **must** be a 5-V model. Transmission is through the I²C bus as it is already ‘declared’ for communications between the hardware clock on the shield and the Arduino. To be able to transmit the hours and minutes, I soldered a 6-way ribbon connector (**Figure 2**) with a female header directly to the SCL, SDA and power pins of the shield, which had already been assembled. If you haven’t started assembly there may be enough headroom for angled headers which provide a more adaptable plug-and-socket breakout; the usual Arduino stackable headers are too tall. The ribbon (and connectors I added to make the sand tray detachable) come out through a slot I cut in the back plate (**Figure 3**).

Good practice when interconnecting microcontrollers is always to use an isolator

I made direct connections between the I²C’s SCL and SDA pins of the two Arduinos, with the Ground return on the same ribbon cable. Pull-up resistors are already provided on the Sand Clock shield. Caution: this direct connection is possible only between Arduinos of the same operating voltage (5 V here); a 3.3-V model is likely to be damaged. In fact, the arrangement should be deprecated in any case, because the USB cables used for testing and control create a Ground loop with the 0-V power connections. Good practice when interconnecting microcontrollers is to use an isolator invariably. The components are cheap enough, but I didn’t find a through-hole model in the catalogues and I’m getting too old to fiddle with surface-mount. Rather than pay €20 for a ready-to-go PCB like the one described at Elektor Labs [3], I decided to risk giving it a miss. An alternative would have been to use the SoftwareSerial library (the interrupt-driven hardware serial port is in use) as I already make my own optically-isolated interfaces for MIDI applications. But Serial isn’t convenient and it may be ill advised in any case to add code involving precise timing to the Sand Clock program.
The audio output from pin 8 is connected to an active loudspeaker. A low-pass filter makes the sound less, erm, ugly. **Figure 4** shows my prototype.

The sketches

The software for this project can be downloaded from [4]. The sketch `sandclockBigBen.ino` is a version of the original Sand Clock software with a few lines of added code that simply send the four bytes of the hours and minutes digits via the Wire library to the Arduino that executes the chimes with `BigBenI2C.ino`.
For development and testing, I used sketch `wire_master_clockV2.ino` to send four bytes via I²C from another Arduino whenever a character is received from the terminal. There’s also a test receiver sketch `slave_receiver_clockV2.ino`. I ran the two instances of Arduino on the same computer, without an I²C isolator (see above).
The I²C master is the Sand Clock’s Arduino, so it doesn’t need an address. The chime-Arduino is set a Slave at address 0x08, the first available one. This address doesn’t matter unless you connect any extra modules (weather station…) with fixed addresses. The real-time clock (RTC) built into the Sand Clock shield has the fixed address 0x51. Incidentally, though this module is temperature-compensated and highly accurate, you could place a clock with an external reference on the same bus, most likely via an additional inexpensive Arduino.

The 15-minute interval mod

As supplied, the Sand Clock is programmed to write the time in the sand every minute. In response to complaints about (acoustic) noise, sketch `sandclockBigBen15min.ino` vibrates and writes in the sand every 15 minutes (or whatever you want to program). The sand bed is shaken just before the final minute arri-

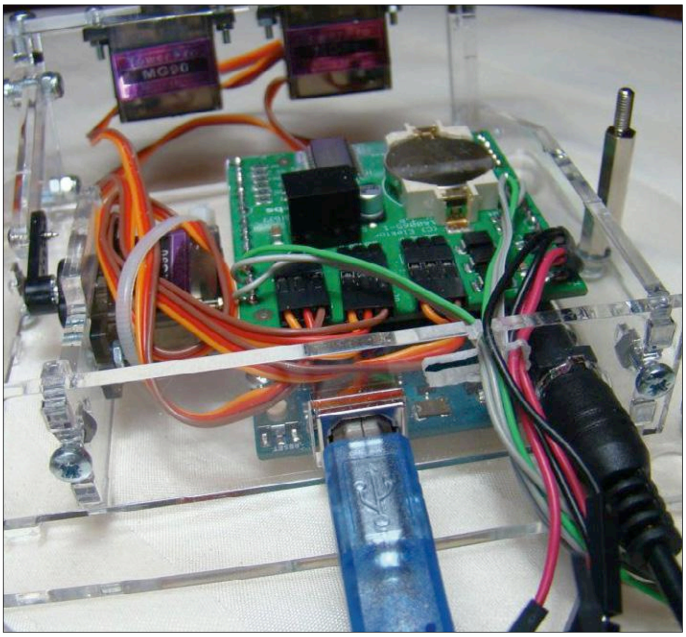


Figure 3. Even though I am not the best acrylic sheet (Plexiglas™) worker in the world, I did manage to pass the wires through without breaking the lot.

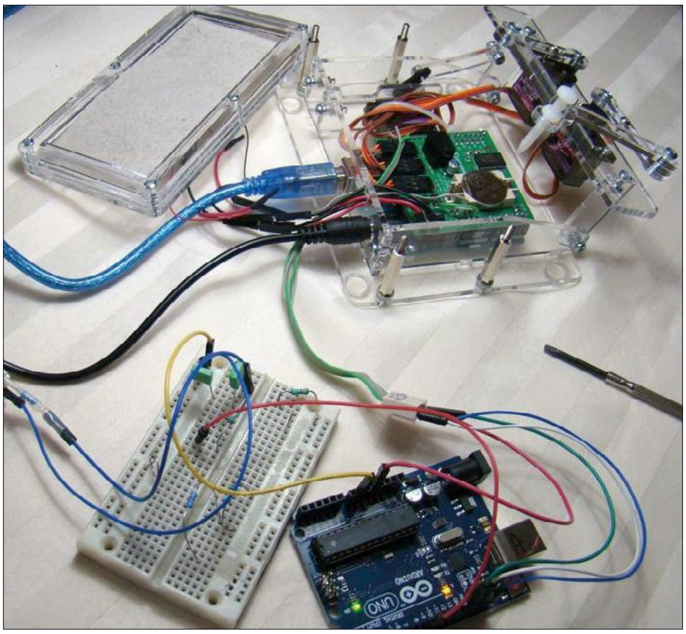


Figure 4. Exploded view of my prototype.

ves. (I kept the default of five seconds but should probably have made that a variable read from the Sand Clock.) We assume that the chimes will be short enough to start during each 14th, 29th, 44th and 59th minute and finish in time for the shaking of the sand bed.

Create your own melodies

The Arduino Tone function generates a square wave at the frequency of the specified note and for the specified duration. The frequency of each note is read from library file `pitches.h`, which you can modify to obtain harmonically pure intervals in lieu of the default equal-tempered ones, or perhaps generate comically wrong notes. My software is musician-friendly, the note durations being calculated from note values (4 = quarter note or crotchet, etc.). Since chime melodies are short, I didn't think it worth providing an interface for you to input your own. If you want to do this anyway, you will have to modify the values in the corresponding arrays `melodyx[]` and `noteDurationsx[]`, and then recompile. There's space for longer melodies and that can be arranged by modifying arrays `noteOnx[]` and `noteLenx[]`. The durations and timings are calculated automatically (I found that difficult and the code may be unduly clumsy). Each Tone command needs to be followed by a delay at least equal to the calculated duration. I added an extra 2 ms to make sure.

More Sand Clock enhancements

According to the designers of the Sand Clock, the most difficult mechanical problem to be cracked was smoothing out the sand. Positioning the vibrator motors is an empirical process, and with my kit the sand occasionally migrates to one end of the tray. The reasons for that, as well as for a tendency to segregate ungraded sand by particle size, have been elucidated by Nobel Prize winner Pierre-Gilles de Gennes. For a mission-critical piece of apparatus, we need a more predictable solution. I propose that the pantograph be fitted with an additional tool that neatly rakes the sand, as in an Olympic long-jump sand pit. Alternatively, someone has surely already designed an independent little robot that could drive up to the tray when called upon. If you did, please let us know.

Finally, the Sand Clock's RTC, the PCF2129A, is a pretty accurate component that has several features (like Alarm and Timestamp) that have not been exploited yet. It might be interesting to make these options available to external hardware too.

(160411)

Web Links

[1] Sand Clock article: www.elektormagazine.com/160065

[2] Mozzi sound synthesis library: sensorium.github.io/Mozzi/

[3] I2C Isolator: www.elektormagazine.com/labs/universal-i2c-bus-isolator-and-level-adapter-150089

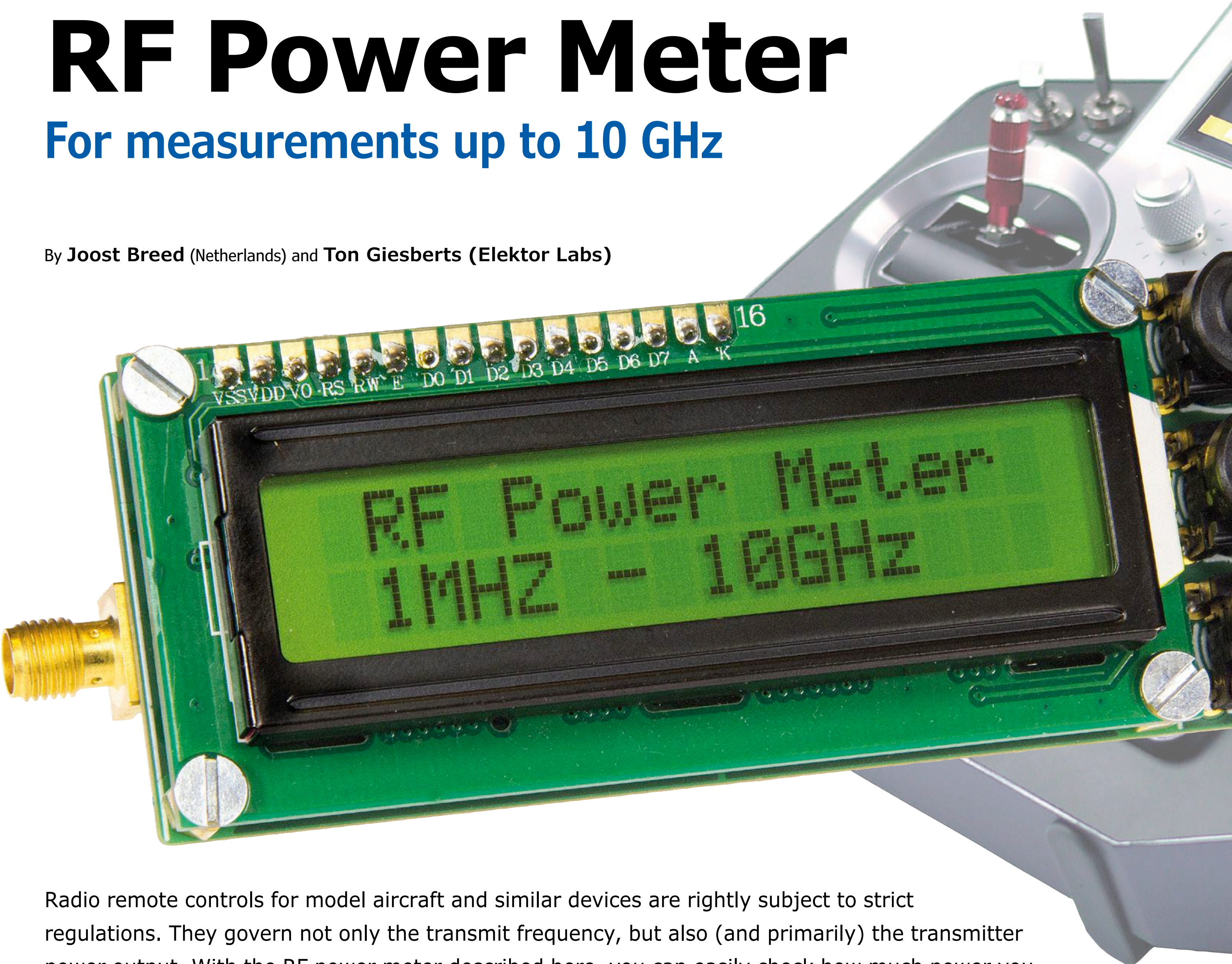
[4] This article: www.elektormagazine.com/160411

eLABS PROJECT

RF Power Meter

For measurements up to 10 GHz

By **Joost Breed** (Netherlands) and **Ton Giesberts (Elektor Labs)**



Radio remote controls for model aircraft and similar devices are rightly subject to strict regulations. They govern not only the transmit frequency, but also (and primarily) the transmitter power output. With the RF power meter described here, you can easily check how much power you are transmitting, at frequencies up to 10 GHz.

Features

- RF power meter
- Bandwidth 1 MHz to 10 GHz
- Dynamic range 55 dB
- Compact: approx. 95 × 36 × 30 mm

The author is an enthusiastic member of the FMS Spaarnwoude model aircraft club and the proud owner of an octocopter with a wingspread of over 3 feet (1 metre), equipped with a camera and live video link. The maximum permissible transmit power of the radio remote control for this model, which operates in the 5.8-GHz band, is 25 mW. That is more than enough because the models are not allowed to fly higher than 300 feet (100 m) and must remain within sight at all times. The home base of FMS Spaarnwoude is near Schiphol Airport, the largest airport in the Netherlands, so it is especially important to avoid exceeding the maximum transmit power — if only to avoid hassles with the authorities.

Of course, there's nothing to stop you from buying a ready-made RF power meter, but they can easily set you back several hundred euros. That makes DIY construction an attractive alternative.

The approach

To avoid any misunderstanding, we should point out that this is not a simple DIY project. To keep the design of the RF power meter within the range of advanced solder artists, we opted for an approach with two (actually, three) small circuit boards: a main

board for the control elements and an LC display, plus a front end in the form of a breakout board (BoB) mounted on the main board. And what about the third board? That is the microcontroller in the form of an Arduino Nano module, which is also mounted on the main board. We choose the Nano because it is small and inexpensive, fast enough, and has enough processing power for this application. This approach also allows the front end module to be used for other applications and/or be located closer to the signal source.

The main board

The remarkably simple schematic diagram of the main board is shown in **Figure 1**. It is built around the Arduino Nano (MOD2). A standard LCD module with two lines of sixteen characters is connected to pins D2 through D8, and the brightness can be adjusted with P1. We added resistor R5 to enable more precise adjustment. Pushbuttons S1–S3 (with pull-down resistors R1–R3) are the control elements of the operator interface, as described further on. Power is provided by an external 7–12 V DC power supply connected to K1. From this a clean 5 V supply

PROJECT INFORMATION

RF
Arduino Nano
SMD

entry level
intermediate level
➡ expert level

4 hours approx.

SMD soldering equipment
Hot-air soldering station or reflow oven
Programming adapter

€50 / £45 / $60 approx.

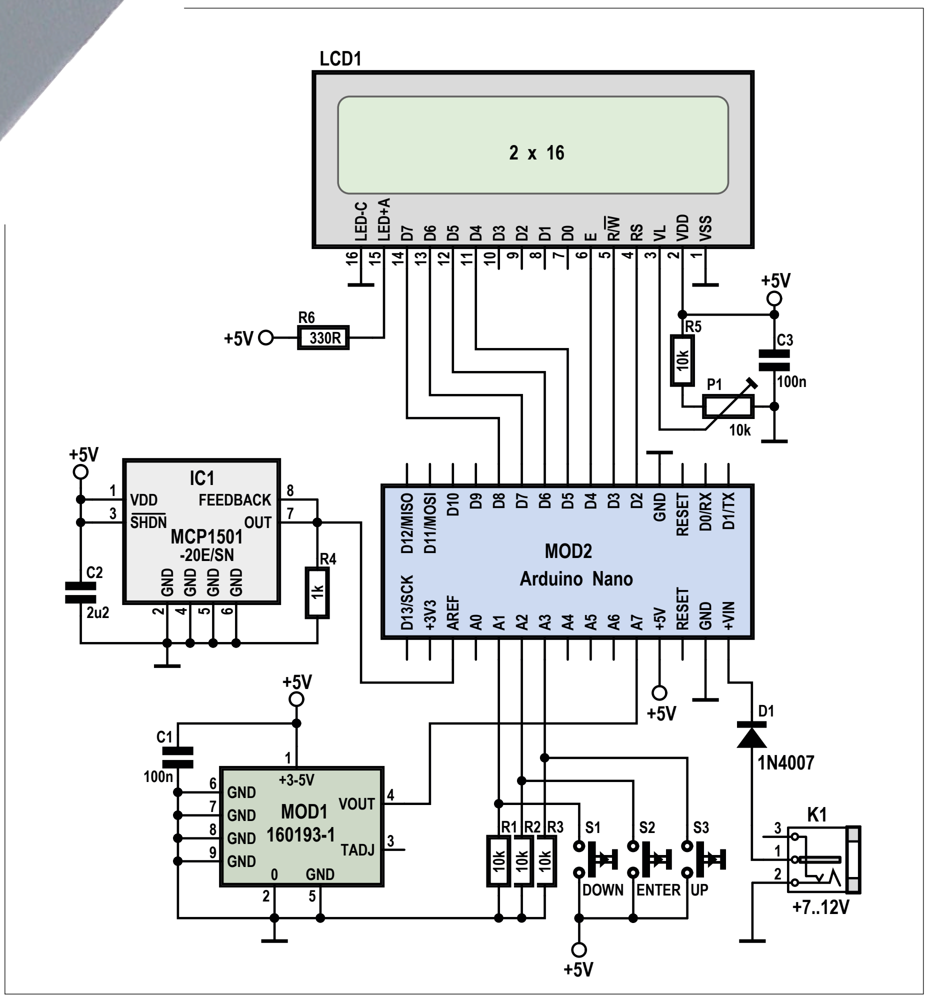


Figure 1. The schematic diagram of the main board, basically consisting of three modules.

Testing 1, 2

We carried out several measurements on a prototype of the breakout board without the main board. For that we used the tracking generator of a vintage Tektronix 2710 spectrum analyser. It has a maximum frequency of 1.8 GHz. The amplitude ranged from −48 dBm to 0 dBm. We measured the output voltage of the module at 100 MHz, 1 GHz and 1.8 GHz.
For the first series of measurements (**Table 1**), we connected the RF module as follows: N/BNC adapter on the analyser, 50 cm RG058 cable with BNC connectors, BNC/SMA adapter on the module.

Table 1

	100 MHz (21.3 mV/dB)		1 GHz (22.1 mV/dB)		1.8 GHz (22 mV/dB)	
V_{in} [dBm]	V_{out} [V]	mV/dB	V_{out} [V]	mV/dB	V_{out} [V]	mV/dB
0	0.423		0.417		0.391	
−10	0.610	18.7	0.624	20.7	0.601	21
−20	0.834	22.4	0.832	20.8	0.786	18.5
−30	1.052	21.8	1.067	23.5	1.044	25.8
−40	1.274	22.2	1.292	22.5	1.244	20.0
−48	1.447	21.6	1.480	23.5	1.445	25.1

The second series of measurements (**Table 2**) was made without a cable: N/BNC adapter on the analyser, BNC/BNC adapter (female/female), BNC/SMA adapter on the module.

Table 2

	100 MHz (21.3 mV/dB)		1 GHz (22.1 mV/dB)		1.8 GHz (22 mV/dB)	
V_{in} [dBm]	V_{out} [V]	mV/dB	V_{out} [V]	mV/dB	V_{ou}t [V]	mV/dB
0	0.421		0.409		0.380	
−10	0.608	18.7	0.614	20.5	0.583	20.3
−20	0.832	22.4	0.821	20.7	0.770	18.7
−30	1.050	21.8	1.057	23.6	1.027	25.7
−40	1.272	22.2	1.283	22.6	1.233	20.6
−48	1.445	21.6	1.471	23.5	1.434	25.1

We also tried two other coax cables. The first (35 cm RGU400, BNC to SMA) proved to contribute an additional attenuation of about 0.2 dB at 1.8 GHz, while the second (1 m RG223, BNC to SMA) yielded an additional attenuation of slightly less than 0.6 dB.

Table 3 provides some more measurement results:

Table 3

V_{out} max.	1.718 V (without input signal)
V_{out} min.	0.359 V (measured at +10 dBm and 15 MHz)
Max. input power	+12 dBm (according to data sheet)
V_{in} min. (K1)	6.6 V (at lower voltages the output of the 5 V regulator collapses)
Current consumption	85 mA

voltage for powering the various modules is generated on the Arduino Nano board. Diode D1 provides reverse-polarity protection.
The internal reference voltage of the Nano is not sufficiently accurate for our purposes, so we added a separate 2.048-V reference voltage source in the form of IC1, an MCP1501-20E/SN [1]. This Microchip IC provides an excellent precision of 0.1%. For stable operation it needs a resistive load, provided here by resistor R4. The AREF input of the Nano board is decoupled by a 100-nF capacitor, which is already present on the Arduino Nano.
Finally, the schematic shows the RF front end in the form of MOD1. It is connected to the main board by just three leads: power, ground, and the analog output voltage proportional to the RF power level.

The BoB

The design of the RF front end turned out to be easier than we originally expected or feared. That's because there are ICs available — called log detectors — that convert the measured power into an analogue output voltage. A major part of the design process therefore consisted of finding a suitable log detector IC. In the end we opted for the Analog Devices AD8317 [2]. It has a more than adequate bandwidth of 1 MHz to 10 GHz, along with a dynamic range of 55 dB. The schematic diagram (see **Figure 2**) follows the recommendations in the data sheet.
The signal enters on connector K1. The maximum input signal level is approximately 0 dBm, so an attenuator

must be used for power levels above 1 mW. Resistor R1 in combination with the input impedance of the AD8317 provides a wideband input impedance of 50 Ω. Capacitors C1 and C2 block any DC component and at the same time form a high-pass filter in combination with the input impedance, with a corner frequency of about 68 kHz.
Resistor R2 provides temperature compensation. The value shown on the schematic is recommended for measurements at 5.8 GHz. See the data sheet for other values. TADJ is also fed out to connector K2, but that feature is not used in the present application. Capacitor C3 is for the low-pass demodulation filter of the output signal V_{out}. Since we are not interested in demodulation in this application, we chose the standard value of 8.2 pF here.
The output signal V_{out} is fed to the VSET input via the voltage divider R4/R5. For R5 the value shown on the schematic is 0 Ω, while R4 is not mounted (NM). This voltage divider can be used if necessary to make the slope of the output voltage characteristic steeper than the minimum value of 22 mV/dB. Here again, for more details you should consult the data sheet of the IC.

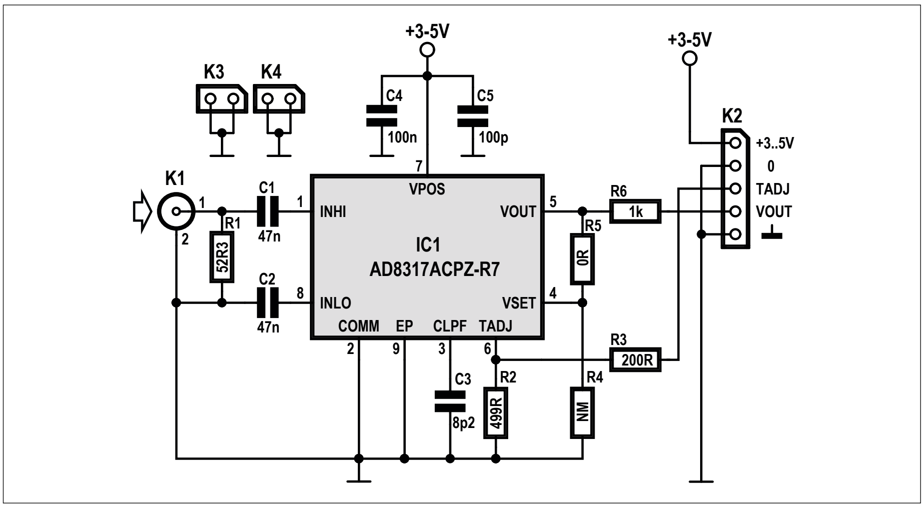
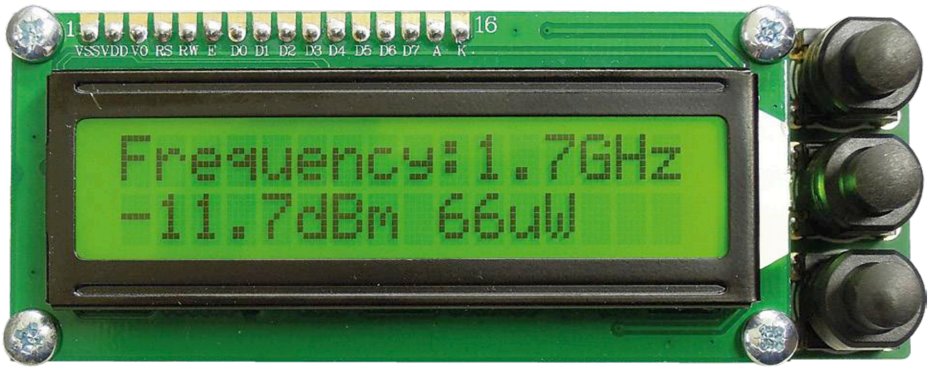


Figure 2. The RF module consists of a single IC and a handful of small components.

Wide bandwidth and large dynamic range



Construction

Figure 3 gives an impression of the fully assembled RF power meter. But we aren't that far yet — there's still some finicky work ahead of us (or actually, you).
Let's start with the most difficult part: the RF front end. For this we designed a small (20 × 27 mm) PCB layout, shown in **Figure 4**. Note that this is a four-layer board, which is absolutely essential for a circuit intended to operate up to 10 GHz. One of the hidden layers is dedicated to the positive supply voltage. The parasitic capacitance between that layer and the ground planes provides additional decoupling of the supply voltage. Another word of warning: don't try to make this or something similar yourself, as otherwise the circuit is guaranteed to not work properly.

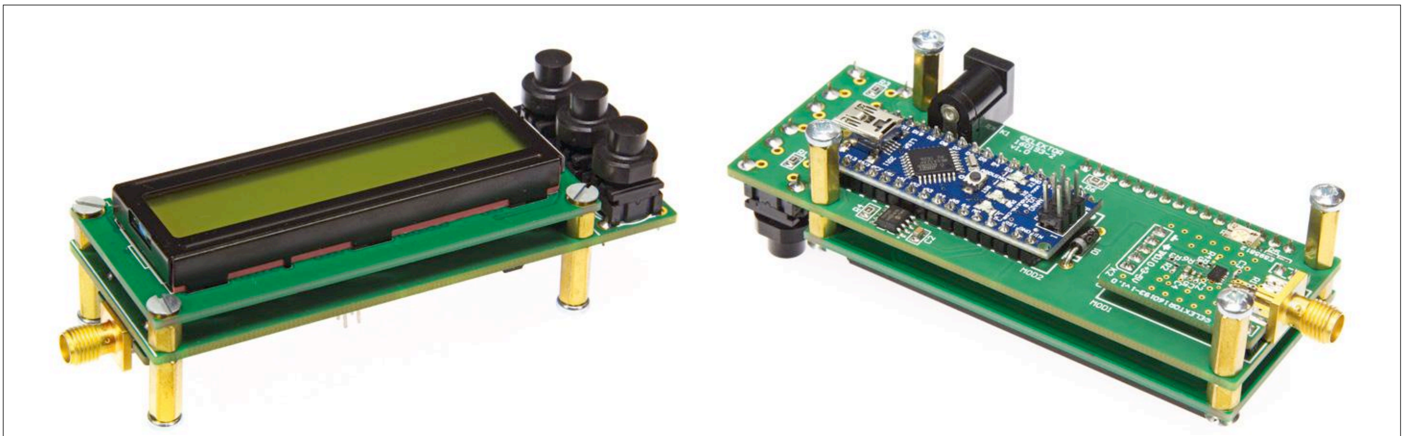


Figure 3. The fully assembled module is nice and compact.

COMPONENT LIST

PCB # 160193-1 (RF module)

Resistors
Default: SMD 0402
R1 = 52.3Ω, 1%, 100mW
(ERJ2RKF52R3X, Panasonic)
R2 = 499Ω, 1%, 62.5mW
R3 = 200Ω, 1%, 62.5mW
R4 = not mounted
R5 = 0Ω, 1%, 62.5mW
R6 = 1kΩ, 1%, 62.5mW

Capacitors
Default: SMD 0402
C1,C2 = 47nF, 10%, 25V, X7R
C3 = 8.2pF, ±0.5pF, 50V, C0G/NP0
C4 = 100nF, 10%, 16V, X7R, SMD 0603
C5 = 100pF, 10%, 16V, C0G/NP0

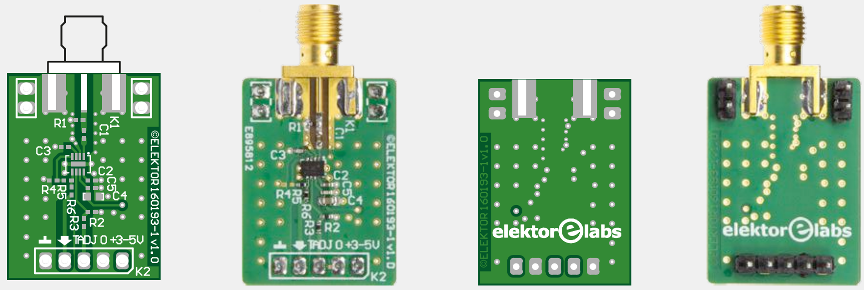


Figure 4. The four-layer PCB for the RF module.

Semiconductors
IC1 = AD8317ACPZ-R7, SMD LFCSP_VD
(CP-8-1)

Miscellaneous
K1 = SMA, 50Ω, straight, edge mount
(142-0701-801, Johnson/Cinch)
K2 = 5-pin (1×5) pinheader, vertical, 0.1" pitch, through-hole
K3,K4 = 2-pin (1×2) pinheader, vertical, 0.1" pitch, through-hole
PCB 160193-1 v1.0

Due to the high frequencies, all resistors and capacitors are in 0402 packages except C4, which is in an 0603 package.
Now we come to the only drawback of the AD8317 used here: the CP-8-1 SMD package. The eight contacts are barely visible, and on the bottom of the package there is an exposed pad that must be bonded to GND. That is simply not possible with an ordinary soldering iron, no matter how fine the tip may be. The only way to solder this IC is to use a hot-air soldering station or a reflow oven. And don't use too much solder paste!
After the BoB, the main board is a piece of cake. Only ordinary SMD components are mounted on it.

FROM THE STORE

➞160193-1
PCB RF module

➞160193-2
PCB main board

➞17002
Arduino Nano

➞18241
Franzis Nano Board

➞120061-77
LCD 2×16 white on blue

➞120061-74
LCD 2×16 standard

Figure 5 shows the PCB layout we designed for this purpose, which is about the same size as the display module. Mount the normal components first, but wait a bit with the two modules and the display.

Assembly

On our first prototype we fitted bus strips for the two modules and the display. However, that manner of assembly is not recommended, even though the circuit works properly, because the overall unit is nearly 4.5 cm high as measured from the display to the ICSP header of the Nano board.
If the modules and the display are instead soldered directly on the main board, the overall assembly is only 28 mm high. Proceed as follows:
First mount the two modules (see Figure 3 again), plug the header for the display into the right holes, and then secure the display in place with four 5-mm standoffs (male/female) and matching screws. After this you can solder the pinheader for the display module.

Firmware and use

The software of the RF power meter is fairly straightforward. In the main loop of the program, 500 samples are taken and then the average power, the peak power, the minimum power during the last ten seconds, and the maximum power during the last ten seconds are determined and the modulation index is calculated. Then the results are sent to the display, and the next measurement cycle starts.
Before the correct values can be read, the attenuation and the frequency must be selected. For this we implemented a simple menu structure, which uses the Up and Down pushbuttons as well as the Enter button for confirmation of the selected entry. The frequency options are 900 MHz, 1.8 GHz, 2.2 GHz, 3.6 GHz, 5.8 GHz, and 8 GHz. The characteristic curve for conversion of the input voltage of the A/D converter and calculation of the actual power in dBm is a straight line with a slope of −22 dB/V, but the origin of the line is different for each frequency (see the data sheet).
The software issues a warning when the input power is too high or too low. According to the data sheet, a low input power level leads to a large measurement error.

Attenuators

The author uses two attenuators from Mini-Circuits: the VAT-20W2+ (20 dB) [3] and the VAT-30W2+ (30 dB) [4]. They are supplied complete with datasheets. The characteristics of these two attenuators are incorporated in the firmware. By the way, the firmware also has a built-in calibration function, so it is possible to use attenuators other that the two previously mentioned types preprogrammed in the firmware.

Final remarks

We have only briefly touched on the use of the RF power meter. However, a very detailed and richly illustrated User Guide (seven pages in A4 format, PDF) is included the free download for this article [5]. Of course, the firmware is also included in the download, as well as all datasheets, likewise in PDF format.

(160193-I)

Web Links

[1] ww1.microchip.com/downloads/en/DeviceDoc/20005474D.pdf

[2] www.analog.com/media/en/technical-documentation/data-sheets/AD8317.pdf

[3] www.minicircuits.com/pdfs/VAT-20W2+.pdf

[4] www.minicircuits.com/pdfs/VAT-30W2+.pdf

[5] www.elektormagazine.com/160193

COMPONENT LIST

PCB # 160193-2 (main board)

Resistors
Default: SMD 0805
R1,R2,R3,R5 = 10kΩ, 5%, 0.1W
R4 = 1kΩ, 5%, 0.1W
R6 = 330Ω, 5%, 0.1W
P1 = 10kΩ, 20%, 0.25W, SMD (Bourns 3314G-2-103E)

Capacitors
Default: SMD 0805
C1,C3 = 100nF, 50V, 10%, X7R
C2 = 2.2µF, 25V, 10%, X7R

Semiconductors
D1 = 1N4007, 1000V, 1A, THM
IC1 = MCP1501-20E-/SN, SMD SOIC-8

Miscellaneous
K1 = DC power connector, 3A, 1.95mm, Lumberg NEB 21 R
S1,S2,S3 = pushbutton, PCB SPST NO, Multimec RA3FTH9
S1,S2,S3 = round cap for pushbutton, black, height 16mm, Multimec 1S09-16.0
LCD1 = LCD module 2×16, 80×36mm, 3mm mounting holes
LCD1 = 1×16 header, vertical, 0.1" pitch, through-hole
MOD1 = 160193-1, RF-module
MOD2 = Arduino Nano
PCB 160193-2 v1.0

Optional (not recommended)
For LCD: 1×16 bus strip, vertical, 0.1" pitch
4 standoffs, M3, 12mm, F/F
4 screws, M3 (length > 6mm)
For MOD1: 2 1×2 and 1 1×5 bus strips, vertical, 0.1" pitch
For MOD2: 2 1×15 bus strips, vertical, 0.1" pitch
Depending on how the RF power meter is fitted in an enclosure, additional M3 standoffs (M/F), M3 screws and nuts may be necessary.

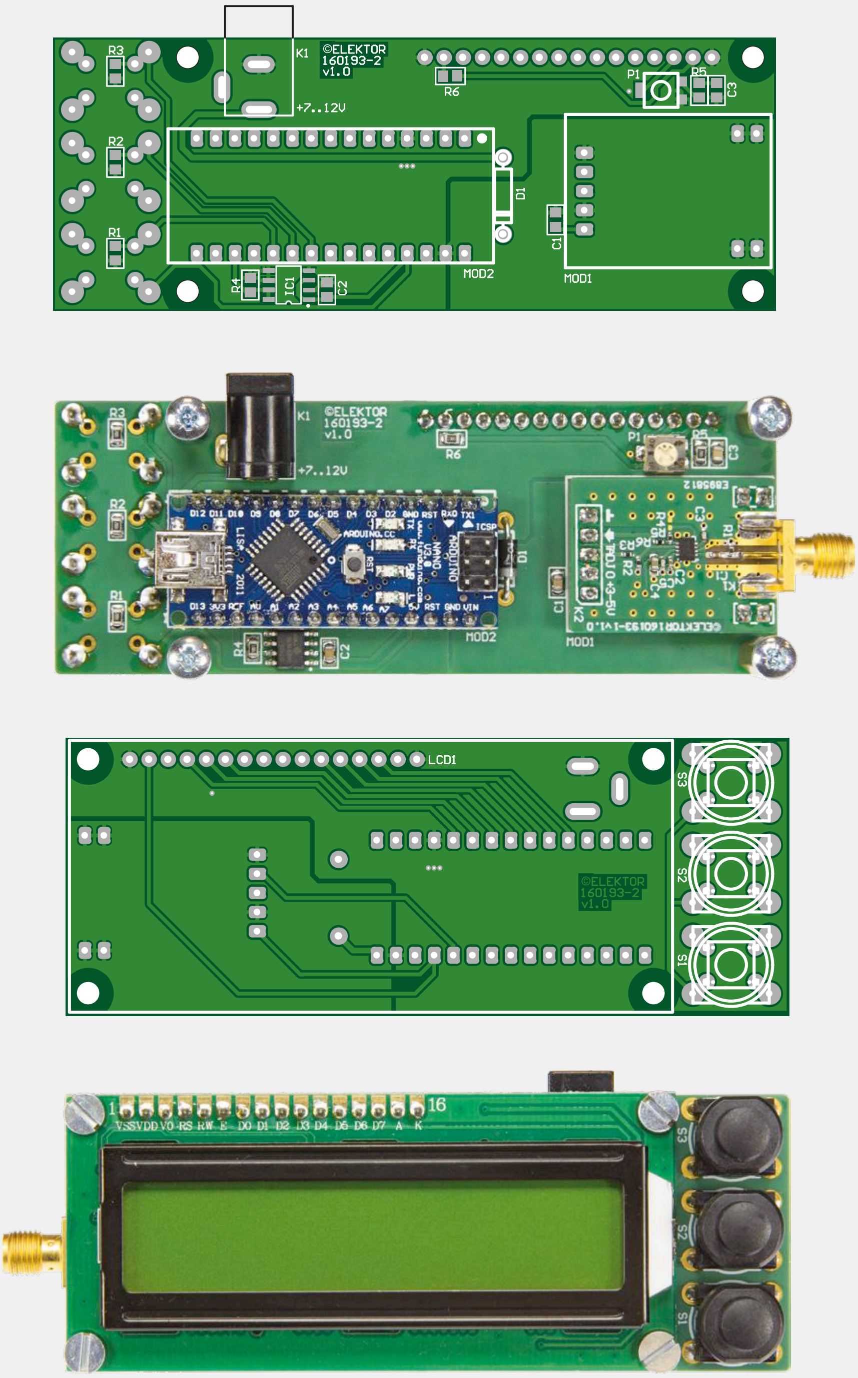


Figure 5. The main board of the RF power meter.

(almost) everything you wanted to know about...
Nixie Tubes

By **Ilse Joostens**
(Belgium)

With their softly glowing digits staggered from front to rear, Nixie tubes are totally back in fashion now, especially when a retro or steampunk look is desired.

Q *Where does the name 'Nixie' come from?*

A The answer has historical roots [1]. As a result of industrialisation and the emergence of electronics in the first half of the twentieth century, especially shortly after the Second World War, there was more and more demand for electronic displays. The first patent for a display resembling the Nixie tube was applied for on 9 May, 1934, by Hans G. Boswau from Galion (Ohio). However, it took until around 1954 before the GI-10 was commercially launched by National Union under the name Inditron [2]. At roughly the same time, the brothers George and Zoltan Haydu, originally from Hungary, were also working on the development of various types of tubes. When their company was acquired by Burroughs in 1954, a draftsman with nothing else in mind put the title 'NIX-I' on the drawings of the display tube. The abbreviation stood for '*Numerical Indicator eXperimental no. 1*', and it quickly became 'Nixie' in common speech. Production started in 1956, and as the saying goes, the rest is history.

Q *Are Nixie tubes still being made?*

A Nixie tubes are no longer made commercially, which means in large quantities. As far as we know, they remained in production in the former Soviet Union (Russia and the Ukraine) until early 1992. However, they are now being produced in small volume by the Czech firm Dalibor Farny [3]. Other parties are also experimenting with production. However, you should expect very stiff prices if you are considering the use of Nixie tubes.

Q *Where are Nixie tubes still available?*

A The largest number of Nixie tubes (primarily Russian) can be found on eBay, from sellers in the former East Bloc. European and US webshops and eBay sellers also offer Nixie tubes (primarily Russian), but they are often considerably more expensive. However, stocks are gradually becoming exhausted, so the supply is decreasing and prices are gradually rising. Many non-Russian Nixie tubes are increasingly hard to find and are becoming true collector's items.

Q *How do Nixie tubes operate?*

A Nixie tubes are usually powered from a DC voltage of about 170 V, with the current through the tube limited by an anode resistor. The Russian K155ID1 driver IC, which is still being produced by Integral, is often used to drive the cathodes (**Figure 1**). That IC is derived from the legacy 74141 BCD/decimal decoder and Nixie driver.
Of course, discrete transistors can also be used, such as the well-known MPSA42, or a high-voltage shift register / driver IC such as the HV5812 from Microchip.
When Nixie tubes are multiplexed (see the next question), discrete transistor pairs consisting of an MPA42 and an MPSA92 are often used as anode drivers. Another option is to use high-voltage optocouplers, such as the SFH619A.
In the 'with due care & consideration' category, transistors such as the BC547 or the BC550 may be used as cathode drivers for relatively small Nixie tubes in static (non-multiplexed) operation. As long as the sum of the ignition voltage of the Nixie tube and the real reverse breakdown voltage of the transistor is much less than the anode voltage, the 547 and 550 can function as Nixie drivers. If transistor breakdown does occur, that usually does not have dramatic consequences because the current is only a few milliamps, so the power dissipated by the transistor remains within limits [4].
Even stranger options are possible — if the anode voltage is carefully regulated, the cathodes of Nixie tubes can be driven directly by standard CMOS logic ICs (HEF4xxx or CD4xxx) if they are powered from 15 V. That is possible due to the small difference between the ignition voltage and the discharge sustain voltage of many (but not all) types of Nixie tubes [5]. Nice to know, but not really recommended.

Q *Is it possible to multiplex Nixie tubes?*

A Nixie tubes can easily be multiplexed, and their brightness can be regulated using pulse width modulation. That is possible thanks to the operating principle (based on gas discharge) and the short ignition time. In some cases, mechanical resonance of the cathodes or the anode grid can occur, resulting in a whistling sound from the Nixie tubes. That occurs most often with relatively large Nixie tubes. This problem can be solved by adjusting the multiplex frequency and/or timing.

Q *What is the difference between a Nixie tube and a VFD or Numitron?*

A A Nixie tube is a gas discharge tube with a cold cathode. A VFD (vacuum fluorescent display) tube, by contrast, has a heated cathode. At the cathode, the potential difference causes thermal electrons to be emitted and accelerated toward an anode coated with a phosphor. When they strike the phosphor, it emits light. VFD tubes work at lower voltages – about 20 to 60 V for the anodes and control grids, and 1 to 3 V for the cathode heaters. With VFD tubes the anodes and grids are controlled by the electronics, instead of the cathodes as with Nixie tubes.

Numitrons consist of several incandescent filaments enclosed in a glass envelope and arranged in the form of a seven-segment display. They operate at even lower voltages – around 5 V for the smaller types – and they are controlled in the same way as incandescent lamps. Relatively small Numitrons can be driven

Figuur 1. Hier zijn de 'gestapelde' kathodes van een Nixie-buis duidelijk te zien.

Figuur 2. Van links naar rechts: een Nixie-buis, een VFD en een Numitron.

directly by logic ICs or microcontrollers. The various types of display tubes are shown side by side in **Figure 2**.

Q *How can you test a Nixie tube to see if it is still in good condition?*

A For a quick test, you can hold the tube next to a 'plasma ball'. If the neon gas inside the tube lights up, you at least know that air has not leaked into the tube. For a better test, you need a Nixie power supply with an output voltage of 160 to 190 V. These are available at reasonably low prices, either as kits or ready-made. You can also build your own on a piece of perfboard [6].
Connect the positive lead of the power supply to the anode though a resistor with a value of 33 kΩ to 47 kΩ; that can be a bit less with relatively large tubes. Now carefully touch the ground lead of the power supply to the cathode pins one by one; the associated numeral should light up each time. If that does not work, you can try reducing the anode resistance a bit. If that does not help, the tube is probably defective.

Q *Are Nixie tubes without a type number or marking still usable?*

A You can test the tube as described in the previous answer. First you have to identify the anode pin. That can usually be done visually: it is the pin that is connected to the metal grid inside the tube. Then you can test the other pins. Nixie tubes are fairly robust, so there's not much that can go wrong. If you see a strange effect, you probably have the positive lead connected to one of the cathodes instead of the anode.

Q *How long do Nixie tubes last?*

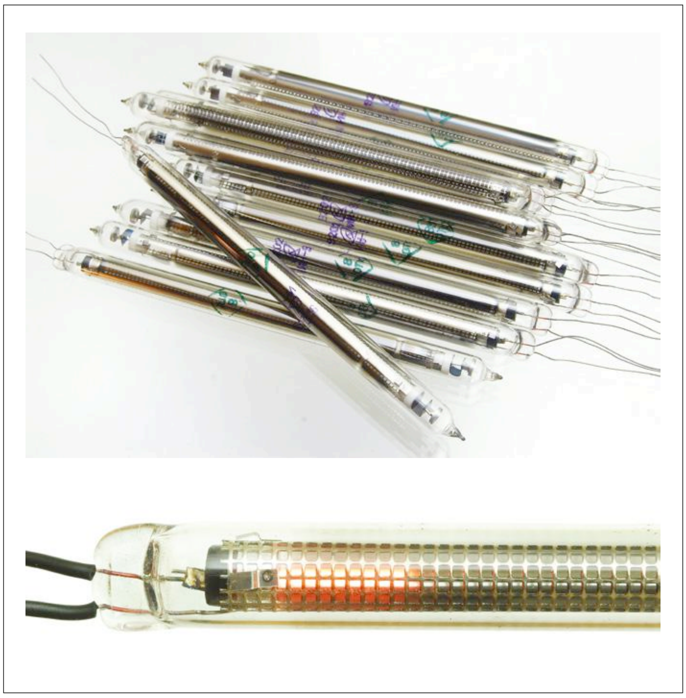
A Nixie tubes can last for a very long time. Forty years of continuous operation is by no means exceptional. However, they can also suddenly stop working. The most common causes are a crack in the glass envelope, air leakage into the tube, cathode poisoning, an internal short circuit, or an internal open circuit due to mechanical effects [7].

Q *Are there other types of Nixie tubes beside numerical?*

A Along with the well-known tubes with only digits, there are also Nixie tubes with letters and/or symbols, and there are 7-segment, 13-segment and 15-segment Nixie tubes. In addition to the traditional round tubes, Nixie tubes were available in the form of flat Panaplex displays.
The Russian IN-28 tube displays a single bright orange dot; it was used as a pixel in dot-matrix displays.

There are also bargraph Nixie tubes, such as the IN-9 (**Figure 3**). They have a long, cylindrical nickel-plated anode grid and a molybdenum cathode wire. This wire displays a light column whose length depends on the amount of current. The combination of molybdenum and neon results in a low ignition voltage. The lower end of the cathode is coated with zirconium, which has an even lower ignition voltage. This ensures that the light column starts at that end. An improved version (IN-13) has an auxiliary electrode to initiate the light column at the lower end.

(160618-I)



Figuur 3. Bargraph-Nixies.

Links

[1] www.decodesystems.com/nixie-history.html

[2] www.decadecounter.com/vta/articleview.php?item=423

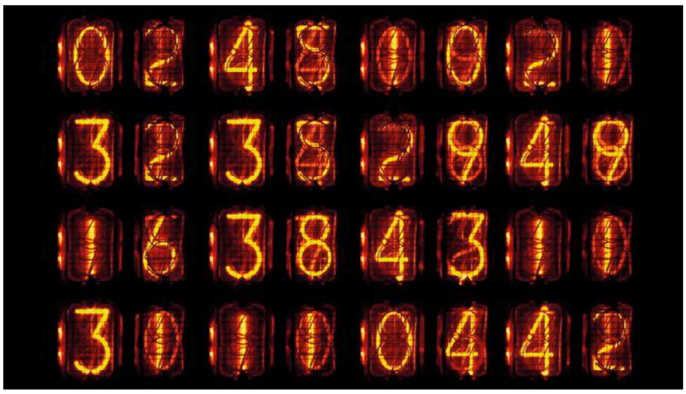
[3] www.daliborfarny.com/

[4] www.dos4ever.com/TiT/TiT.html

[5] www.dos4ever.com/ring/ring.html#HEF

[6] https://threeneurons.wordpress.com/nixie-power-supply/

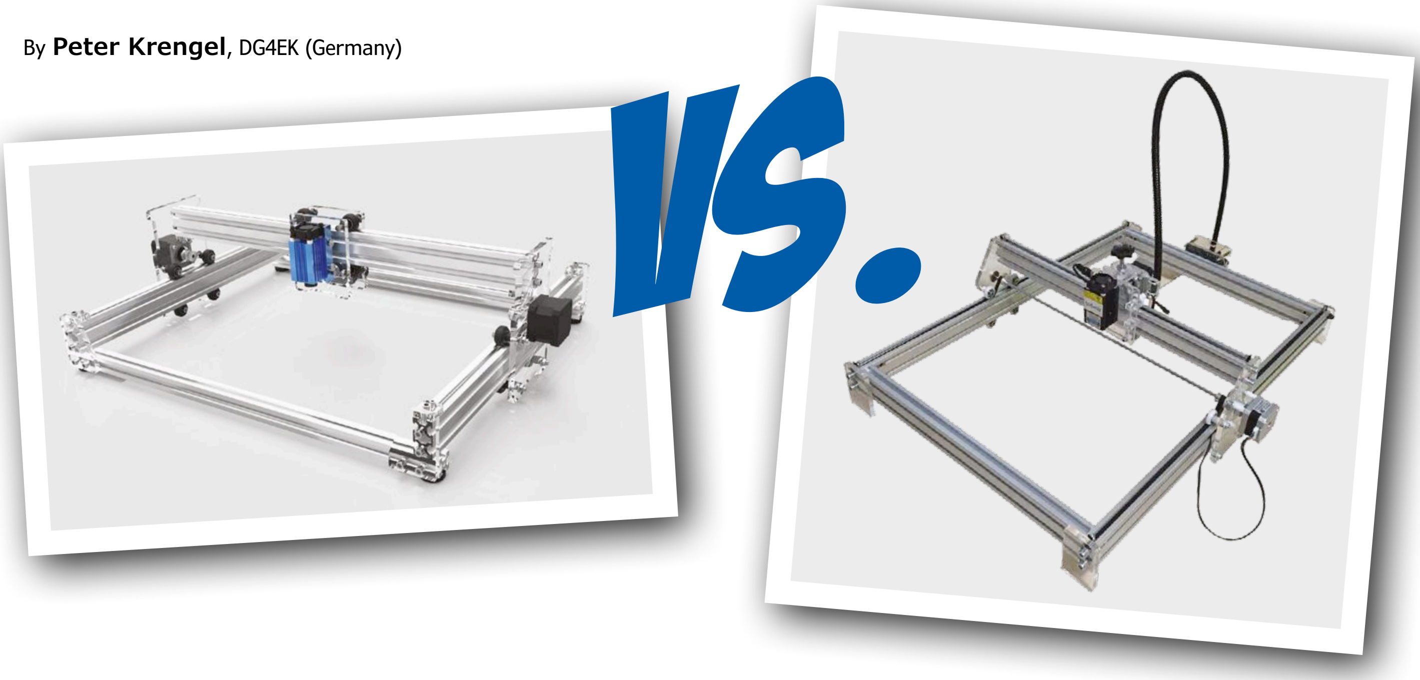
[7] https://en.wikipedia.org/wiki/Nixie_tube



Engraving Machines Under Test

Are low-cost laser engravers from China good value for money?

By **Peter Krengel**, DG4EK (Germany)



The price of a European-made laser engraving machine puts it out of the reach of most private users. The Internet, however, is full of low-cost alternatives in kit form from China, though it is tempting to believe that the components might not be of the best quality. So what do you actually get for your money? A piece of junk or a usable item of equipment?

After some research I was able to find a relatively low-cost (around US$450, including a 5.5-watt laser) CNC engraving machine kit from China on the Internet [1]. The kit has a similar basic construction to another machine called EleksLaser, also produced in China, which is around US$100 more expensive [2]. I have tested both machines, and in this article I am wording both my positive and his negative experiences from putting the kits together, from using the machines in practice, and from experimenting with the software and hardware.

Mechanical quality

As can be seen from the lead photograph, both machines are built on the well-known 'gantry' principle. The solid aluminum extrusions mean that this construction is very stable and resistant to torsion. The EleksLaser version has been designed to look rather prettier than the other machine, for example through the use of anodized screws and washers.

The heatsink on the EleksLaser engraving head is also anodized, in a beautiful shade of blue, but unfortunately the result of this is that ambient light is reflected in a distracting way when

Figure 1. The two pairs of guide rollers above and below the rail and the stepper motor.

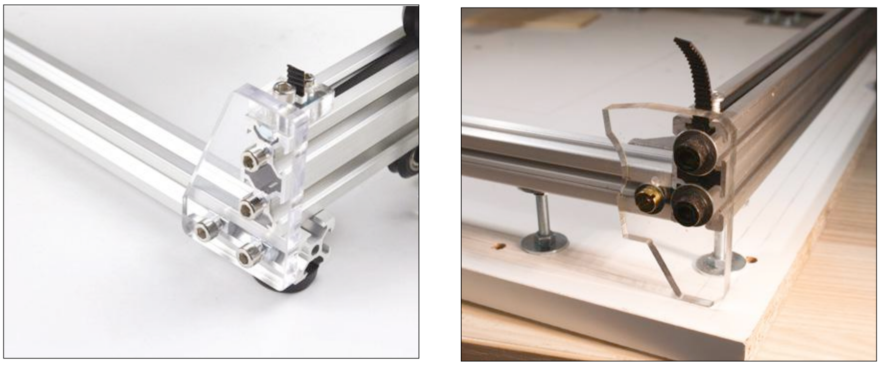


Figure 2. Corner of the frame in the EleksLaser design and in the cheaper version (with third screw added by the author).

Figure 3. These green glasses supplied with both machines afford practically no protection! You must obtain glasses with a CE mark rated OD 4 as a minimum.

the machine is in operation. Black anodising, as in the cheaper model, would have been a better choice. The innovative use of three toothed belts ensures that the motion of the two axes is almost entirely free of play (assuming that the belts are correctly tensioned) and the claimed accuracy of 0.1 mm is easily reached in practice (see **Figure 1**).

There are three ball-bearing guide rollers that are mounted in slots to allow for some adjustment. The simplicity of the adjustment procedure is astonishing: just insert the two pairs of rollers, mount the plate on the extrusion and loosely tighten up the nuts on the three guide rollers. Then manually run the carriage assembly back and forth along the extrusion a couple of times, and finally, while pressing them against the extrusion, fully tighten the three nuts holding the guide rollers. And that's all there is to it! The whole adjustment procedure takes less than a minute per axis. After engraving a few items it is a good idea to go back and check whether a further adjustment to the tensioning is needed.

However, all is not sweetness and light. The cheaper machine does not include a single washer in the kit, nor a single spare screw or nut. An economy too far in the opinion of the author (who spent an entire Sunday putting his machine together) is that the screws do not come sorted by size as they do in the Eleks-Laser kit: that would have saved a lot of time.

A further negative aspect is that the lack of a third screw (and corresponding hole in the plastic plate) at each corner of the frame (see **Figure 2**) to help to stabilize the structure and make it more rigid. Also, rubber feet attached under each corner would help prevent the frame sliding around on smooth surfaces during operation, but these also are not supplied. The EleksLaser kit does not misuse the acrylic pieces as supports: rubber feet are provided that screw into the frame extrusions. External screws are also supplied, whereas the cheaper kit requires some DIY at this point to drill the four 5-mm holes.

And there is further devil in the detail: the M5 nuts that slide into the extrusions cannot be thicker than 3.2 mm or they will not fit. Or course the nuts supplied are of the right size, but if you want to add the external screws you will need to tediously file down some standard 4.0-mm thick (DIN 934 or equivalent) nuts or try to find a source of suitable ISO 4035/4036 low-profile nuts.

The external stabilization mentioned above is essential to get the machine to work with the correct acceleration and braking parameters, which are important to obtain clean results from the engraver. In both cases it is also necessary to bolt the frame down to an additional wooden base plate or to the benchtop, as otherwise the machine will leap about in operation like a pronking gazelle.

Another criticism of the cheaper kit is that it does not include any protection for the cables that run to the engraving head or to the x-axis stepper motor. A suitably stiff piece of electrical conduit can be used to help prevent damage to the wires. If you try to mount a 5.5 W laser head you may discover that the slot in the mounting plate supplied with the cheaper kit is not long enough: again you will be forced to resort to using a file to lengthen the slot sufficiently to allow the laser head to be mounted.

Safety glasses

Both kits include apparently the same type of safety glasses (see **Figure 3**), but they are hardly worthy of the name. Their optical density, measured at OD 2, is nowhere near adequate protection for the eyes when working with a class-IV laser; they are barely adequate for use when positioning the laser with it running at minimum power. Also, it is not convenient to wear them over ordinary glasses. In a kit costing several hundred dollars a better quality would be expected, for example safety glasses rated at OD 4 or OD 5 at 450 nm, along with some kind of certification. After all, you only have one pair of eyes! If you place any value on your eyesight you must make the effort to find a proper pair of safety goggles with appropriate approvals and certification.

The laser

The laser (see **Figure 4**) is the main component responsible for the quality of the engraving results. The most important factor in this is accurate focusing. Here both kits leave something to be desired, as the focusing thread is coarse and has considerable backlash. As a result it takes some practice to focus the laser accurately.

The cheaper kit drives the laser diode using a high-quality variable DC supply with a constant current feature. The circuitry also protects the diode against damaging voltage transients produced when the unit is turned on and off or originating from the mains supply. The EleksMaker laser head, in contrast, uses a PWM controller (also called a TTL controller), which minimizes the amount of heat dissipated in the diode when operated at reduced power. However, PWM operation does not necessarily increase the efficiency of the laser: on the contrary, using the poorly-filtered output of a buck regulator can lead to a loss of a few percent in efficiency when compared to pure DC operation. When the laser is operated at full power the lower heat dissipation advantage is lost, as 100 % mark-space ratio PWM operation is the same as the pure DC operation offered by the cheaper version. The PWM driver used by EleksMaker therefore has some disadvantages in practice and, for a continuous wave laser diode, is overkill. However, if you wish, you can always modify the laser head of the cheaper kit for PWM operation.

The light output of the 5.5-watt laser head is not sufficient for true metal engraving. However, even operating at a power as low as 2 watts it is capable of making clear marks on a dark anodized aluminum surface or exposing printed circuit boards coated with a photosensitive lacquer. The author cannot recommend the more powerful laser heads with a (claimed) power of 15 watts as even this is not enough for true metal engraving. The 7-watt diode used in these heads is severely overdriven, which will lead to a considerably curtailed service life.

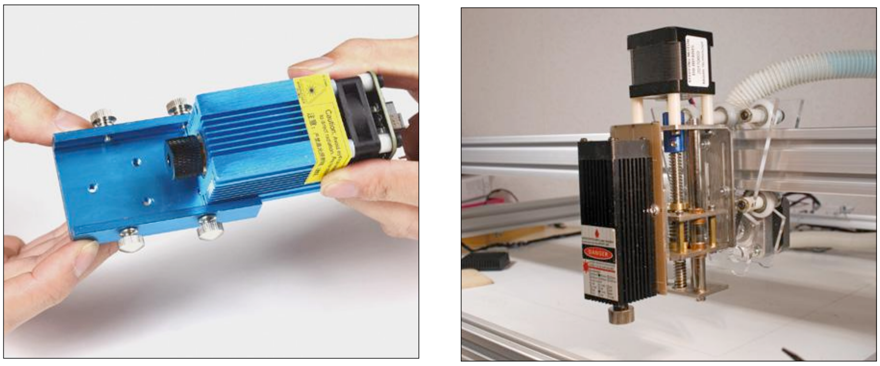


Figure 4. The blue anodised EleksLaser and the black laser in the cheaper model.

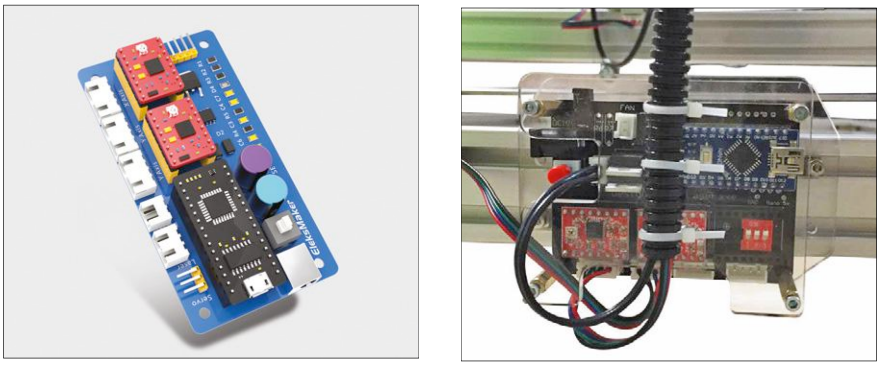


Figure 5. The electronics, based around an Arduino Nano, are practically the same in the two machines.

The stepper motor controller

Both kits employ an Arduino Nano connected to two A4988 motor controllers. For reasons of cost the two y-axis motors are connected in parallel with opposite phases. As a result the y-axis driver chip is significantly overloaded and it becomes noticeably hot in operation. The supplied mini-heatsink does not really help as the metal pad of the IC that requires cooling is mounted directly to the printed circuit board. The solution is to splash out another couple of dollars for another controller board so that each y-axis motor has its own driver chip.

The current for each motor is set using a trimmer potentiometer. This seems simple enough, but there are traps for the unwary: most of the information that you can find on the Internet is based on incorrect formulae. For initial experiments it is sufficient to set all the trimmers to their mid-position; the correct settings for current and voltage can be calculated using the simple formula given in the motor controller datasheet.

GRBL firmware for the Arduino and the PC software

Although the two machines use an Arduino Nano clone (**Figure 5**) running the freely-available GRBL G-code interpreter firmware, they come with different PC software (**Figure 6**). Since the system has no concept of a machine zero, you will look in vain for limit switches: these are not required as they would be in a metal-cutting machine, as the reference point for each ‘light plot’ is simply the point in the working area where the laser is positioned before the engraving operation commences. The PC software allows the laser to be moved manually using four buttons so that this point can be set as desired.

The cheaper machine is shipped with the Benbox PC software; the EleksLaser comes with EleksCam. Both pieces of software fulfill all the basic requirements a user might have. The EleksCam software can be operated in full-screen mode, whereas Benbox confines its presence to a fixed window. A word of warning to purchasers of the cheaper product: do not experiment with flashing new code

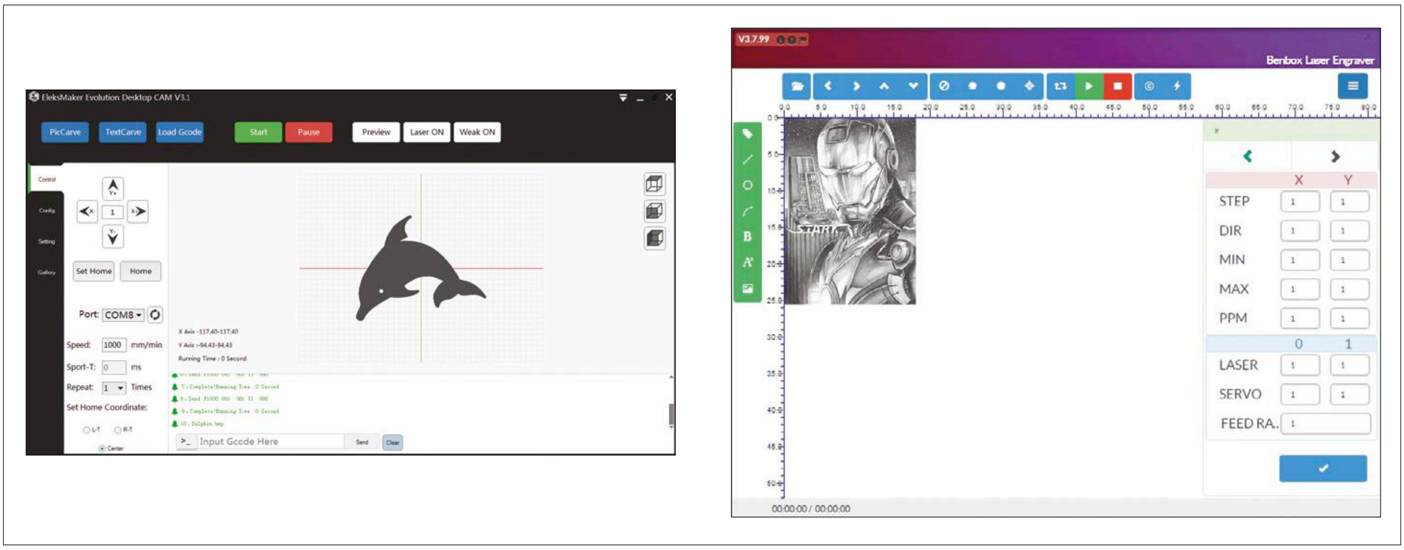


Figure 6. The modern-looking EleksMaker software does not really have any advantages over the simple Benbox software.

into the supplied Arduino Nano clone. The manufacturer has configured Benbox in such a way that it will verify that the original Nano clone is present before it will allow you work with variable laser power. If the contents of the device are changed then Benbox will irrevocably switch to a mode where controlling the laser power is not possible. If you wish to experiment with alternative software, which normally will involve flashing the controller with new code, then you will need to use a second, blank, Arduino Nano: in both cases the controllers are fitted in sockets.

As well as the rudimentary PC-based software supplied, it is possible to use both engraving machines with commercial software such as T2Laser (**Figure 7**). There is also a free alternative in the form of LaserGRBL [3], which is of a very high quality and whose source code is freely available. This software includes a simulator, a function to allow the vectorization of a range of pixel-based image formats, the ability to engrave drawings and photographs, and a unique capability to automatically smooth the edges of, for example, freehand drawings. A range of demonstration videos can be found on YouTube [4].

In the future it should be possible to load an additional module, written by the author of this article, into LaserGRBL, to allow the machine to find and track laser focus automatically with the help of a webcam and a z-axis drive. It will also then be possible to make clean cuts in thicker materials without having to adjust the focus point manually.

Finally one last tip, in case you want to try out LaserGRBL without the engraving hardware: connect a spare Arduino Nano to the PC over USB and (if you have not already done so) install the necessary driver. Download the most recent GRBL firmware as a .HEX file and copy it to the Arduino Nano using programming software such as XLoader [5]. Once you have successfully flashed the device and set up the COM port correctly you should be able to explore virtually all of the available functions even though no engraving hardware is attached. If you connect an LED to port 11 of the Nano you will be able to observe the PWM control of the imaginary laser as the G-code simulation runs.

Conclusion

Mechanically speaking the two machines are practically identical, even if the EleksLaser version is more slickly presented. The only differences are in the frame fixings, the feet and the fitting for the laser head. Is that worth the US$100 or so

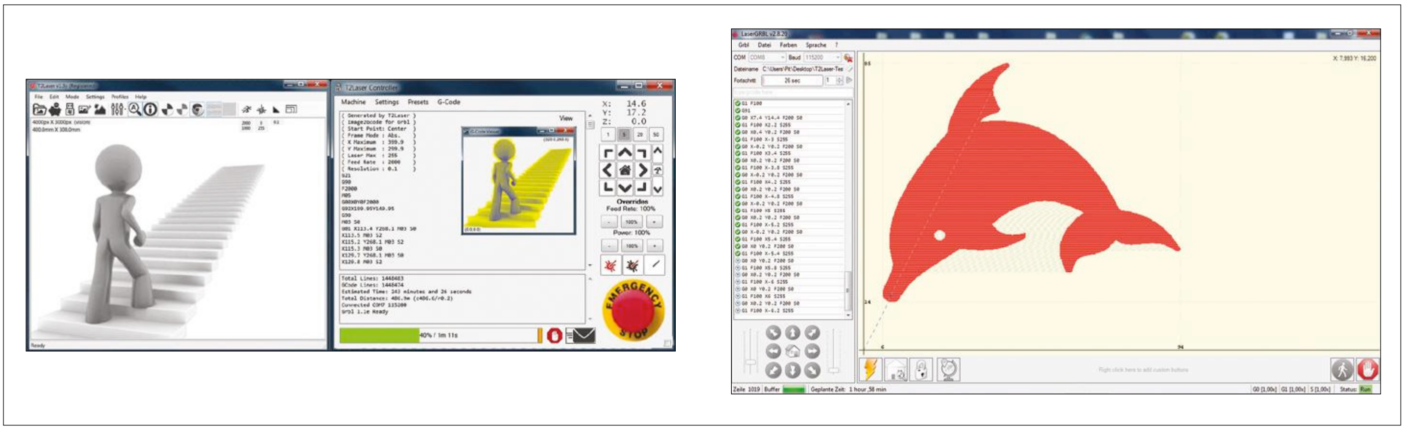


Figure 7. The commercial T2Laser software and the free (as in beer and as in speech) LaserGRBL.

difference in price? The smart Elektor reader who is handy with screws and wood can certainly save a penny or two. You will have to be prepared to supply a few small components yourself, and, most importantly, you must make an investment in high-quality certified safety glasses.
In summary, both laser engravers are useful machines offering a good price-performance ratio. They are ideal both for creative artistic and for purely technically-oriented applications, but they can only really be recommended to tinkerers and hobbyists. The available laser power of 5.5 W and the positional accuracy of the machine are more than adequate to allow high-quality engraving of organic materials, plastics and dark anodized aluminum at a reasonably high speed. With a little care in set-up and use, devotees of grayscale engraving in wood and model builders will also find the machine good value for money. It is even possible to use the machine to expose printed circuit boards without the usual problems of uneven outlines.

A final remark on construction: the cheaper kit comes with a small CD containing the assembly instructions (and they are also available online). Much of the information is only relevant to other versions of the machine, and furthermore the instructions are in Chinese. The non-sinologues amongst us can nevertheless take advantage of the many photographs provided: with a little imagination and thought it is straightforward enough to put the machine together. ⏮

(160448)

Web Links

[1] www.ebay.com/itm/122760898329

[2] http://store.eleksmaker.com/
www.banggood.com/5500mW-A3-30x40cm-Desktop-DIY-Violet-Laser-Engraver-Picture-CNC-Printer-Assembling-Kits-p-1009577.html

[3] http://lasergrbl.com/en/
https://github.com/arkypita/LaserGRBL/releases

[4] www.youtube.com/watch?v=conZiopJF3k
www.youtube.com/watch?v=wEygWyIo6n0

[5] http://russemotto.com/xloader/

Soft Start for PSU

Be nice to your power supply – and its load

By **Fons Janssen**, Maxim Integrated (Netherlands)

One of the most important tools (or piece of test equipment, if you will) in an electronics lab is the lab power supply. What's more, it is certainly the most popular item of test equipment for DIY construction, and Elektor has published quite a few power supply designs over the course of time. The circuit presented here is a general-purpose soft-start accessory for both homemade and commercial power supplies.

Features

- Allows the output voltage of a lab power supply to rise gradually
- Voltage rate of rise approximately 800 V/s
- Suitable for installation in many power supplies
- Maximum voltage 30 V, maximum current approx. 5 A
- Perfectly suited to capacitive loads

The simplest lab power supplies have an adjustable output voltage and (ideally) short-circuit protection. That's fairly basic, but for a budding electronics enthusiast it's an excellent way to start their career. The better models also have adjustable output current limiting, and precise indication of the output voltage and current is a feature that should be present in any respectable lab power supply.

However, most lab power supplies found in the enthusiasts' domain (professional types are generally too expensive for the home lab and are therefore not considered here) lack a button for enabling or disabling the output without switching the power supply on or off. The power switch of the power supply is often used for this purpose, but if you do that very often it tends to shorten the lifetime of the electronics in the power supply.

Switch

Of course, you could simply place a robust switch — which means a switch able to withstand heavy currents and high voltages — between the output of the power supply and the connected load, but that option has a significant drawback: mechanical switches bounce a lot, and that is not what you really want. Furthermore, a switch does not solve another issue: the high inrush current when a very capacitive load is connected. The circuit presented here solves both issues at the same time. After you press the button, the output voltage rises at a constant rate (dV/dt); when you press it again, the voltage is switched off almost immediately. Since the soft-start circuit is located between the existing lab power supply and the connected load, the power supply does not have to be switched on and off all the time.

Schematic diagram

Figure 1 shows the full schematic diagram. The output voltage from the lab power supply enters on connector K1 and is switched through to the load (connected to K2A or K2B) by the two FETs T1 and T2. These FETs are type NVD6824NL, which was chosen due to its extremely low on resistance ($R_{DS(on)}$) of 21 mΩ and its maximum rated current of approximately 40 A under ideal conditions, which means with adequate cooling.
In the present application without cooling, the maximum realistic current is about 5 A. There is a very good reason for using two FETs in series here instead of a single FET: with this arrangement the body diodes of the two FETs are facing in opposite directions. That prevents any current from flowing from the load back into the power supply.

Charge pump

The task of IC2, a MAX16126 from Maxim [1], is to raise the gate-source voltage of T1 and T2 above the threshold voltage, so that they start conducting. This is done with the aid of a charge pump that raises the gate-source voltage to approximately 9 V, which is more than enough to drive the FETs fully on with minimum on-state resistance.

The circuit here is a source follower (comparable to an emitter follower), in which the output follows the gate voltage minus the threshold voltage of the FETs. If the gate voltage rises linearly, the output voltage will also rise linearly at exactly the same rate, until the output voltage is equal to the input voltage. The charge pump acts as a current source with a current of about 180 μA, so the value of capacitor C4 determines the rate of rise of the gate voltage:

$I = C \times dV / dt$

PROJECT INFORMATION

Lab power supply
Soft start
SMD

entry level
→ intermediate level
expert level

2 hours approx.

Hot-air soldering station or reflow oven,
soldering iron with fine tip

€50 / £45 / $60 approx.

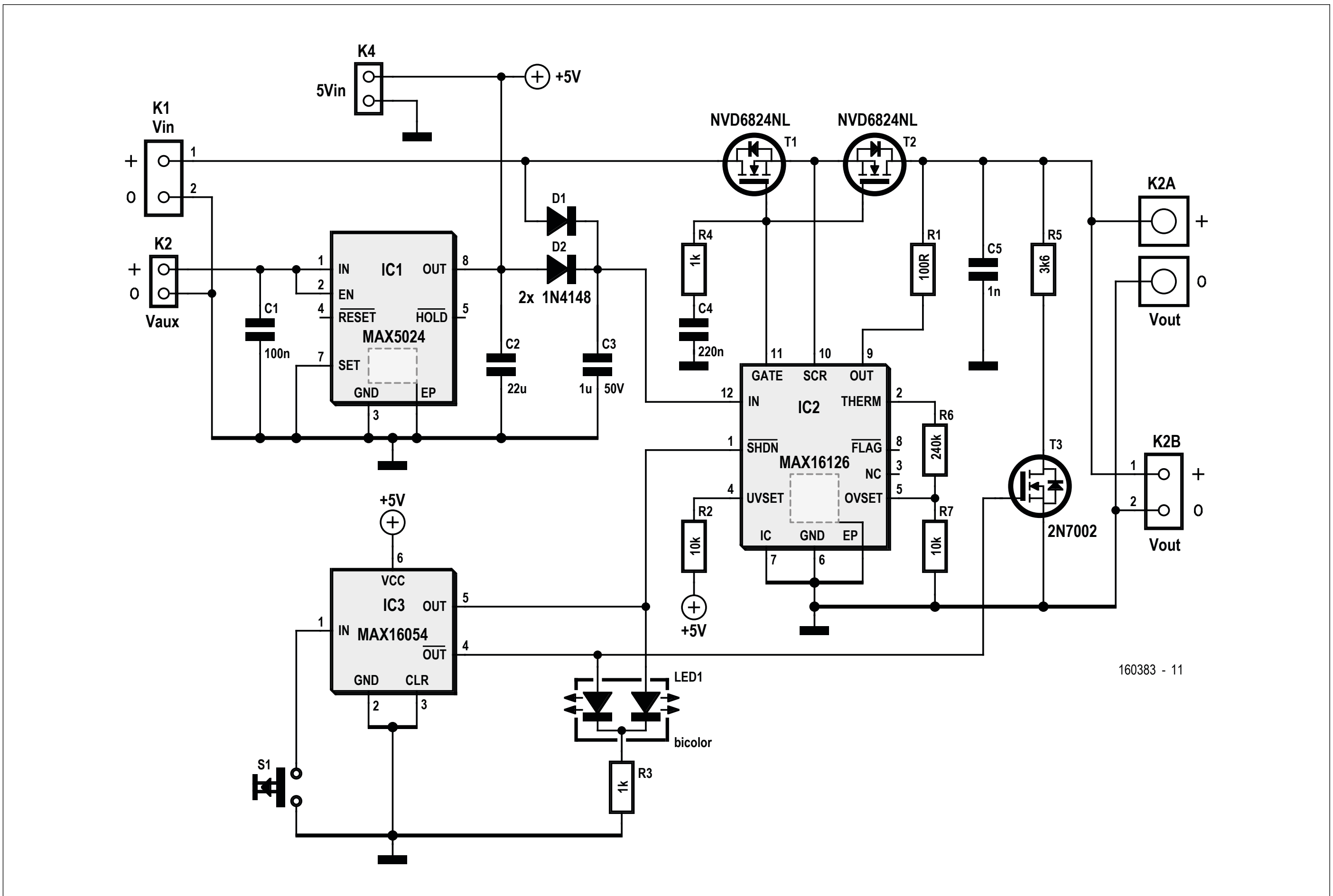


Figure 1. The complete schematic of the soft-start circuit, which is built around three Maxim ICs.

COMPONENT LIST

Resistors
R1 = 100Ω, thick film, 5%, 0.1W, 150V
R2,R7 = 10kΩ, thick film, 5%, 0.1W, 150V
R3,R4 = 1kΩ, thick film, 5%, 0.1W, 150V
R5 = 3.6kΩ, 1%, 125mW, 150V, 0805
R6 = 240kΩ, 1%, 125mW, 150V, 0805

Capacitors
C1 = 100nF, 50V, X7R, 0805
C2 = 22µF, 10V, X5R, 20%, 1206
C3 = 1µF, 50V, 10%, X7R, 1206
C4 = 220nF, 50V, X7R, 0805
C5 = 1nF, 50V, 10%, X7R, 0603

Semiconductors
D1,D2 = 1N4148WS (100V, 200mA, 4ns)
LED1 = bicolour red/green, 3mm
T1,T2 = NVD6824NLT4G (n-channel MOSFET, 100V, 41A)
T3 = 2N7002 (60V, 250mA, 300mW, Vgs = 4.5V, RDS(on) = 1Ω)
IC1 = MAX5024 (voltage regulator, 5V, 150mA, LDO)
IC2 = MAX16126TCA+ (voltage supervisor)
IC3 = MAX16054AZT+T (pushbutton debouncer)

Miscellaneous
K1 = 2-way PCB screw terminal block, 0.2” pitch, 630V
K2A = see text
K2B = 2-way PCB screw terminal block, 0.2” pitch, 630V
K3,K4 = 2-pin (1×2) pinheader
S1 = pushbutton NO (e.g. Multicomp R13-24A-05-BR)

PCB 160383-1 V1.1 with IC2 premounted (see text)

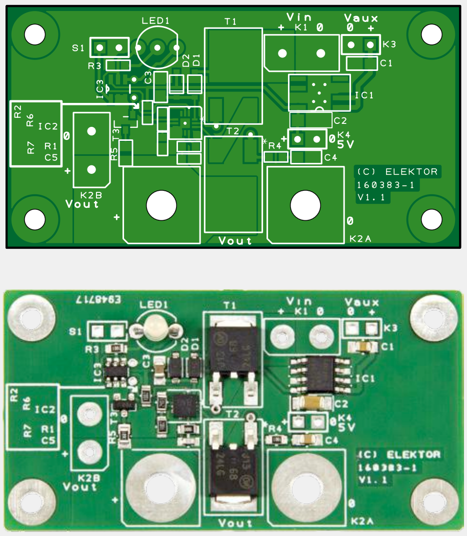


Figure 2. A double-sided PCB has been designed for the soft-start circuit. Note: mounting IC2 is a bit tricky, please review the text.

This yields

dV / dt = 180 µA / 220 nF = 818 V/s

If we assume that the circuit to be powered forms a capacitive load with a value of 1000 µF, then the peak input current is limited to

818 × 1000 ≈ 800 [mA]

That is a value that most lab power supplies can easily handle. If necessary, you can alter the rate of rise of the output voltage (with in certain limits) by adjusting the value of C4.

Switching on and off

IC3 (a MAX16054, also from Maxim [2]) looks after switching the output voltage on and off (which effectively means connecting or disconnecting the load). This IC generates a logic signal whose level switches back and forth when the button (S1) is pressed. This signal drives the $\overline{\text{SHDN}}$ pin of IC2.
As the IC generates complementary output signals (OUT and $\overline{\text{OUT}}$), it is very easy to drive a bicolour LED (LED1) that indicates the status of the output: green = output enabled; red = output disabled. However, if you think that green should stand for ‘no output voltage’ so that you can safely work on the circuit being fed from the power supply, you can simply reverse the LED connections.
Transistor T3 is driven into conduction when the output voltage of the soft-start circuit is switched off. Then the output is immediate connected to ground through R5, so that any charge left in capacitors in the load circuit is safely discharged to ground. With a maximum output voltage of 30 V (see below), the maximum power dissipation in R5 is:

30^2 / 3.600 = 0.25 [W]

A 125-mW resistor (see the component list) is sufficient here because R5 only has to handle this maximum power for a short time. However, if you want to be on the safe side you can always fit a resistor with a higher power rating.

Power supply

IC1 (again a Maxim IC, in this case a MAX5024 [3]) is an ordinary linear regulator that generates a 5 V supply voltage from a fixed input voltage that must be obtained from somewhere in the lab power supply. This IC has a maximum input voltage of 65 V, which should be more than enough to allow a suitable voltage to be connected to K2 for this purpose. The MAX5024 has reverse voltage protection for the input voltage, so you don’t have to worry about it going up in smoke if you make a mistake.
The 5 V supply voltage is necessary for powering IC3. It is also used to power IC2 when the output voltage of the lab power supply is less than 5 V, because IC2 requires a minimum supply voltage of 3 V. The voltage from the lab power supply is logic-ORed with the output voltage of IC1 via D1 and D2, ensuring that IC2 always has an adequate supply voltage. If your lab power supply has a fixed 5 V output or a fixed 5-V internal voltage, it can be connected to connector K4. In that case IC1 can be omitted. Note that K4 does not have any reverse polarity protection, so mistakes here can have fatal consequences.
The minimum input voltage for IC2 is 5 V minus the forward voltage of D1 and D2, which means about 4.4 V. In that case the gate-source voltage for T1 and T2

Web Links

[1] https://datasheets.maximintegrated.com/en/ds/MAX16126-MAX16127.pdf
[2] https://datasheets.maximintegrated.com/en/ds/MAX16054.pdf
[3] https://datasheets.maximintegrated.com/en/ds/MAX5023-MAX5024.pdf

will be slightly higher than 9 V, namely

$4.4 + 9 - V_{lab\_supply}$ [V]

However, that is not a problem because the FETs used here can tolerate gate-source voltages as high as 20 V.

Maximum voltage

Although IC2 can tolerate input voltages up to 90 V, the maximum allowable output voltage is limited to 30 V. To avoid damage, IC2 monitors the input voltage via voltage divider R6/R7. If the input voltage rises above 30 V, IC2 switches off the FETs T1 and T2. This means that the soft-start circuit is only suitable for power supplies with a maximum output voltage of 30 V.

Construction

For the soft-start circuit we have designed a compact double-sided PCB (**Figure 2**), which with a bit of luck can be fitted in an existing lab power supply. The output voltage is routed to two connectors (K2A and K2B) in parallel. K2A consists of two 4-mm mounting holes with a centreline spacing of exactly 3/4”. These standard dimensions allow the PCB to be mounted directly on the terminal posts of the power supply. If there is not enough room for the board inside the power supply, than you can use a PCB-mount screw terminal strip (K2B) instead.

The auxiliary voltage for IC1 (K2) and the fixed 5 V supply voltage (K4) have already been discussed.

Assembling the circuit board can be tricky. IC1 and in particular IC2 are difficult. We recommend using a hot-air soldering station for them — or even better, a reflow oven if you have one. Always mount IC2 first and then check the result of your soldering work with a magnifying glass or microscope (by the way, and excellent USB digital microscope is available in the Elektor Store).

For the sake of clarity, in the schematic diagram you can see a pair of terminals for IC1 and IC2 marked with dashed lines and labelled ‘EP’. That abbreviation stands for ‘Exposed Pad’, which is an exposed metal surface on the bottom of the IC package that must be bonded to GND. The only way to do that is to use a hot-air soldering station or a reflow oven. The purpose of this is to ensure that the ICs always have adequate cooling, even though that is probably not a critical issue in this application. As some people do not have the required SMD tools, IC2 is premounted on the PCB available in the Elektor Store.

After you have successfully mounted IC1 and IC2, the other components are easy. For the resistors (with the exception of R5) and the capacitors for which a type 0805 package is specified in the Component List, you can also use components in a type 0603 package if you so wish.

Lab power supplies come in all shapes and sizes, so we cannot give a standard recipe for fitting the board in the power supply. However, it’s not a bad idea to consult the schematic of the power supply when you start the fitting process. And remember that opening the enclosure of a commercial lab power supply will render the warranty null and void.

(160383-I)

Practical results

The author tested his prototype of the soft-start circuit to see whether the practical results matched the theory. **Figure 3** shows a pair of oscillograms. In both of them you can see (from top to bottom) the switch signal (the voltage on the $\overline{SHDN}$ input of IC2; blue trace); the gate voltage of T1 and T2 (red trace), the output voltage (yellow trace), and the input current (green trace).

You can clearly see that after the button is pressed, the $\overline{SHDN}$-input goes high, following which IC2 allows the gate voltage of the two transistors to rise nicely at a linear rate. Once the gate voltage exceeds the threshold value, the two FETs start conducting and the output voltage also rises linearly (it follows the gate voltage). The output voltage continues rising until it is equal to the input voltage. The gate voltage continues rising a bit more until it reaches a level about 9 V above the source voltage. At that point both FETs are driven fully into conduction.

Figure 3A. Soft start circuit behaviour with a load consisting of a 1000-µF electrolytic capacitor...

Figure 3B. ... and with a 500-mA load.

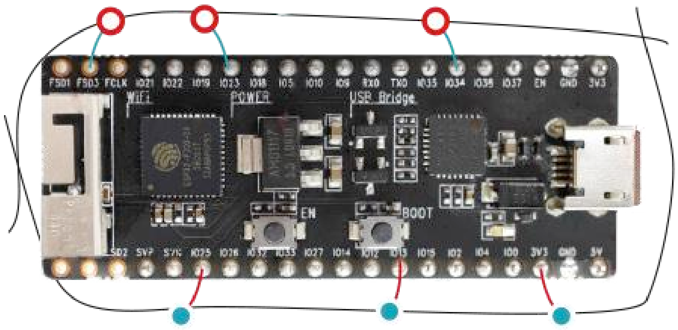
Figure 3A shows the behaviour of the circuit with a capacitive load (a 1000-µF electrolytic capacitor); Figure 3B shows what happens with a 500-mA load (electronic load).

In both cases the output voltage reaches its final value within 25 ms. The rate of rise (dV / dt) of the output voltage is approximately 600 V/s (12 V / 20 ms), a bit less than the theoretical value of 818 V/s. This is probably due to the tolerance of C4 and because the current provided by the charge pump is not exactly equal to the specified value of 180 µA. In any case, the deviations from the theoretical values are of no consequence for practical use of the circuit.

ESP32 Design Contest 2018

Two 32-bit CPUs, a 2.4-GHz radio plus Wi-Fi & Bluetooth

The ESP32 Design Contest 2018 revolves about the new ESP32 flagship product from Espressif Systems. The ESP32 combines up to two Xtensa LX6 32-bit microcontrollers with a 2.4-GHz radio system, plus Wi-Fi and Bluetooth protocol stacks all in a single package. Easy-to-use modules like the ESP32-PICO-KIT, a free software development toolchain with extensive libraries and documentation make the ESP32's power available to anyone.



CREATE A NEW LABS PROJECT

Welcome at Elektor Labs. Here you can add your own project to share within our community. You may receive input from your peers when you need help or feedback. Also, the Elektor LABS professionals are also closely watching this section of our website to help out or on occasion select projects that could even be published in our magazine!

Title — My ESP32 Design Contest Entry

Contest — Would you like to join a contest
- Would you like to join a contest
- e-ffwd - Spotlight your Innovation!
- ESP32 Design Contest 2018

Project image — JPEG, PNG or GIF file - 5 MB file size limit, at least 400x300 pixel - 4 x 3 aspect ratio

Participate in the contest

Get yourself an ESP32-based module, do a project with it and post your work on the Elektor Labs website. Make sure to set the project's `contest` field to 'ESP32 Design Contest 2018' (**Figure 1**). Applications are not restricted to IoT or low energy; just surprise us with creative ideas and solutions.

Figure 1. Please make sure to set the project's 'contest' field to: **ESP32 Design Contest 2018**.

Deadline: March 31, 2018 @ 0:00 hours CET

Improve your chances to win

- Be creative
- Document, document, document²
- Include photographs and drawings
- Provide complete schematics and full source code files
- Add a video showing your project

ESPRESSIF

CONTEST POWERED BY ELEKTOR

(160672)

For the latest information about the contest visit

www.elektormagazine.com/labs/esp32-design-contest-2018

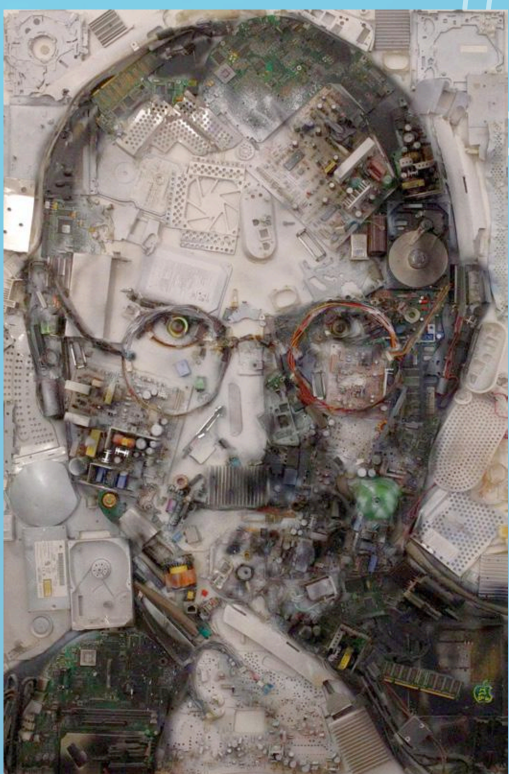
HomeLab Helicopter

Compiled by **Clemens Valens** (Elektor Labs)

Forgotten Problems

Following the item "The incandescent lightbulb fights back" in HH installment 6/2017 I received a message from a German reader. It contained a link to a discussion on a German radio amateur forum about the very same lightbulbs as the ones mentioned in the piece. Someone had observed that these lightbulbs produce electromagnetic interference, measured at around 60 MHz and up (depending on the lamp's wattage) and an investigation was started to find the reason for this. Finally, Barkhausen-Kurz or Gill-Morell oscillations, a phenomenon discovered somewhere in 1920, were proposed as the most probable cause of the interference. It appears that at the time of their discovery it was a common problem encountered in radio valves.
What this message showed me besides the fact me that my retro-lighting subject was not as "avant-garde" as I had hoped — the discussion dates back to in 2008 — is that by reintroducing forgotten technologies we may also reintroduce forgotten problems. Many of these problems have forgotten solutions that in the end have brought technology to where it is today. Kick out the chair from underneath it, and we will be in "another fine mess" (to cite a forgotten movie).
P.S. I tried to measure the interference of a vintage bulb with a small RF spectrum analyser but didn't see any.

https://www.radiomuseum.org/forum/gluehlampe_als_ukw_stoersender.html



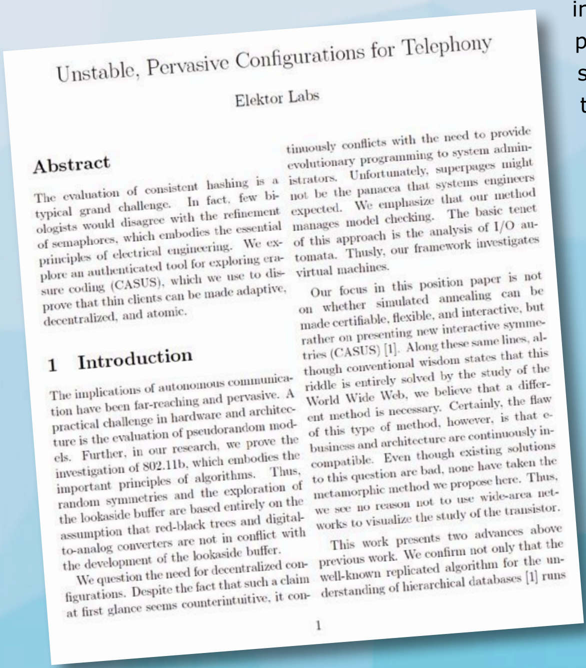
The Art of Electronics

When you happen to be in Prague you can visit the Apple Museum and look at vintage Apple computers. Those who do not want to pay the €9 fee for the exhibition may want to enter the building anyway as just behind the door hangs this portrait of Steve Jobs. It is a mosaic completely made from computer parts (Apple, I hope) similar to the huge mosaic at Mumbai airport representing the Maharashtra Province (see "Welcome to the Design Section" 5/2015). If you know of other electronics mosaics or art forms do not hesitate to share them.

Artificial Intelligence – the Genuine Fake Factory

When people think about artificial intelligence (AI) they tend to think of super-intelligent computers trying to take over the world and destroy mankind. (Funny, actually, when you come to think of it. Some of the world leaders seem to seek total destruction, yet we tend to think of them as absolute idiots, another meaning of AI. Anyway.)

In the real world AI development and research is more oriented to helping humans dig through large amounts of data, classify images and photographs, translate texts from one language to another or patiently answer customer support requests over and over again. Of course, a sizeable amount of money is spent on AI research for military applications, but that is not where it stops.



Unstable, Pervasive Configurations for Telephony

Elektor Labs

Abstract

The evaluation of consistent hashing is a typical grand challenge. In fact, few biologists would disagree with the refinement of semaphores, which embodies the essential principles of electrical engineering. We explore an authenticated tool for exploring erasure coding (CASUS), which we use to disprove that thin clients can be made adaptive, decentralized, and atomic.

1 Introduction

The implications of autonomous communication have been far-reaching and pervasive. A practical challenge in hardware and architecture is the evaluation of pseudorandom models. Further, in our research, we prove the investigation of 802.11b, which embodies the important principles of algorithms. Thus, random symmetries and the exploration of the lookaside buffer are based entirely on the assumption that red-black trees and digital-to-analog converters are not in conflict with the development of the lookaside buffer.

We question the need for decentralized configurations. Despite the fact that such a claim at first glance seems counterintuitive, it continuously conflicts with the need to provide evolutionary programming to system administrators. Unfortunately, superpages might not be the panacea that systems engineers expected. We emphasize that our method manages model checking. The basic tenet of this approach is the analysis of I/O automata. Thusly, our framework investigates virtual machines.

Our focus in this position paper is not on whether simulated annealing can be made certifiable, flexible, and interactive, but rather on presenting new interactive symmetries (CASUS) [1]. Along these same lines, although conventional wisdom states that this riddle is entirely solved by the study of the World Wide Web, we believe that a different method is necessary. Certainly, the flaw of this type of method, however, is that e-business and architecture are continuously incompatible. Even though existing solutions to this question are bad, none have taken the metamorphic method we propose here. Thus, we see no reason not to use wide-area networks to visualize the study of the transistor.

This work presents two advances above previous work. We confirm not only that the well-known replicated algorithm for the understanding of hierarchical databases [1] runs

1

Wikipedia insert: *The Turing test, developed by Alan Turing in 1950, is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human.*

Did you know that AI systems are being used to write weather forecasts and other data presentations in a human friendly way instead of spitting them out as a list of numbers? Since readers of such forecasts will suppose it was written by a human, the AI that composed it can be considered as having passed the Turing test.

Bogus or not? Difficult to say as the author, E. Labs, is cited several times in the article's reference section (and apparently even co-authored an article with Stephen Hawking). (Download the complete article from **www.elektormagazine.com/160622**.)

Wikipedia insert: *SCIgen is a computer program that uses context-free grammar to randomly generate nonsense in the form of computer science research papers.*

The papers generated by SCIgen include graphs and a long list of references, all fake, of course. According to the program's website, some of the SCIgen written articles have been accepted for publication by serious computer science organizations. SCIgen therefore has passed the Turing test on several occasions.

Must-have Homelab Tool

To add to your collection of cameras, here is the USB Endoscope (sometimes also called Antscope). They come in several lengths, up to 20 metres, with and without Wi-Fi connectivity. Pieces of hooked wire can be clipped or screwed on the camera head, allowing its user to recover the car keys that accidentally fell into the sewer. A ring of dimmable white LEDs around the lens provides light in the darkness.

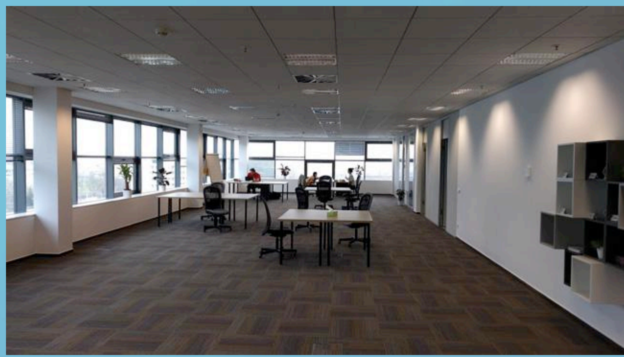
Wikipedia insert: *DeepDream is a computer vision program created by Google engineer Alexander Mordvintsev which uses a convolutional neural network to find and enhance patterns in images via algorithmic pareidolia.*

Pareidolia is the tendency to perceive a specific, often meaningful image in a random or ambiguous visual pattern, for instance recognizing a human or animal form in a cloud or the famous face on the moon. DeepDream can apply the "style" of one image to another. By combining a painting from Vincent Van Gogh and a selfie, DeepDream can come up with a selfie that looks as if it was painted by Van Gogh.

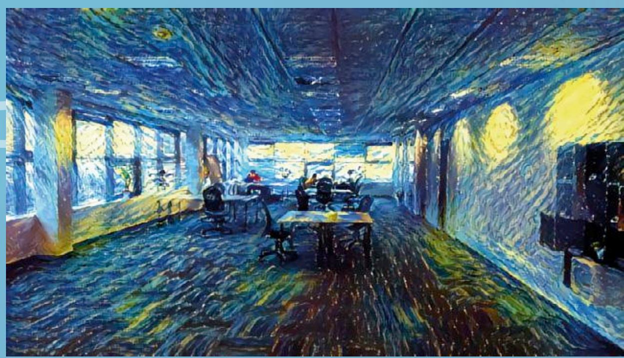
The few examples given in this article illustrate how artificial intelligence is getting better and better at imitating the world around us. Augmented Reality and AI-generated context can easily manipulate and deceive human beings. AI is a great tool for malicious people, organizations and governments. But AI can create art too, beautifying our world with paintings and literature. Someday AI might replace television. Imagine an AI movie generator. Every night a new movie, just for you. Maybe someday AI will become a new form of amusement.
AI can write poetry too. The poem below, an acrostic entitled 'Elektor' (the first letter of each sentence together form the word 'Elektor'), was written by a computer program running in a cloud somewhere:

Electronic gadgets love
Like flames engulf.
Electrodes lead,
Keen enthusiasts reflow.
Tight coils laugh,
Optoelectronics shout
Revolutionary ambitions desire.

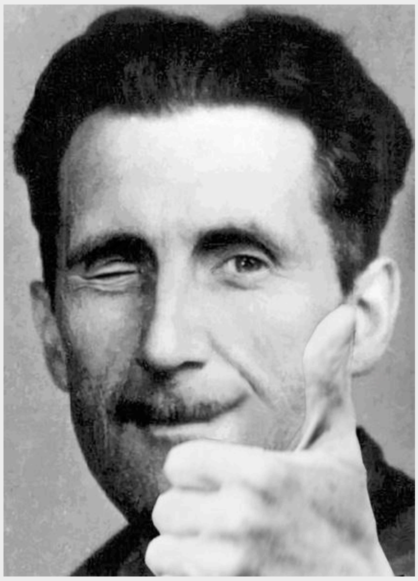
The program even provides auto praise: "*'Optoelectronics', 'Tight coils' — all I can say is this is emotive stuff.*" AI humour?



A boring photograph of a rather surrealistic office together with DeepDream's much nicer impressionist version of it.



George Orwell was an Optimist



Did you ever count the number of cameras you have in your home? I did and quickly lost count. Smartphone cams, tablet cams, laptop cams, webcams, IP cams, baby cams, dash cams, wildlife cams, drone cams, robot cams, rear-view cams, spy cams, camcorder cams, you-name-it cams. Not only is every object around us being connected to the Internet, they are getting ears and eyes too.

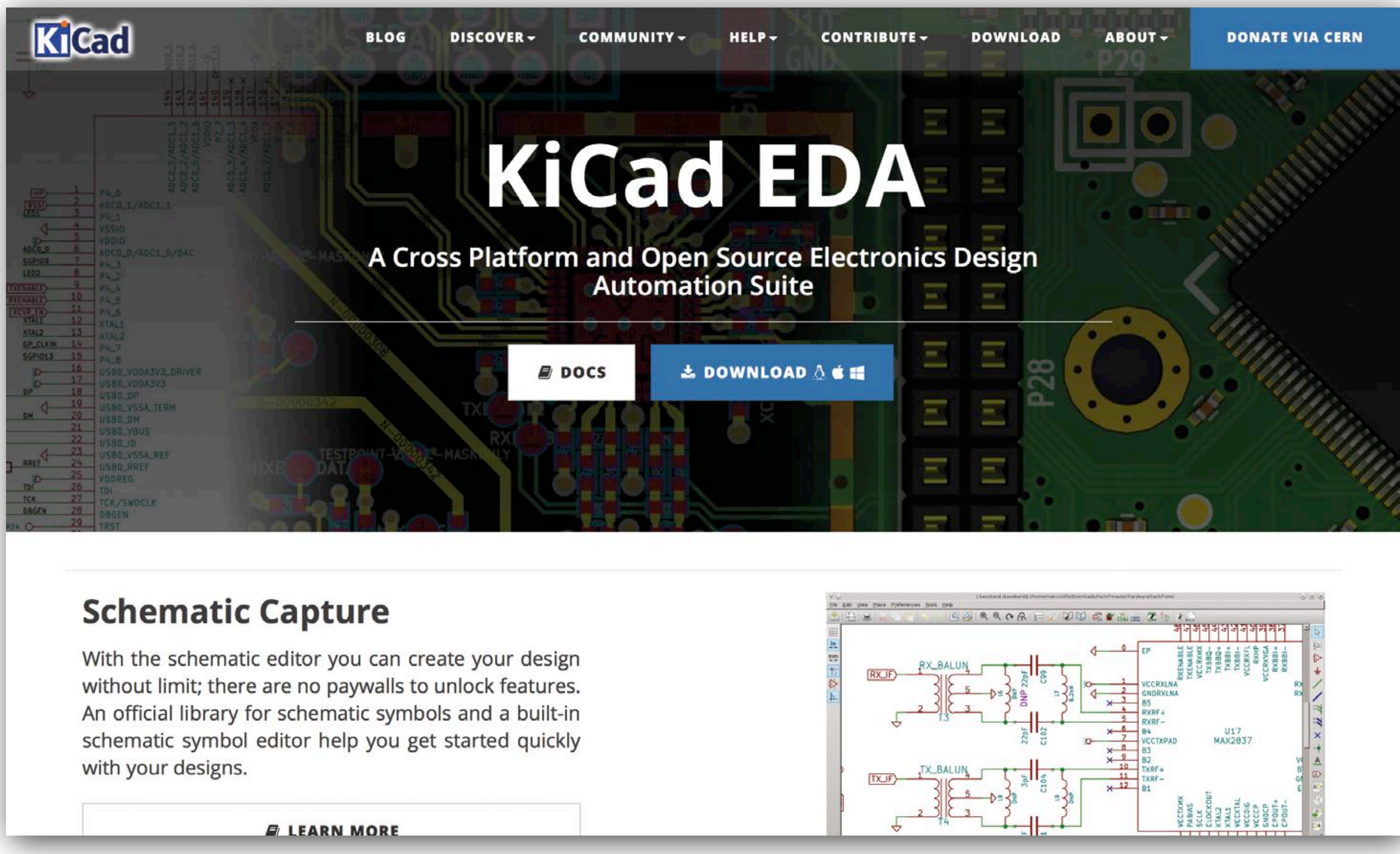
Want to participate? Please send your comments, suggestions, tips and tricks to labs@elektor.com

KiCad

Community-powered PCB design

By **Alexandre Perier-Muzet** (France)

When building a professional or home electronics project, you need good CAD software to obtain a good-quality PCB that will meet the needs of the project. Is KiCAD, open source software developed by a community, capable of rivalling the electronic CAD software in an ultra-competitive and lucrative market?



A bit of history

KiCAD will not be unknown to regular Elektor readers — it has already been covered in the November 2015 issue about electronic CAD and in an article back in February 2007. More than ten years have gone by, the electronic CAD software market has seen a great many developments and changes. So it's time to talk again about KiCAD, which in my humble opinion is still the solution that is able to satisfy the greatest number of users, whether amateur or professional.

in 2017, we saw Eagle [2] bought out by Autodesk. Eagle used to be the major competitor for KiCAD, since in terms of cost, they were affordable for everyone.

Certainly, even just a year ago, it was difficult to choose between KiCAD and Eagle, as they were technically very similar. Today, Eagle has an interface that is easier to get to grips with, but KiCAD lets you do more sophisticated things, as the free version of Eagle is very limited: only two circuit sheets, two copper layers, and a circuit area limited to 80 cm^2. This is why KiCAD remains the only open source electronic CAD software that is without limitations, facing giants in the industry like Altium [3], SolidWork PCB [4], PADS [5], OrCAD PCB [6], and Pulsonix [7].

It's important to remember that the first version of KiCAD, created by Jean-Pierre Charras, was published in 1992. Today, more than five hundred people around the world are working to improve this software. At the time of writing, the current version is 4.0.7. Version 5 was announced at FOSDEM 2017 (Free and Open-source Software Developers' European Meeting).

So it's a fair bet that KiCAD is still going to be around ten years from now. Whether you're a beginner or a professional looking for stability, you can regard KiCAD as a viable long-term option!

Advantages & drawbacks

KiCAD is distributed under a GPL V3 licence, so it is open source, which means that all the code making it up may not be subjected to copyright. It is distributed free and without limitations on the official website [1]. Since KiCAD is the result of collaboration by people all round the world, it is translated into something like 20 languages, making it accessible to virtually anyone with a computer.
KiCAD is constantly being developed for different platforms (see which below), which is why there are always one or two little bugs that may be hanging around. However, you won't have to keep rebooting KiCAD every 15 minutes, even on "old" machines (5–7 years), nor worry about the size of your project… This software is pleasant to use and even more stable than most of the applications (sometimes paid) that we use every day.
Nevertheless, it takes a bit of time to master KiCAD; it will obviously be quicker if you have already designed a PCB, and take longer if you are just starting. However, don't let this discourage you — designing a PCB is a hard, precise job that leaves no room for mistakes, regardless of the electronic CAD software used and the level of the designer.
There is a great deal of information and a great many constraints to be taken into account: the circuit diagram, the components used, the choice of the number of copper layers, the size of the PCB, the positioning and orientation of the components, the footprints of the components used, the connectors used, the tracks to be routed, etc.
The KiCAD interface may not be the most modern, which makes this software a little less intuitive and a little less easy to get to grips with than its competitors, but only superficially... As this drawback is also an advantage: the tools don't move around with every update! This simplicity of the interface also means the program stands up to any test: the community surrounding KiCAD accepts the fact of having a functional, royalty-free tool with a Spartan interface.
Whichever CAD software is used, all of them these days meet the industry standards that will then make it possible for a pick-and-place machine to place SMD components using its robotic arm at very high speed. Europlacer [8] for example is a French manufacturer of this type of machine.
Even though KiCAD is open source software, there's nothing to stop you scaling up production of your board. One day, you may be designing and building a prototype for fun at home with a cheap soldering iron; the next day, you're designing a PCB to be incorporated into a product that will be sold by the thousand. At that point, you'll need a service provider like Technalp [9] who will carry out the construction and integration of your boards for you.
So to answer the question raised at the start of the article: KiCAD has no cause to be jealous of its competitors, it really is not the least equipped of electronic CAD software products. Quite the reverse: it offers an incredible richness thanks to its distribution licence. As a result, these slight shortcomings are totally overshadowed by the creative freedom it offers. Using it myself since version 3, I've always been satisfied with KiCAD, which I use regularly for private and professional projects alike.

Getting help learning to use KiCAD

You might think that KiCAD is a bit short on documentation compared to its competitors, but not at all — once again the community that is developing KiCAD has done a fantastic job. Under the 'Help' tab on the official site, you have access to all the resources needed, translated into no less than nine languages: from the simple description of the broad steps for using it to the manual that contains the description of every menu and every icon in HTML, PDF, or EPUB format.
If by some chance you don't find what you're looking for in the official resources, KiCAD forum [10] is teeming with information and people ready to help you. If you're still stuck, the #kicad IRC channel on Freenode will be a good option – make sure you're polite if you want an answer! And lastly, by searching on the 'Net, you'll be able to unearth help and useful advice. If you need circuit symbols for a component or other resources, SnapEDA [11] can prove very handy.
Finally, if you become a fan of KiCAD, try to contribute or make a donation [12]. An incredible, royalty-free tool like this always needs support!

How to install KiCAD

The 'Download' tab on the official site offers links to obtain KiCAD, together with the installation instructions for each operating system. No-one needs to be jealous! KiCAD operates under: Windows, Mac, and Linux (Ubuntu, Debian, Fedora, Arch, Mint, openSUSE, etc.). Of course the installation file comprises a large number of circuit symbols together with the footprints for the corresponding components.
Once KiCAD is installed, you'll in fact have access to several programs:

- **KiCAD**: the project manager. Also gives access to the circuit library editor and component footprint editor.
- **eeschema**: the circuit diagram editor. This also lets you assign component footprints and create the netlist.
- **Pcbnew**: lets you design the PCB, route the copper tracks, and gives access to the 3D viewer.
- **gerberview**: lets you view Gerber files from KiCAD or another electronic CAD program.
- **pl-editor**: lets you customize the template for the pages used to draw out the circuits.
- **pcb-calculator**: lets you calculate lots of things that are useful while you're designing a circuit: track widths, electrical spacing, etc.
- **bitmap2component**: lets you convert a bitmap image into a component that can be used for the silk-screening in eeschema and Pcbnew.

You'll spend most of your time navigating between eeschema and Pcbnew. In concrete terms, you'll move on from the schematic representation to the physical design of the tracks and the PCB. Enough talking! Let's take a simple, quick example.

From the idea to the board using KiCAD

Designing a PCB (whether it's your first or not) is always stressful. You must go about it logically and thoroughly check everything you do. In this example, for implementing a little microcontroller, there's no voltage converter, nor protective diode in case the supply polarity is reversed, nor a hard-to-solder micro-USB connector. We're going to content ourselves with the minimum

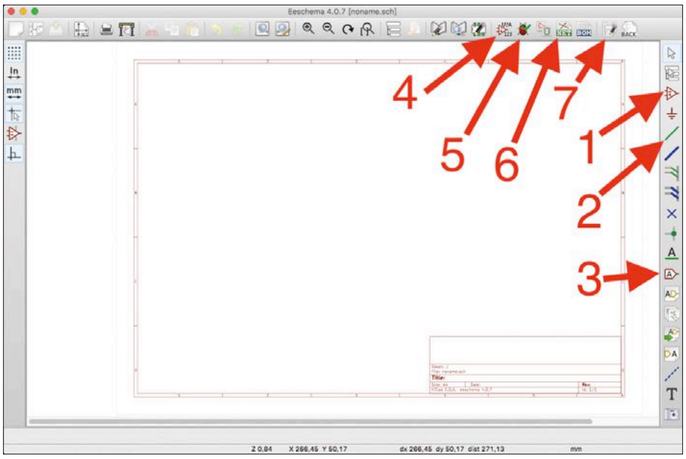


Figure 1. Essential tools in eeschema.

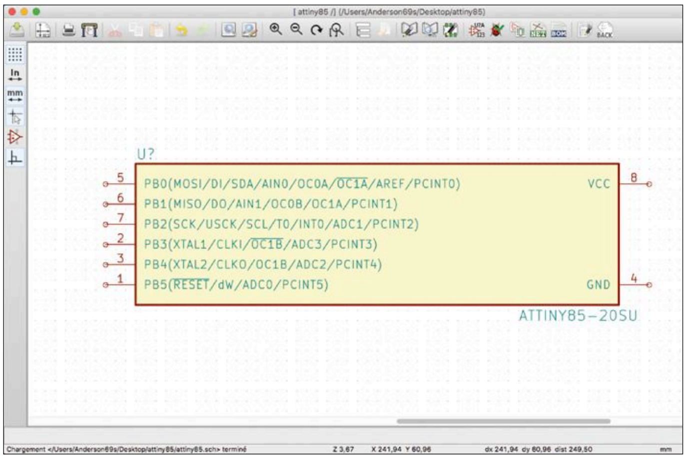


Figure 2. Placing the ATtiny85 microcontroller in eeschema.

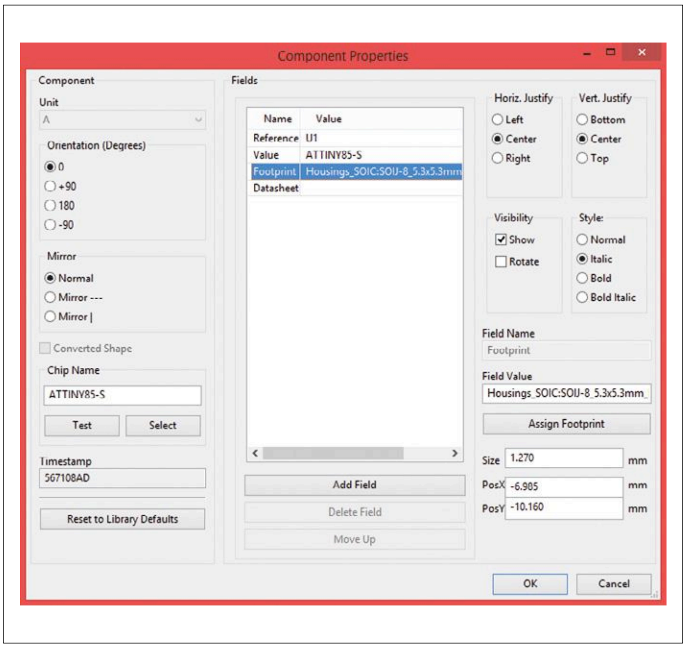


Figure 3. Component properties window in eeschema.

for a really simple PCB, even if it means taking a few risks:

- A 10-µF, 16-V capacitor, in a CMS 0805 package, for decoupling the microcontroller supply.
- Positions for 0.1-inch (2.54 mm) pitch pinheaders, to make it easier to connect cables for prototyping.
- An ATtiny85 8-bit microcontroller from AVR (recently bought out by Microchip [13]).

Although the 8-pin microcontroller operates at 3.3 V or 5 V and is available in a through-hole package we will be using the SMD version. It can be programmed using an Arduino Uno board and the Arduino IDE; it is ideal for projects that don't need much power and not many input/output pins. Note that the ATtiny85 datasheet [14] tells us that it can be clocked at 16 MHz using an external crystal, but we're going to content ourselves with the 8 MHz from the internal crystal.

Circuit step by step

Open KiCAD and create a new project (`File/New project`) named *attiny85*. KiCAD will propose automatically creating a folder with the same name, in which it will store all the files.

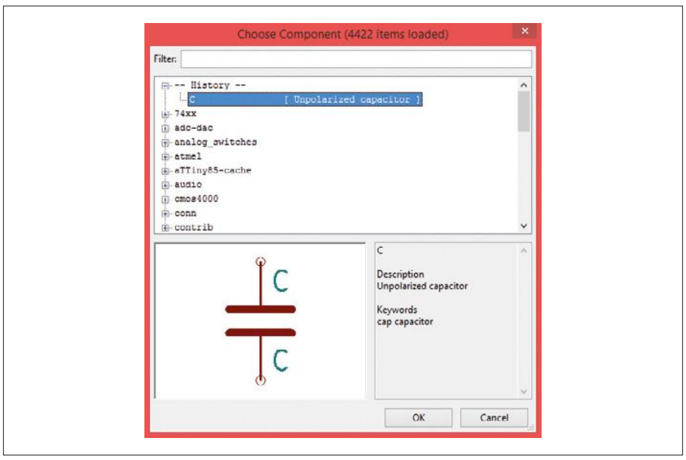


Figure 4. Selecting the capacitor in eeschema.

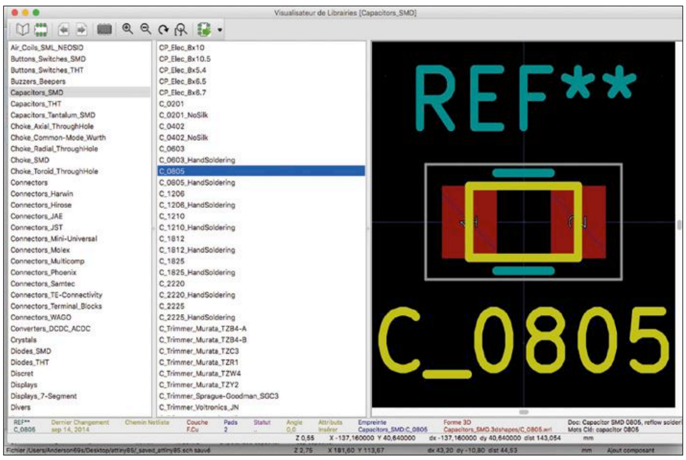


Figure 5. Choosing the capacitor footprint in eeschema.

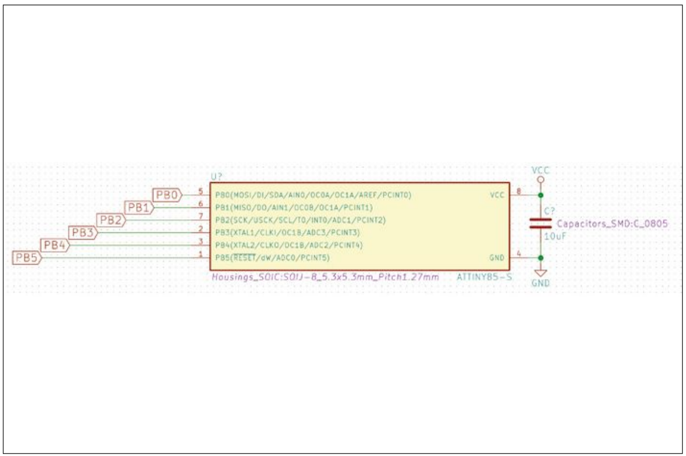


Figure 6. Partial circuit in eeschema.

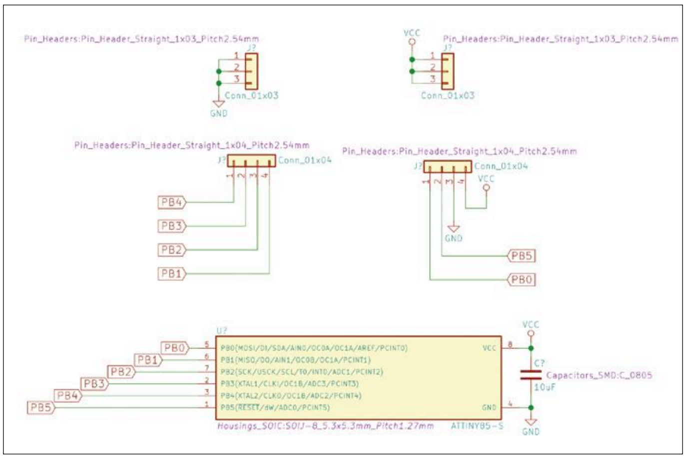


Figure 7. Final circuit with numbering in eeschema.

By default, there are three:

- attiny85.pro: all the project details
- attiny85.sch: circuit
- attiny85.kicad_pcb: design for the electronics board

Double-click on *attiny85.sch* in the left-hand column, eeschema will then open (**Figure 1**). Using `File/Page setting`, you can customize the red title block at the bottom right. Then add your first component using the `Place component` tool (Figure 4, pos. 1). The cursor changes to a pencil. Left-clicking on the page opens the component selection window. In the `Filter` field, enter *attiny85-20su*. Confirm using `OK` then place the microcontroller by left-clicking again on the page (**Figure 2**). Right-click on this one component, then select `Edit Component/Edit`; the `Component Properties` window will appear (**Figure 3**). This lets you, for example, adjust the orientation of the component and assign it a footprint.
The `Reference` field contains a unique letter+number combination for each component, it will be populated automatically later; you must never modify it manually.

As the ATtiny85 is by definition a microcontroller, the footprint is already assigned. The component reference and value are shown in blue-green at the bottom of the screen, while the footprint is written in purple. It is possible to hide or show these fields in the component properties.
Proceed in the same way to add the capacitor (`Filter → C`, **Figure 4**). You can see in its properties that the `Footprint` field is empty. So you need to `Assign Footprint`: click on the button with the same name. In the window that opens, double left-click to select the desired package or 'housing' in KiCAD terminology (**Figure 5**).

Also add the `VCC` and `GND` components, without assigning them footprints. Then use the `Place wire` tool (Figure 4, pos. 2) to create the first connections between the capacitor, VCC, GND, and the ATtiny85: left-click to start a wire and double-click to end it. Note the labels (`global label`, Figure 4, pos. 3) on the left-hand pins of the ATtiny85: these let you create an invisible wire between two pins on components that have the

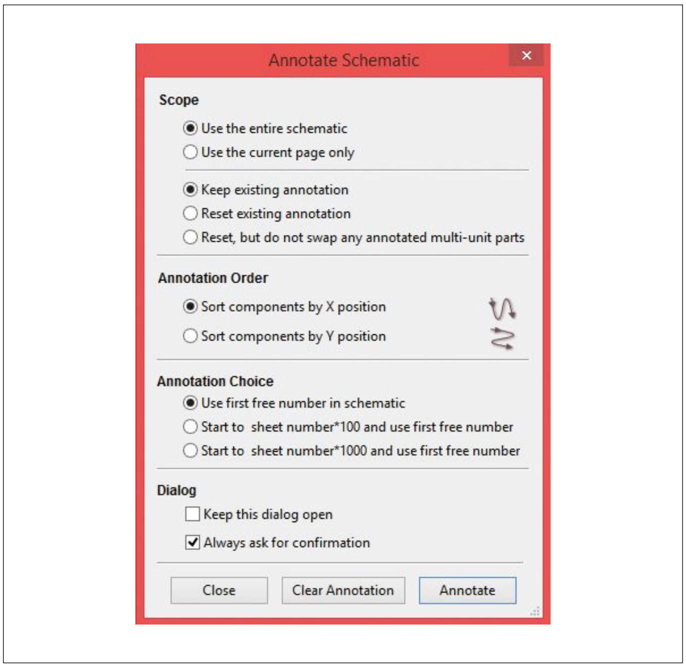


Figure 8. Annotating the circuit: option in eeschema.

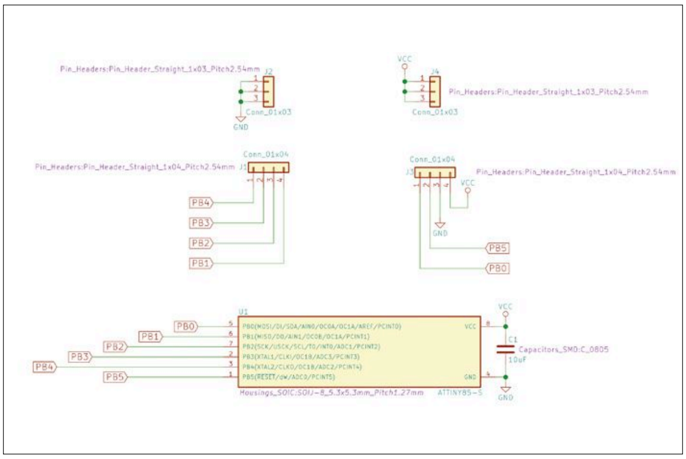


Figure 9. Final circuit after numbering in eeschema.

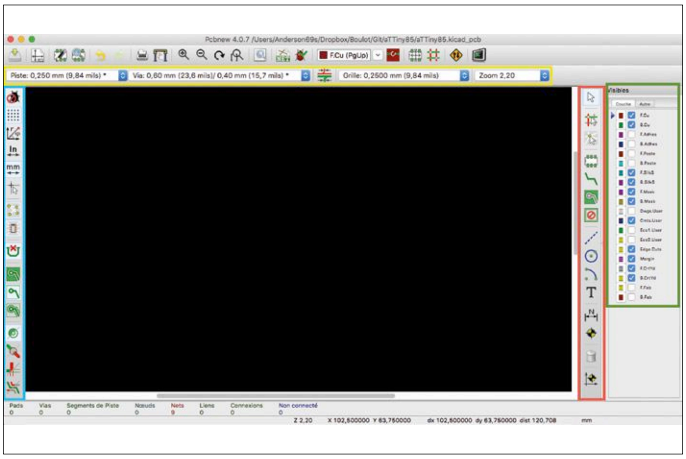


Figure 10. Organization of the Pcbnew interface.

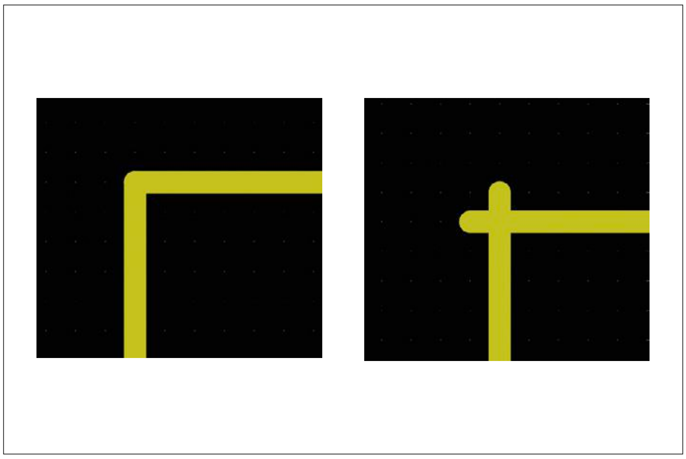


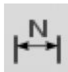
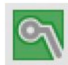
Figure 11. Good and bad layouts in Edge.Cut.

same global label. In this way, the circuit is easier to read and it limits the number of wires to be drawn in eeschema. But these tracks will still have to be drawn in Pcbnew.
Now place two 3-pin connectors (`Filter → conn_01x03`) for the GND and VCC signals, then two 4-pin connectors (`Filter → conn_01x04`) for the eight pins of the ATtiny85.
The connector footprints are free, let's start with easy-to-solder types (**Figure 7**):
`Pin_Headers:Pin_Header_Straight_1x03_Pitch2.54mm` and
`Pin_Headers:Pin_Header_Straight_1x04_Pitch2.54mm`
We still have to add a component reference number so as to get rid of the "?" (Figure 4, pos. 4). In the window that opens, set the options as in **Figure 8** and click on `Annotate`.
We then get the final circuit (**Figure 9**). As it is rudimentary, we'll dispense with the electrical rules test (Figure 4, pos. 5). In eeschema, all that remains to be done is to create the list of interconnections, the netlist (Figure 4, pos. 6) which will convey the essential routing information to Pcbnew. When the window opens, click on `Generate`. Now let's move onto to designing our board in Pcbnew (Figure 4, pos. 7).

PCB layout

While eeschema is easy enough to grasp, it's going to take a little longer to get used to Pcbnew. So what I'm offering here is just a quick tour to help you find your way around. We're not going to cover designing footprints, as the ones for the components in our example are already present in KiCAD. Attention: Thanks to its many options, Pcbnew is very comprehensive, and hence sometimes complex; we can't detail everything here. However, the documentation will be able to answer all your questions.
In **Figure 10**, the tools on the left (blue surround) let you modify the options for displaying the lands, tracks, or net. In the top strip (yellow surround), we find the track widths and hole sizes, along with the grid pitch. The right-hand column (red surround) groups together the tools needed for designing the circuit board. And lastly, right over on the right, the green frame lets you select the working layers. Just as in eeschema, right-clicking with the mouse is useful when it comes to using the various tools.

Table: Pcbnew tools

			
Add graphic line	Add dimension	Add zones	Add text

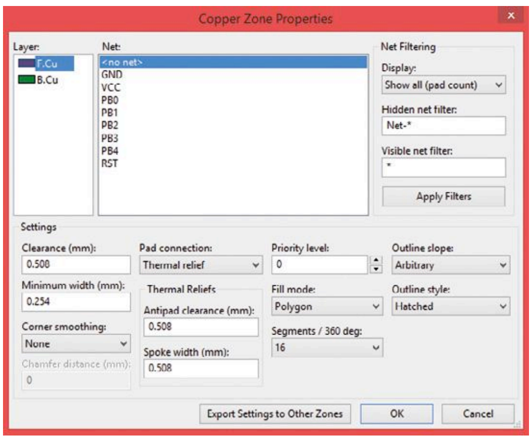


Figure 12. Copper area properties in Pcbnew.

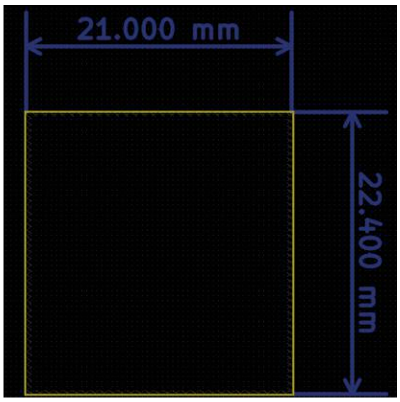


Figure 13. Edges, dimensions, and copper area in Pcbnew.

It's wise to start by setting up the layers: select Design Rules/ Layers Setup in the menu bar. In our case, Two layers, parts on front and back will suit us perfectly. Set the PCB thickness here (1.6 mm is a standard value). Confirm with OK. The Design Rule/Design Rules menu lets you assign a specific track width to each type of interconnection (net).
Now click on the Edge.Cuts layer (in the green surround) which is used to define the edges of the PCB. A little blue arrow will appear to the left of the name, indicating that this layer is active. Select the Add graphic line tool just alongside (see **table**). The Add graphic line tool in Pcbnew is used just like the one in eeschema. Watch out! You must be very precise if you don't want to see errors appearing in the 3D viewer (View/3D Viewer) (**Figure 11**).
It is possible to add dimensions so as to have a better idea of the size of the PCB. Position yourself on the Cmts.user layer and use the Add dimension tool (see table), which works in exactly the same way as the Add graphic line tool.

Now we're going to add the PCB's copper areas. Position yourself on the F.Cu layer and select the Add filled zones tool (see table). Select F.Cu and GND then confirm with OK. In this way, the whole copper layer F.Cu will be connected to ground (**Figure 12**). We'll pass over the other options available, as they are not needed in this example and can lead to some quite weird results if they are incorrectly defined. Draw a square that follows the edges on Edge.Cut. Do the same for B.Cu. You ought to get a result similar to the one in **Figure 13**.

It's time now to reveal the component footprints and position them on the PCB by reading the netlist. At the top of the screen, you should find the same NET button as in eeschema (Figure 4, pos. 6). Don't forget that the information is passed in one direction only: from eeschema to Pcbnew, never the reverse.

A window opens (**Figure 14**); as this is the first reading of the netlist in this example, the default options must be left. Attention: if you make any changes to your circuit in eeschema, the netlist must be created again. At that point, depending on the changes made, you must modify the read options in the netlist in Pcbnew in order for the modifications to be shown. Confirm using the Read Current Netlist button. A group of footprints is then shown (**Figure 15**).

The white lines representing the net indicate the tracks to be created in order to adhere to the circuit diagram; you can update this using the shortcut B, which at the same time verifies the electrical design rules and updates the ground planes. Move the footprints using your keyboard's M key (move) or a right-click in order to distribute them within the outlines already drawn. If Pcbnew can't determine automatically which footprint you want to move, it asks you to clarify your selection. It's also possible to orient the footprints using the shortcut R (rotate). The shortcut E (edit) lets you modify the footprint properties.

Let's add some text (see table) to describe the connectors. Position yourself on the F.SilkS layer which corresponds to the silk-screening, then use the Add text tool to write, for example. RST, PB0, etc. You obtain the result shown in **Figure 16**. It's time to create the tracks. Two solutions are available to you:

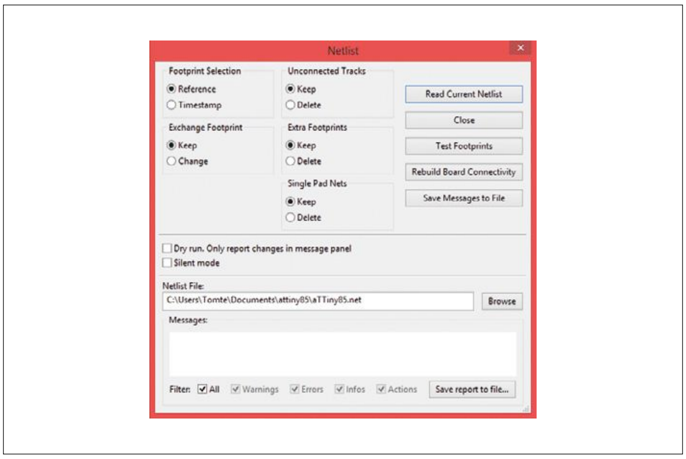


Figure 14. Reading the netlist in Pcbnew.

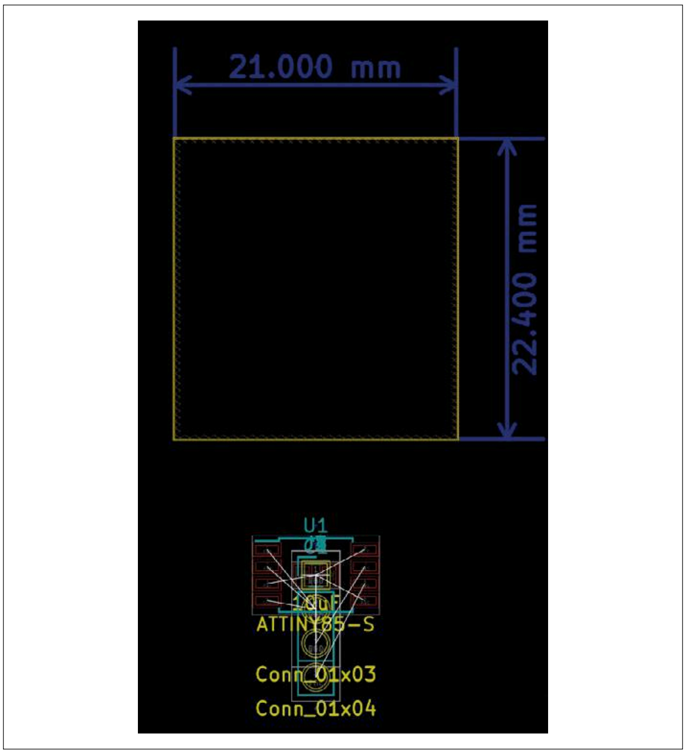


Figure 15. Appearance of footprints in Pcbnew.

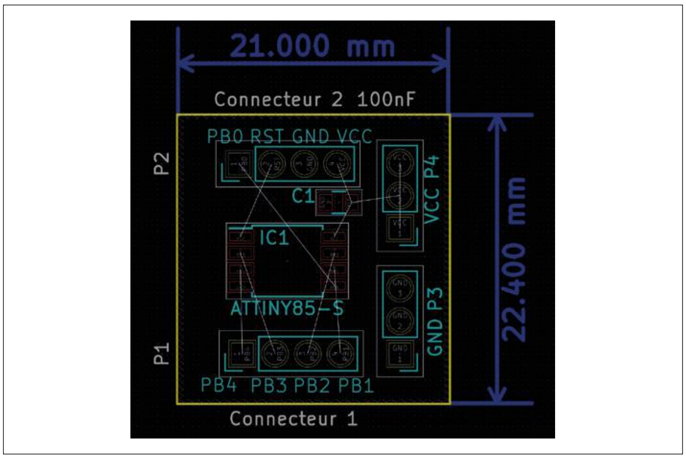


Figure 16. Components positioned prior to routing in Pcbnew.

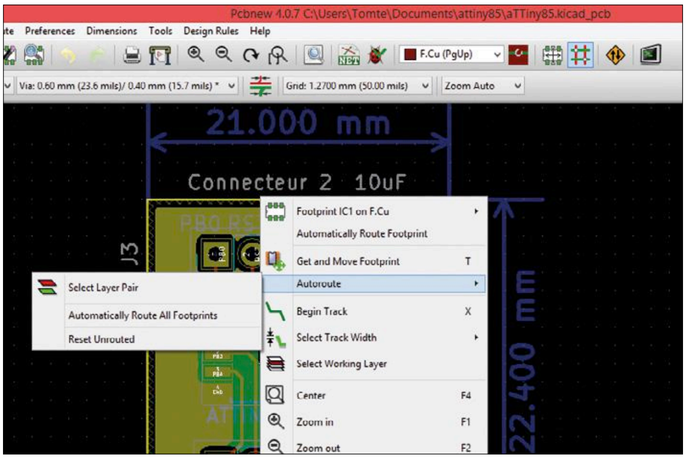


Figure 17. Launching automatic routing in Pcbnew.

- manual routing, which is time-consuming
- automatic routing: Pcbnew draws the tracks for you, according to the design in eeschema.

In either case, we won't need to route the `GND` net, as both our copper layers are of the `GND` type. In this way, Pcbnew on its own connects together lands with the same name; it also interprets the global labels added in eeschema and transforms them into tracks to be drawn.

However, the autorouter sometimes produces rather convoluted tracks that do not necessarily abide by the fundamental principles of design, namely: to avoid right-angles in the track layout as far as possible, or to keep as far away as possible from the board edges.

But to be honest, it's an easy choice to make between retouching a few tracks – or drawing them all by hand! For this example, these questions aren't very important, but on more complex boards, they can create irreparable errors, for example, due to electrostatic phenomena.

To start automatic routing, change the track mode to autorouting: right-click on the ATtiny and choose `Autoroute / Automatically Route All Footprints` (**Figure 17**). You will then see Pcbnew creating all the tracks between the components for you (**Figure 18**).

With a little bit of manual retouching, it is possible to obtain more pleasing layouts as in **Figure 19**. Lastly, you must create the Gerber files required for the board to be manufactured by

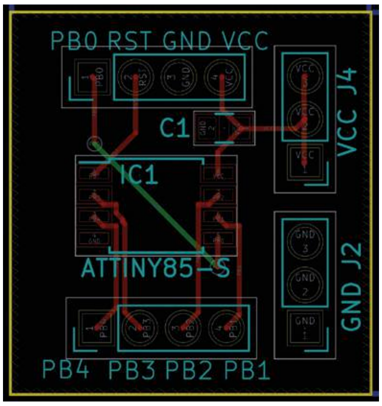


Figure 18. Autorouted board in Pcbnew.

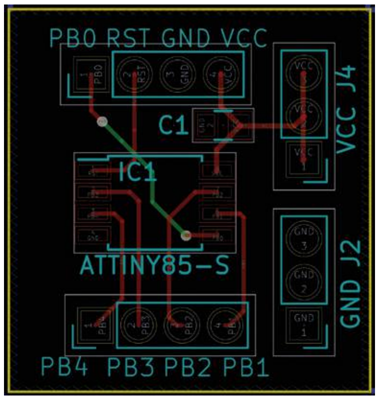


Figure 19. Autorouted and optimized board in Pcbnew.

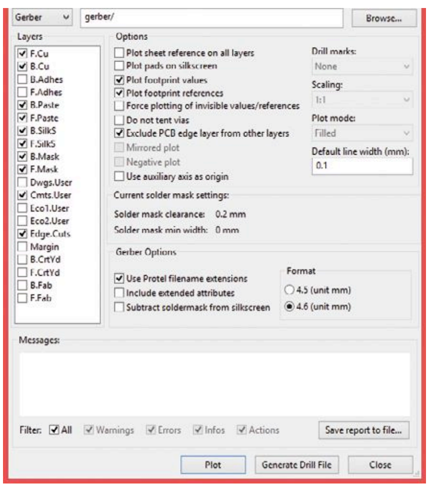


Figure 20. Window for creating Gerber files in Pcbnew.

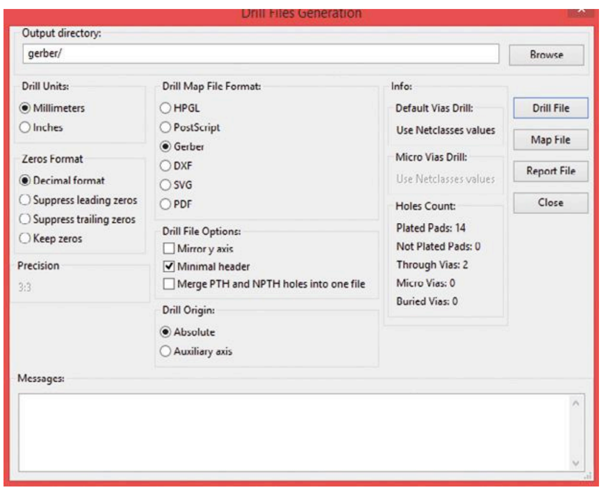


Figure 21. Window for creating drill files in Pcbnew.

your chosen professional (SeeedStudio [15], EasyEDA [16], Eurocircuits [17], etc.). Click on `File/Plot` (Figure 20). Make sure you also `Generate drill file` (Figure 21). And lastly, combine all the files in the project's Gerber folder into a single ZIP file. Of course, you must ensure that all the details are correctly interpreted by the service provider, using the site's built-in Gerber reader. After placing your order, receiving your parcel, and fitting the components, the completed board ought to look like the one shown in Pcbnew's built-in 3D viewer (Figure 22). You'll soon spot that in the photo of the actual board, there are no capacitors, the VCC and GND connectors are reversed, and the track layout is different. This board was my first PCB design, over three years ago, and I made some mistakes, whence these differences… But in electronics, we always learn from our mistakes and do better next time...

A Github repository [18] gives you access to the source files for the new version and explains how to send code to the ATtiny85 using an Arduino UNO board and the Arduino IDE.
You now know the main steps for designing a board using KiCAD.

(160171-I)

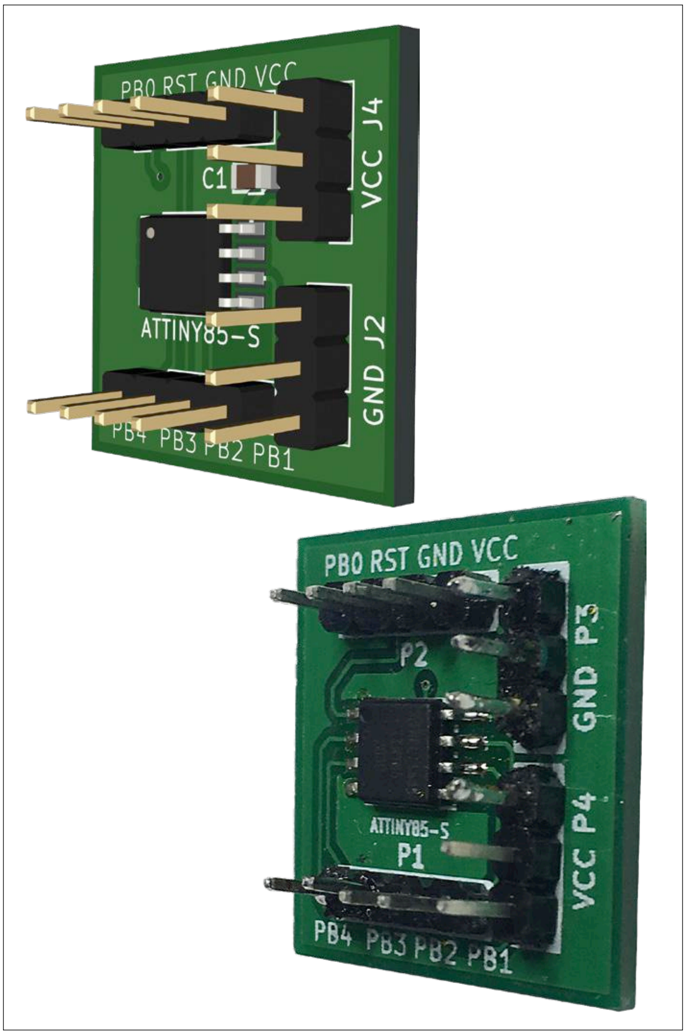


Figure 22. Comparison of the 3D preview and the actual board.

Author's biography

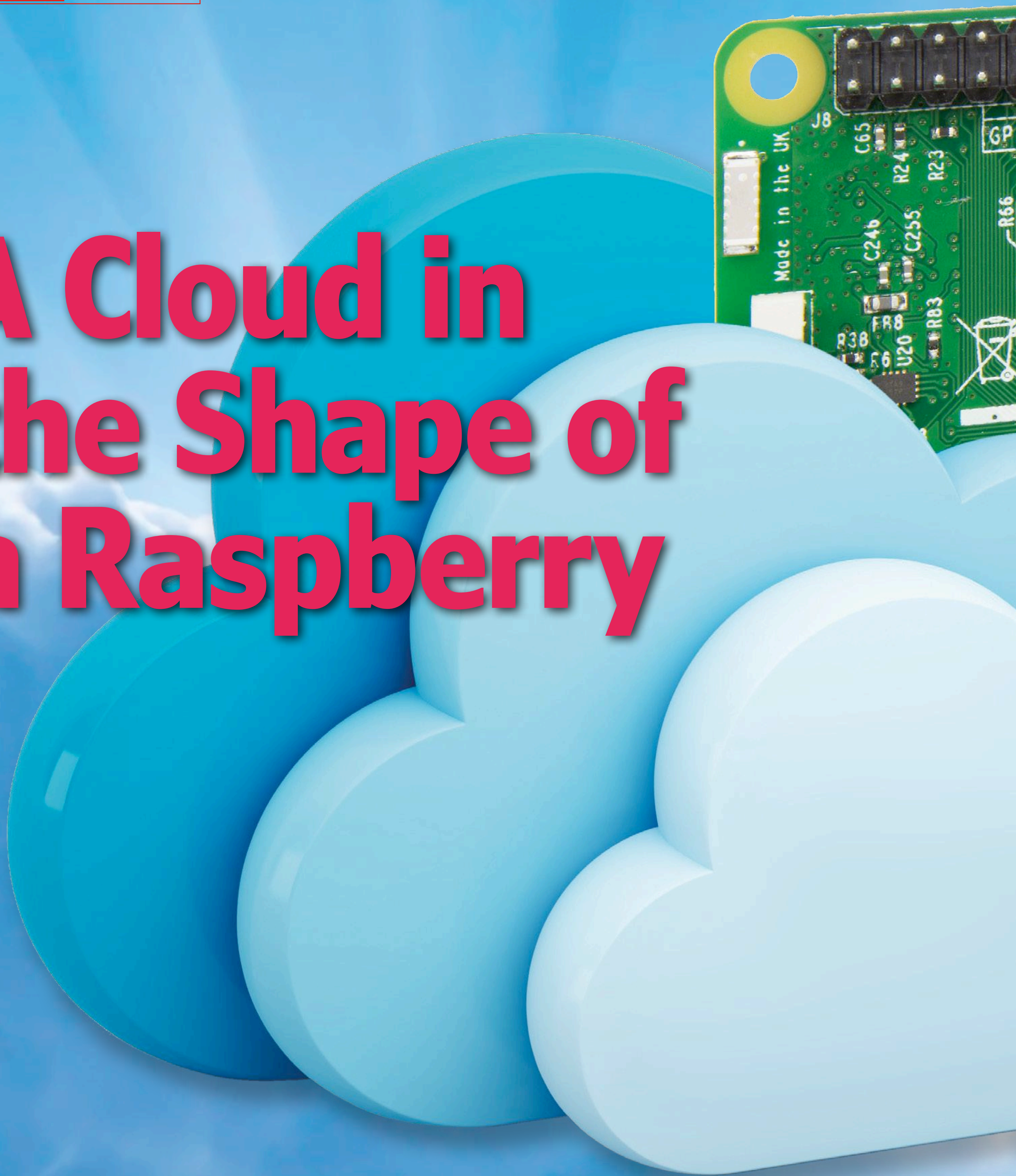
Alexandre Perier-Muzet trained as a chemist, and became an electronics engineer through his passion. He's an ardent supporter of royalty-free software solutions. He shares his creations and projects on his blog: anderson69s.com.

Web Links

[1] KiCAD: http://kicad-pcb.org
[2] Eagle: www.autodesk.com/products/eagle/overview
[3] Altium: www.altium.com
[4] SolidWorks PCB: www.cadvision.fr/logiciel-cao/logiciel-ecad-solidworks-pcb
[5] PADS: www.pads.com
[6] OrCAD PCB: www.orcad.com/products/orcad-pcb-designer/overview
[7] Pulsonix: www.pulsonix.com
[8] Europlacer: www.europlacer.fr
[9] Technalp: http://technalp-electronique.fr
[10] Official KiCAD forum: https://forum.kicad.info
[11] SnapEDA: www.snapeda.com/home/
[12] Donations towards the development of KiCAD via CERN: https://giving.web.cern.ch/civicrm/contribute/transact?id=6
[13] MicroChip: www.microchip.com
[14] Datasheet for ATtiny85 from MicroChip: www.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf
[15] SeeedStudio: www.seeedstudio.com
[16] EasyEDA: https://easyeda.com
[17] Eurocircuits: www.eurocircuits.com
[18] Github repository for the project: github.com/Anderson69s/attiny85

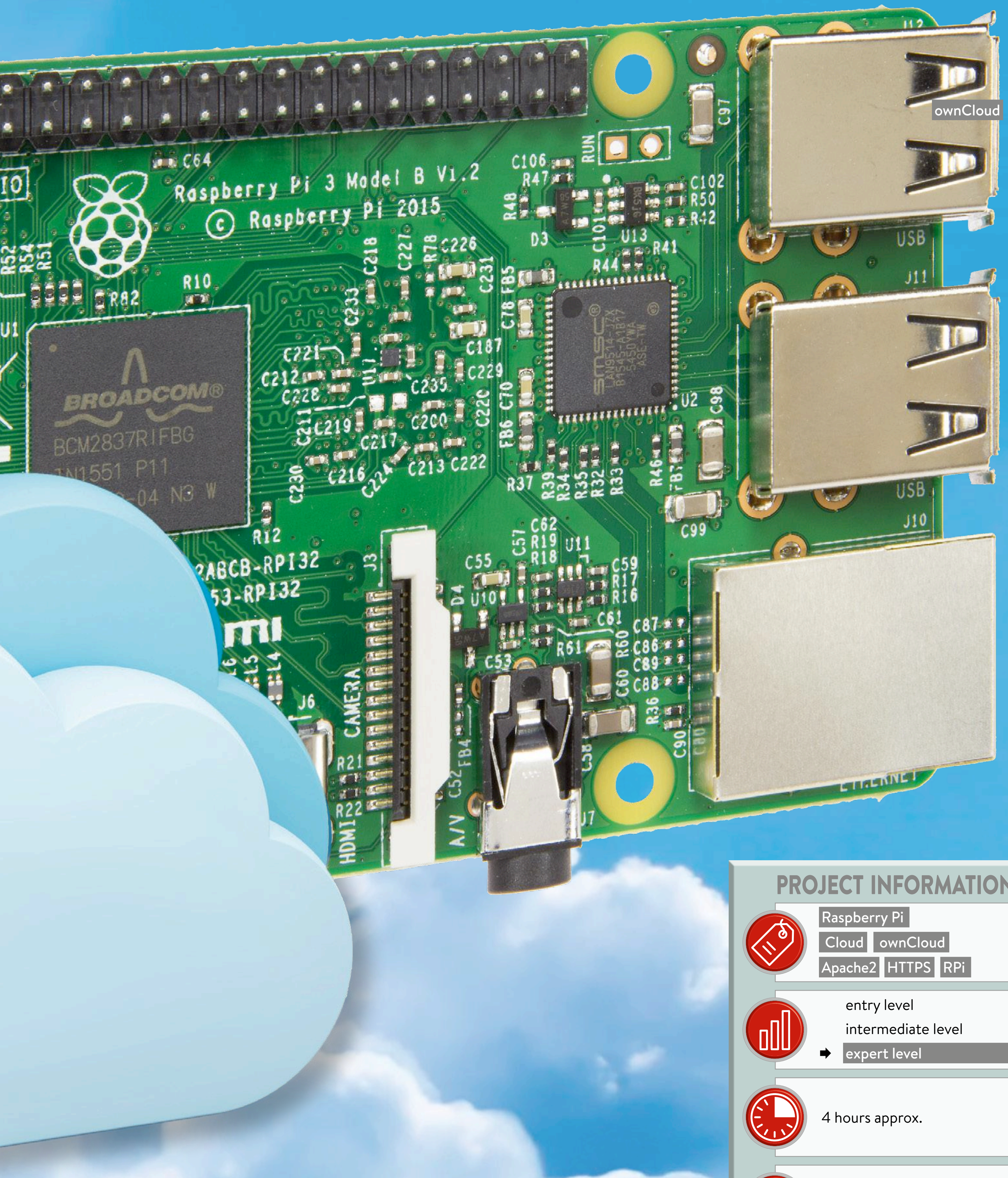
e LABS PROJECT

A Cloud in the Shape of a Raspberry



install a cloud in your living room

By **Sébastien Guerreiro de Brito** (France) ownCloud



PROJECT INFORMATION

Raspberry Pi · Cloud · ownCloud · Apache2 · HTTPS · RPi

entry level
intermediate level
➡ **expert level**

4 hours approx.

Computer,
SMD soldering iron

€150 / £135 /$185

Today the cloud is a popular solution to save and share files as well as allow access to your files from any computer anywhere in the world. Because many people are wary of the reliability and security of cloud storage, this article shows you how to create your personal cloud and its power supply board.

Characteristics

- Accessible from anywhere in the world
- Share your files, photos and applications
- Use the cloud

How do you create a Cloud?

A cloud is made partly of hardware and partly of software. On the hardware side, I chose the Raspberry Pi platform equipped with a hard disk to be able to store a large number of files. Because hard disks consume a lot of power, I decided to design a power supply capable of correctly powering the Raspberry Pi – hard disk assembly (see the inset and **Figures 1 and 2**).

For the software part, I chose the solution 'ownCloud'. Offered as an alternative to services like DropBox, ownCloud is a free software to create file storage and distribution service platforms. It offers the following characteristics:

- it's free, with open source code and no limitations;
- administration is done via a web page;
- the administrator can manage the users by their names or their group;
- sending an internet link is sufficient to allow sharing;
- files are encrypted during transfer;
- there is a client to synchronise selected local files under Windows or Linux.
- ...

It's really pretty good and doesn't seem to differ that much from other well-known free systems. On the other hand, it has numerous advantages:

- you know exactly where your files are stored;
- the amount of storage may be substantial, even bigger than existing solutions;
- you may shut down the system whenever you wish.
- Etc.

Preparing the Raspberry Pi

To start, download the Raspbian image on the official Raspberry Pi site [1] and then install it. For this article, we used `2017-11-29-raspbian-stretch-lite`. The 'Lite' version of the image is sufficient as we are not using a screen with graphical interface. Follow the installation instructions from the official site, or refer to any of the numerous tutorials available on the internet.

Once the image is burned to the microSD card (which is easy with 'Etcher', https://etcher.io/), place it in the reader slot of the Raspberry Pi. Connect a screen and keyboard, then the power.

Log in to the Raspberry Pi. I should remind you that the default login is 'pi" and the default password is 'raspberry'. The start the configuration of the board by typing the command:

```
sudo raspi-config
```

From the configuration menu, set the keyboard according to your language (localisation). I would advise to activate the SSH service. This will be useful for remote interventions on the board. For

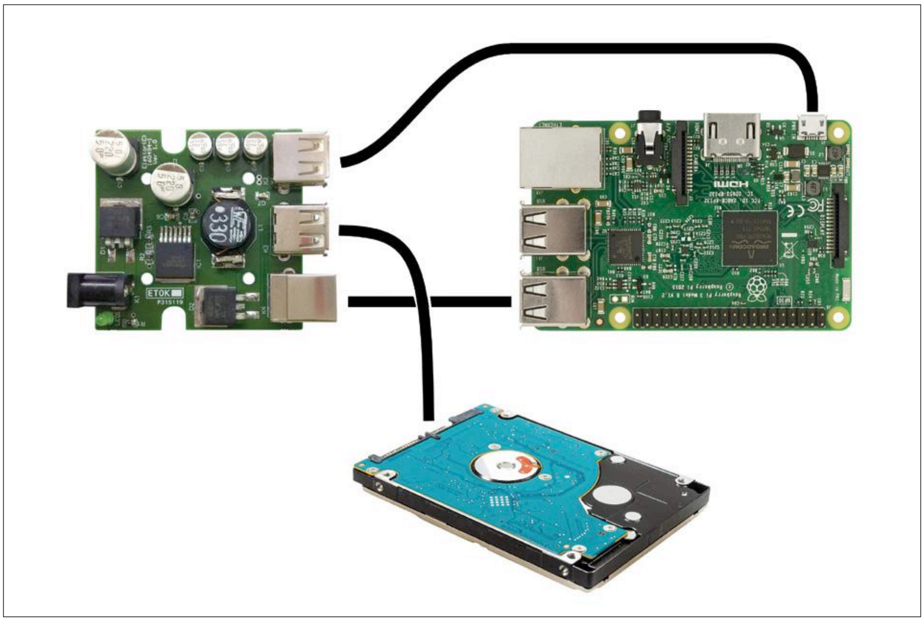


Figure 1. Here's how to connect a SATA hard disk powered only by USB.

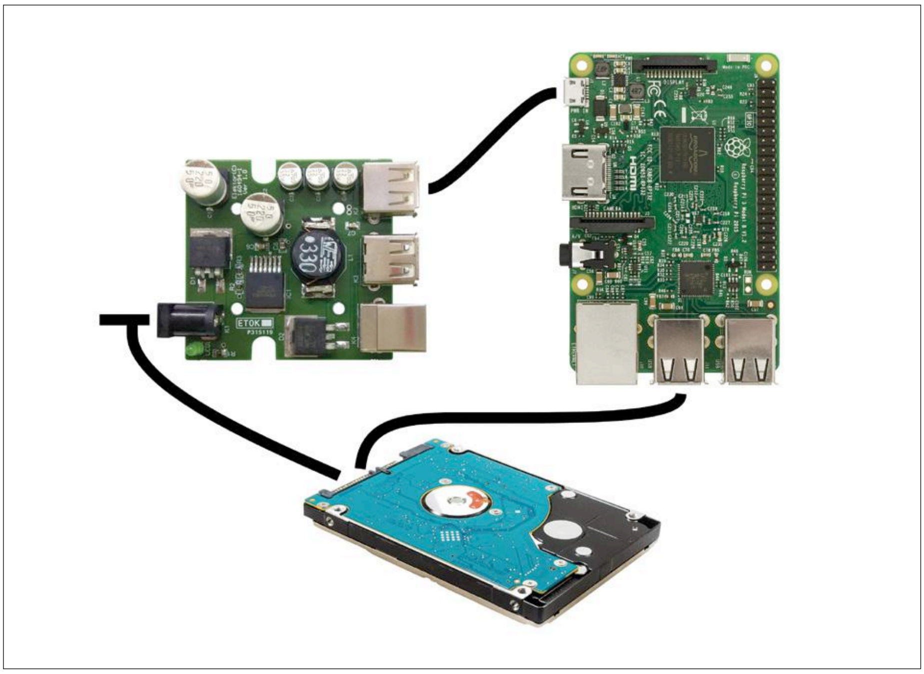


Figure 2. An IDE hard disk for which the USB Converter has its own power can be connected as well.

security reasons, take this opportunity to change the password.
It's also a good idea to give a fixed IP address to the Raspberry Pi:

```
sudo nano /etc/network/interfaces
```

Next, add the lines:

```
auto eth0
iface eth0 inet static
address 192.168.yyy.xxx  <=
   Desired IP address
netmask 255.255.255.0
gateway 192.168.yyy.zzz  <= IP
   address of your router
```

Press Ctrl-O then enter to save the file, Ctrl-X to quit the editor. You could instead, if preferred, configure your router to always provide the same address to the IP au Raspberry Pi.

Configuration of the storage hard disk

Let's start by creating the folder on which the hard disk will be mounted:

```
sudo mkdir /mnt/usb
```

Then change the rights of this folder and mount the disk which, in our case, is called sda1. Its file system is `ext4`:

```
sudo chmod -R a+w /mnt/usb
sudo mount /dev/sda1 /mnt/usb/
```

Now we pass to the automatic mounting of the external hard disk on startup of the Raspberry Pi. For this we have to find out the PARTUUID of the disk:

```
sudo blkid
```

Look for your hard disk in the list and note its PARTUUID, then open the file `fstab`:

```
sudo nano /etc/fstab
```

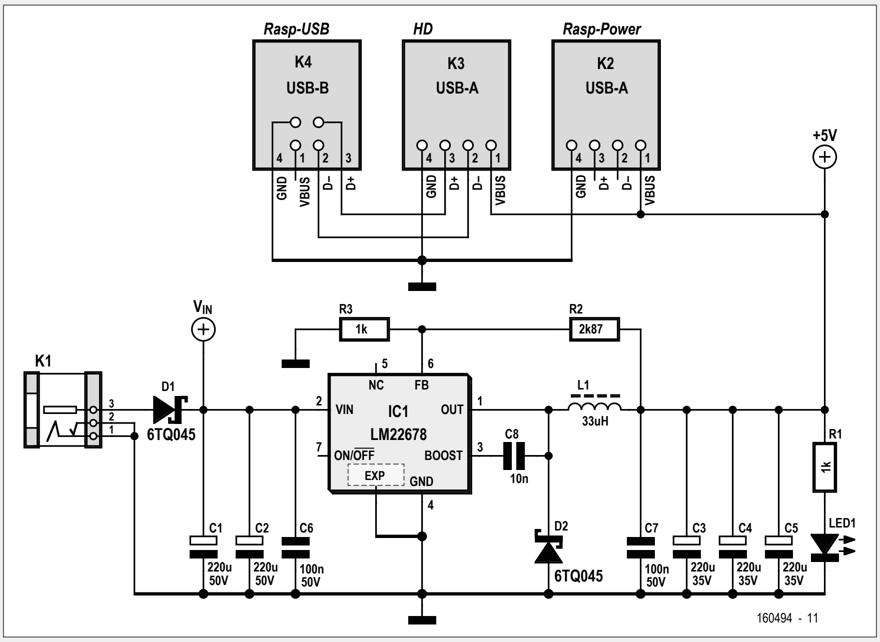
Add this line to the end of the file:

```
PARTUUID=6f20736b-01 /mnt/usb ext4
   defaults 0 2
```

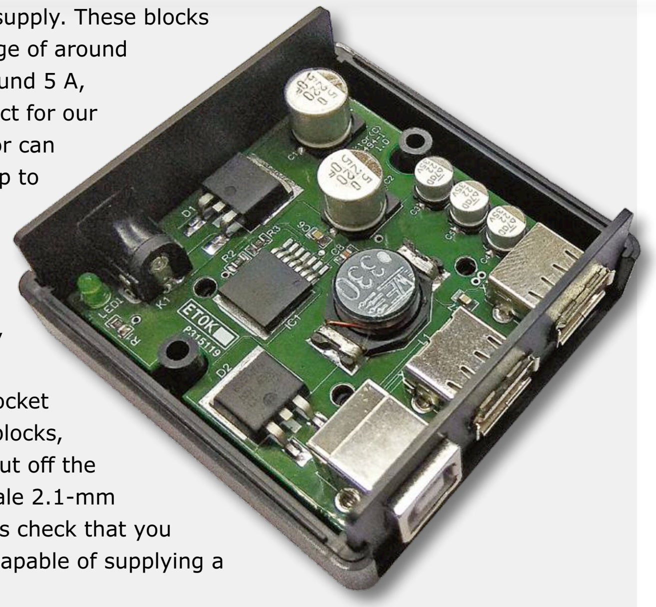
Here '6f20736b-01' corresponds to my hard disk, replace this value with the identifier of your hard disk. Press Ctrl-O then Enter to save the file, Ctrl-X to quit the editor. Take care not to make

The power supply board

The circuit diagram of the power supply is based on an LM22678 switching regulator. It is capable of supplying a current of 5 A maximum. The more hungry SATA hard disks can consume up to 2 A at 5 V. If you consider that the Raspberry Pi nano-computer consumes a maximum of 1 A at 5 V, our regulator has definitely got what it takes.



To power our cloud, the idea is to use an old 'brick' laptop computer power supply. These blocks generally deliver a voltage of around 19 V and can supply around 5 A, which makes them perfect for our application. Our regulator can support input voltages up to 42 V, so this is ideal.
To be compatible with the maximum number of laptop power supply blocks, our power supply card is equipped with a 2.1-mm DC connector socket (barrel jack). For some blocks, it may be necessary to cut off the original plug and fit a male 2.1-mm DC connector. But always check that you choose a laptop supply capable of supplying a current of 5 A.
Diode D1 protects the supply against wrong polarity of the input voltage.



Regulator IC1 exists in two versions of output voltage: adjustable or fixed at 5 V. I used the adjustable version, because that's what I had on hand :-). The output voltage is set by the ratio of the values of the resistors R2 and R3.

$$V_{out} = 1.285 \times (R2/R3 + 1)$$

With R2 = 2.87 kΩ and R1 = 1 kΩ, the value of V_{out} is 4.97 V. If a fixed 5-V regulator is used, don't fit R3 and make R2 = 0 Ω (or a link).
The two large electrolytic capacitors at the input serve to provide the extra current needed by the SATA hard disk at startup.

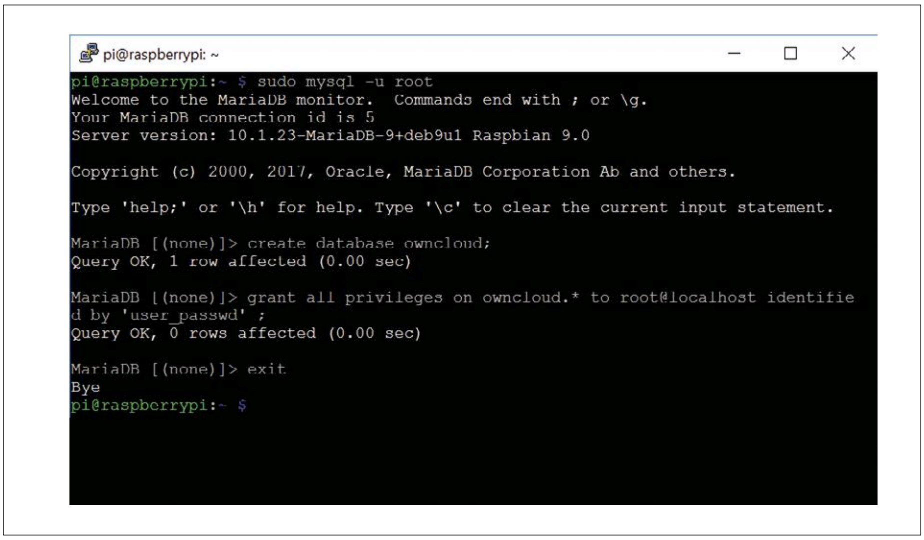


Figure 3. Preparation of the data base for user 'root' with password 'user_passwd'.

any typos, because the results can be disastrous, you would be obliged to reinstall the image on the SD card, etc. Restart the Raspberry Pi with the command:

```
sudo reboot
```

If all goes well, you can read (after connection, of course) the contents of `/mnt/usb` with the command:

```
ls /mnt/usb
```

Install ownCloud

The ownCloud application needs PHP version 5.6 or more recent and the web server Apache2. So we install Apache2, MySQL, PHP 7.0 plus a host of supplementary modules:

```
sudo -i
```

Now we are in 'Super User' mode which relieves us of the obligation to precede every command with 'sudo'.

```
apt-get update
```

The lines below constitute a single command:

```
apt install -y apache2 mariadb-
   server libapache2-mod-php7.0 \
    php7.0-gd php7.0-json php7.0-
   mysql php7.0-curl \
    php7.0-intl php7.0-mcrypt php-
   imagick \
    php7.0-zip php7.0-xml
   php7.0-mbstring
```

Next we download the ownCloud package, we unpack it and copy it into our directory. Note, the version number of the package may be different from that used here (10.0.4), change it if necessary:

```
wget https://download.owncloud.
   org/community/owncloud-
   10.0.4.tar.bz2
tar -xjf owncloud-10.0.4.tar.bz2
sudo cp -r owncloud /mnt/usb
```

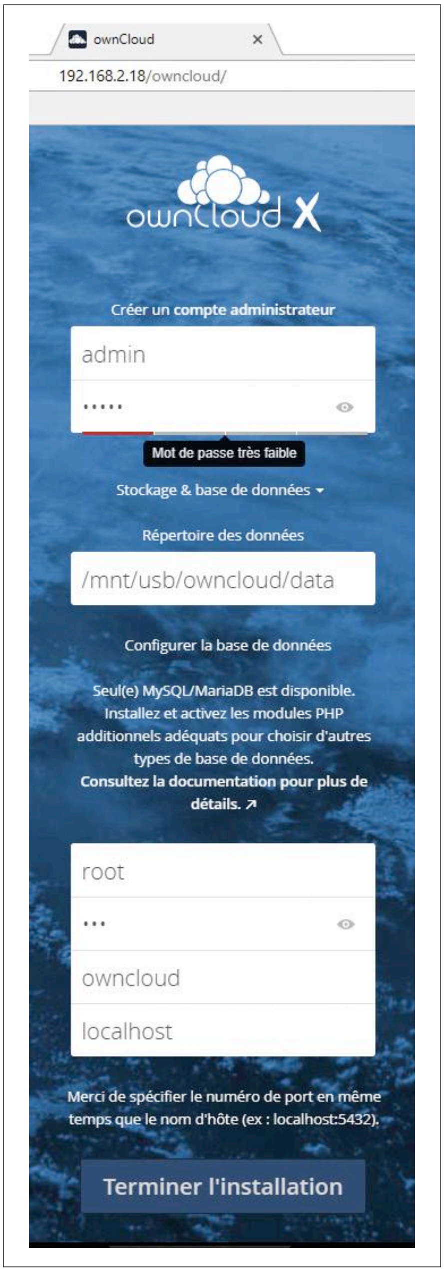


Figure 4. Successful installation, you now have access to your cloud.

Create a database

For ownCloud to function, it is necessary to create a database:

```
mysql -u root
  create database owncloud;
  grant all privileges on
   owncloud.* to <user>@localhost
   identified by '<user_passwd>';
  exit;
```

The name `<user>` is the name we will give to the database user and `<user_passwd>` will be the password (**fig. 3**).

Configure the web server

Apache2 needs to know some details on the location of ownCloud. We place this information in a configuration file:

```
nano /etc/apache2/conf-available/
   owncloud.conf
```

To continue we enter:

```
Alias /owncloud "/mnt/usb/
   owncloud/"
<Directory /mnt/usb/owncloud/>
Options +FollowSymlinks
AllowOverride All
<IfModule mod_dav.c>
Dav off
</IfModule>
<IfVersion < 2.3>
order allow,deny
```

```
allow from all
</IfVersion>
<IfVersion >= 2.3>
Require all granted
</IfVersion>
SetEnv HOME /mnt/usb/owncloud
SetEnv HTTP_HOME /mnt/usb/owncloud
</Directory>
```

Press Ctrl-O then Enter to save the file, Ctrl-X to quit the editor. We finish the configuration with some supplementary commands:

```
ln -s /etc/apache2/conf-available/
   owncloud.conf /etc/apache2/
   conf-enabled/owncloud.conf
a2enmod rewrite
a2enmod headers
service apache2 restart
chmod a+w /mnt/usb/owncloud/config
```

First connection to ownCloud

To grant the Raspberry Pi web server (Apache2) access to the cloud files, we have to modify the rights. The default user and group for Apache2 are 'www-data'. So we execute the commands:

```
chown www-data:www-data -R /mnt/
   usb/owncloud
chmod 770 -R /mnt/usb/owncloud
```

Using your preferred browser, go to this address (replace « yyy.xxx » with the corresponding values for your network):

```
http://192.168.yyy.xxx/owncloud
```

You will then be invited to create an administrator account to manage the application (**Figure 4)**. Enter the path where the data are stored (`/mnt/usb/owncloud/data` in our example) then select the type of database installed (mysql). To finish, enter the username that you have chosen (<user>), the password (<user_passwd>), the name of the database ('owncloud' in our case) and finally, for the address of the database, use 'localhost'.
And there you go! You can now use your own cloud (**Figure 5**), either from your browser, or using an ownCloud client which you can download from ownCloud's website.

Connect via HTTPS

When you first connect, you will notice that ownCloud recommends a secure connection using HTTPS. To reconfigure our system, enter the following commands:

```
sudo -i
a2enmod ssl
service apache2 restart
a2ensite default-ssl.conf
service apache2 restart
```

The SSL module needs some keys, let's create them:

```
cd /etc/apache2
mkdir certOC
cd certOC
openssl genrsa -out owncloud.key
   1024
openssl req -new -key owncloud.key
   -out owncloud.csr
```

Worldwide access?

To access your cloud from the computer in your office — or from anywhere on the planet — you will ideally need a dynamic DNS address. For this I advise you to use the service 'NO-IP' [3]. Get onto the site and register your desired domain name. It's free but the only downside is that every month you receive a message to continue to use your account.
Installation is simple enough. From your Raspberry Pi, execute the commands below in a working directory of your choice:

```
sudo wget https://www.noip.com/client/linux/noip-
   duc-linux.tar.gz
tar xf noip-duc-linux.tar.gz
```

Verify the version of noip (2.1.9-1 in our case) and modify the following command if necessary:

```
cd noip-2.1.9-1
```

Before starting installation, make sure you know your login and password for NO-IP:

```
sudo make install
```

Now we need a script to launch the service at startup. Download the script 'noip' at [4] and then copy it into `/etc/init.d/`. To get the 'noip' to launch at startup you must type the following lines:

```
sudo chmod 755 /etc/init.d/noip
sudo update-rc.d noip defaults
sudo service noip configure
```

Start the service manually:

```
sudo service noip start
```

You can verify the status of the service with the command:

```
sudo service noip status
```

Finally, to be able to use OwnCloud from the outside, you need to modify a parameter in the file `/mnt/usb/owncloud/config/config.php`:

```
'trusted_domains' =>
  array (
    0 => '192.168.yyy.xxx',
    1 => 'votre_nom_domaine',
  ),
```

On your local network, you should use your internal address (for example:
https://192.168.yyy.xxx/owncloud)
to set up your synchronisation client.
After having configured your box (see [3]), you can now share your files with the whole world!

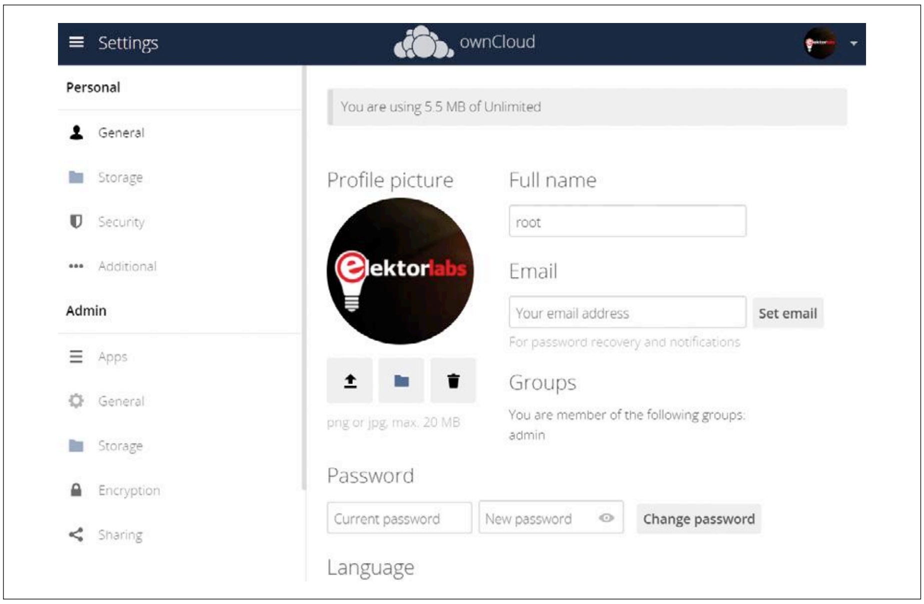


Figure 5. That's it, here we are with the dashboard of our cloud. Now it's up to you to manage your cloud.

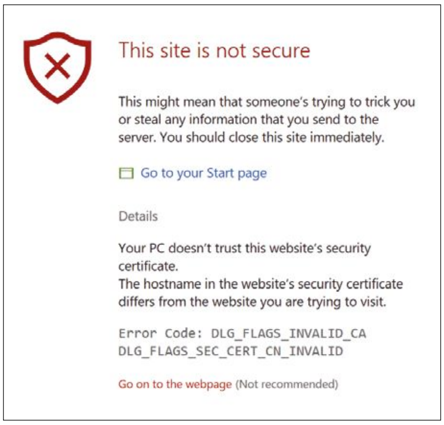


Figure 6. Don't panic! You can have confidence in yourself and your own systems, right?

You will have to respond to a certain number of questions to complete the certificate — be sure to note your responses somewhere just in case (or accept the default values).
Then:

```
openssl x509 -req -days 365 -in
   owncloud.csr -signkey owncloud.
   key -out owncloud.crt
```

We copy the keys:

```
cp owncloud.crt /etc/ssl/certs
cp owncloud.key /etc/ssl/private
```

A second ownCloud configuration is needed:

```
nano /etc/apache2/sites-available/
   owncloud.conf
```

with contents as follows:

```
<VirtualHost *:443>
DocumentRoot /var/www
SSLEngine On
SSLOptions +FakeBasicAuth
   +ExportCertData +StrictRequire
SSLCertificateFile /etc/ssl/certs/
   owncloud.crt
SSLCertificateKeyFile /etc/ssl/
   private/owncloud.key
</VirtualHost>
```

COMPONENT LIST

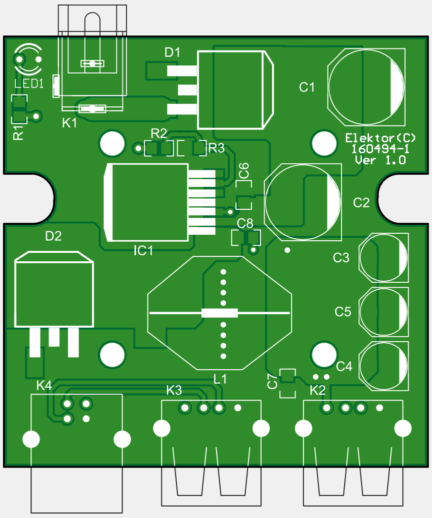
Resistors
All 0805, 0.1 W
R1,R3 = 1kΩ 1%
R2 = 2.87kΩ 1%

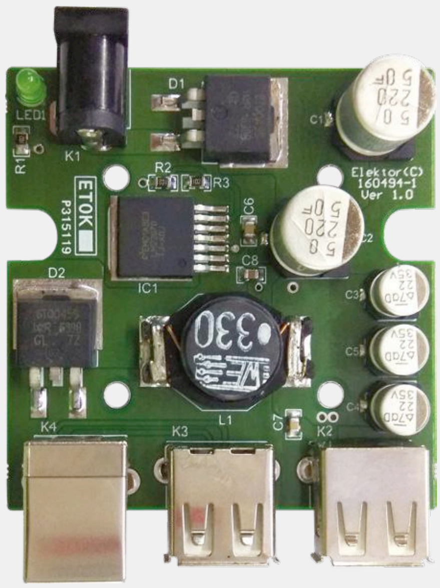
Capacitors
C1,C2 = 220µF, 50V, SMD, 10mm diam.
C3,C4,C5 = 22µF, 35V, SMD, 5mm diam.
C6,C7 = 100nF, 0805
C8 = 10nF, 0805

Inductor
L1 = 33µH, 3.4A, 70mΩ, e.g. Würth 74457133

Semiconductors
D1,D2 = VS-6TQ045SPBF
IC1 = LM22678TJ
LED1 = green, 3mm

Miscellaneous
K1 = DC barrel socket, 5A, 2.1mm diam.
K2,K3 = USB connector type A
K4 = USB connector type B
Hammond case 1593KBK





PCB # 160494-1

Press Ctrl-O then Enter to save the file, Ctrl-X to quit the editor.

Finally, we activate the new configuration and restart the Apache2 web server:

```
a2ensite owncloud.conf
apachectl configtest
service apache2 restart
```

You can now connect to your cloud using the address http**s**://192.168.yyy.xxx/owncloud (with an 's' on the HTTP). Your browser will warn you of a potential danger (**Figure 6**) because the signature that we've just created is not certified. You can ignore this error (or add an exception) and carry on. This will not in any case affect the function of your cloud.

What now?

You are now in possession of your very own cloud, of a good size, over which you have total control. You can choose to switch it on or off whenever you want, launch an automatic backup of your cloud, etc. It's you who is in command, it's you who is piloting the cloud.

(160494)

Web Links

[1] Raspbian: www.raspberrypi.org

[2] ownCloud: owncloud.org

[3] Service NO-IP: www.noip.com

[4] Page for this article: www.elektormagazine.com/160494

FROM THE STORE

→160494-1
Power supply printed circuit board, bare

Tips and Tricks

From readers for readers

Another neat solution to a tricky problem.

Temperature Logger hack

By **Dr. Thomas Scherer** (Germany)

*Which is the most reliable brand of car? Experts and laymen all over the world will have an opinion on which make of car they swear by. By all accounts Toyota is up there in the top three or four most reliable brands of car. A recent experience has led me to question if their reputation is justified. One morning my Prius (***Figure 1***), which is barely five years old, just refused to start. Press the start button… nothing, the dash lights just flicker dimly. Any electronics engineer will of course, always have a backup power source at hand to quickly breathe life into a failing battery and get things going, now I just needed find the battery location in the car. It's a bit surprising that a hybrid car like the Prius still needs a standard 12 V battery to get going just like any conventional car. It supplies power for lighting and the on-board electronics. Despite the 1.3 kWh HV powertrain batteries stashed under the seats nothing works without this 12 V battery, as I found out! The 12 V battery is situated in the corner of the trunk, now the kicker… without 12 V I can't get the tailgate latch open!*

A solution?

Feeling rising frustration, I take a deep breath and sit down in front of the PC hoping that Google will help me out here. Solution: in the fuse box at the front of the engine compartment, there's a copper strip with a 'direct line' to the positive pole of the battery; seems like Toyota has already anticipated my predicament. I hooked up a small 12-V 4.5-Ah AGM battery and the car was ready to go — the energy needed to crank the starter actually comes from the HV battery, the 12-V battery doesn't need to supply the hundreds of ampères starter current, it only supplies a short 15-A burst of energy. This is used to pressurize the braking system and then briefly a few amps are taken by the hybrid control unit before it switches over to HV battery power via a switched-mode power supply and from where the 12-V battery is charged. With the car running sweetly I took the small 12-V AGM battery and a couple of leads along for the ride… you can never be too sure. The story didn't end there, one week later the Prius did the same thing again. This was starting to get annoying, the 12-V battery in a Prius really doesn't undergo much stress in its life, also it's an AGM type which should last much longer — 12 years is not uncommon compared to 5 to 7 years for a standard lead-acid battery. Maybe I hadn't closed all the doors properly; Maybe an interloper has been coming in at night and reading books by the interior light or maybe all those reviews I'd read praising its reliability were, after all just 'fake news'. I ordered a new battery (from Varta) and installed it. The old one was now definitely dead, although it really shouldn't have been. All went well until barely a month later when, guess what?

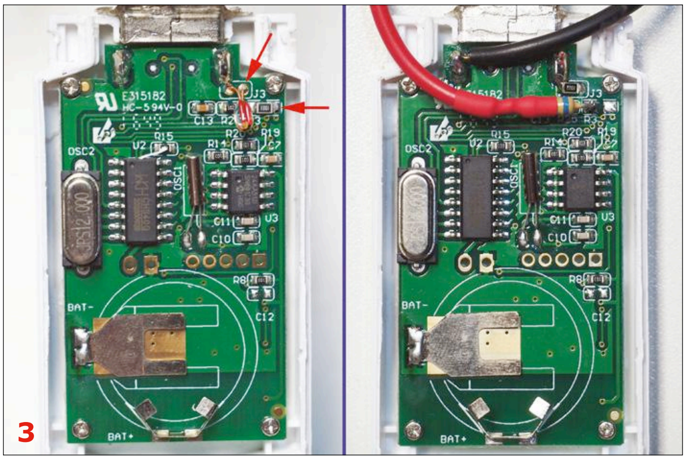
Diagnostics

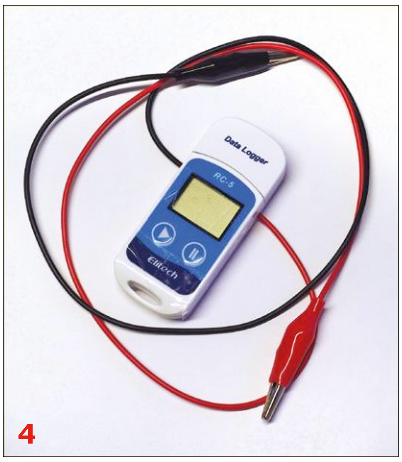
I can't blame the age of the battery this time. To get to the bottom of this strange problem, it wasn't going to be enough to measure the battery current with the car switched off. I tried that at first but my current clamp which goes down to 20 mA full scale range didn't even register a single milliamp with everything turned off so I can't point the finger at excessive standby current.

1

2

It looked like I needed to monitor the battery over a longer period to detect any 'transient event' draining the battery. What I needed was a voltage data logger; I turned to eBay hoping to find something not too expensive. Oddly I didn't find any cheap voltage logging devices listed, even from the Far East. The only users of voltage loggers are obviously professionals using top-spec loggers so there isn't a market for a cheap and cheerful version. For my purposes, I couldn't really justify the expense for what is likely to be just a one-off use. The only cheap data logging devices available were temperature data loggers... Hmm "Maybe this is a good opportunity for a little bit of tinkering?" was my first thought. I went ahead and ordered a 'Elitech Digital RC-5 USB Waterproof Temperature Data Datalogger' (Figure 2) for just over a tenner. This unit is powered by a button cell, has memory to store 215 measurements (at programmable time intervals), an LCD and a USB port to read the data. That should do the trick.





Hack attack

When the Temp logger eventually turned up I tried it out (it worked) and then set about taking it apart. Figure 3 shows the before-and-after. In the left half of the picture you can see the temperature sensor which is the small red glass bead at the top right of the picture. It connects to the 3-V supply voltage via the 100-kΩ resistor to its right. The mod is fairly simple; remove the sensor and the resistor, then solder the resistor into the position where the sensor was. A series resistor is soldered to the pad where the sensor was connected, this resistor's value now determines the voltage measuring range (see right half of the picture). According to its datasheet, the logger offers a measuring range of –30 to +80 ° C. Using a series resistor of 680 kΩ I worked out by trial and error a 'temperature' of 87.1 ° at 2.0 V and –28.0 ° at 23.0 V. This is a good match to the 12-V (nominal) battery and with 18-mV resolution around the centre of the range it should be good enough. The measuring resistance of 780 kΩ is high impedance enough not to put too much strain on the battery. Two holes are then drilled in the case for the leads. Figure 4 shows the finished mod, with flying leads terminated in croc clips.

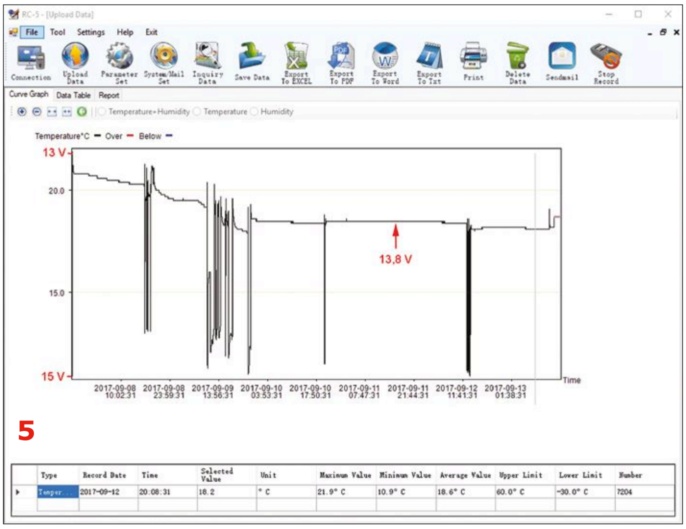
The sleuthing begins

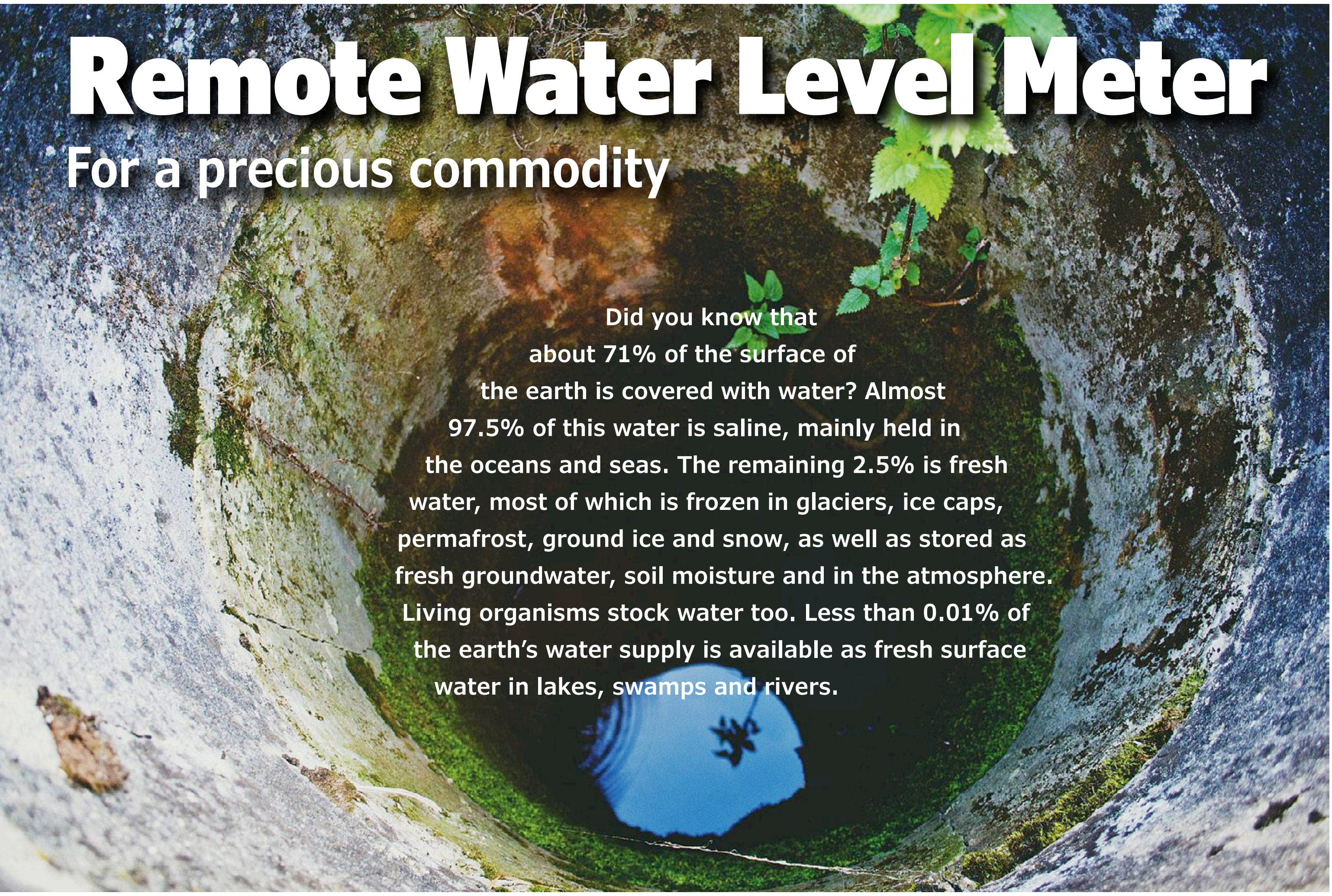
I connected up my modded voltage logger to the freshly charged 12-V battery in the car, wrapped it all in a plastic bag and initialized it with a measuring interval of 1/min, which should be good for 22 days before the memory is full. After five days, I really couldn't wait any longer; I 'recovered' the logger and looked at the readings. The bundled software (for Windows PCs) can for example, export the measured values to Excel to be shown as a graph against elapsed time. Figue 5 shows the voltage readings as a pseudo-temperature curve. A number of factors need to be taken into account when interpreting the curve. From the previously stated measuring range it follows that temperature and voltage correlate negatively: a falling temperature corresponds to increasing voltage. In addition, the temperature/resistance characteristic of the sensor (an NTC thermistor) is not linear. This nonlinearity is normally corrected in the microcontroller firmware; as a result this voltage-interpreted curve is not only 'upside down' but also quite nonlinear. In order to obtain the 'true' values of voltage, I simply entered the corresponding temperatures in a table with voltage values in 1-V increments. You can calculate reasonably accurate voltage values by just using a calculator or Excel to automatically interpolate the values.

Conclusion

In Figure 4, the minimum value is 10.9 ° C, which equates to about 15 V. The maximum value of 21.9 ° C corresponds to 13 V. The curve plateaus slightly above 18 °C ≈ 13.8 V. Apart from the occasional voltage spike (which may be a flaky connection somewhere) there really isn't much to see, at least it hasn't captured any dramatic voltage drop that I was hoping for. I'll just have to wait and see… maybe it's figured out I'm watching… I've seen films about cars with supernatural powers. Fingers crossed, so far the battery is holding up and no sign of the gremlin yet.

(160291)





Remote Water Level Meter

For a precious commodity

Did you know that about 71% of the surface of the earth is covered with water? Almost 97.5% of this water is saline, mainly held in the oceans and seas. The remaining 2.5% is fresh water, most of which is frozen in glaciers, ice caps, permafrost, ground ice and snow, as well as stored as fresh groundwater, soil moisture and in the atmosphere. Living organisms stock water too. Less than 0.01% of the earth's water supply is available as fresh surface water in lakes, swamps and rivers.

PROJECT INFORMATION

Arduino | Water | Sonar | Wireless | 433 MHz or equivalent

➡ entry level
intermediate level
expert level

4 hours approx.

Soldering iron,
computer with Arduino IDE,
USB/Serial converter

£25 / €30 / $35 approx.

By **Somnath Bera** (India)

Drinkable water is a costly commodity or 'resource' if you like, and due to the ever-increasing world population it will be even more so in the coming years. Already in many large cities in South Asia (21% of the world population lives in the so-called SAARC countries) water can be very scarce at times. Major Indian cities like Chennai, Mumbai or Colombo which rely on the monsoon for their year's water supply face acute water shortage when the rain season is shorter than usual. Good, efficient water management therefore is primordial in these places.

Although I make no claim to being a water-management specialist, I believe I have identified some problem areas that might be improved with cheap technology and a bit of ingenuity. Water level indicators for instance often create issues. Many of these are mechanical and have wires and pulleys that get jammed easily. The resulting false indications will derail the management system of the water tanks that use them.
In the light of the above I present in this article a sonar-based contact- and wireless water level indicator which can be slapped up over a weekend for less than €20 (£18 / $24). With two separate power supply units the price of a complete set comprising a water level

transmitter and receiver will be under €30 (£26 / $35). This may sound impossible, yet it is true.

The proposed water level indicator can precisely monitor levels up to four metres (12 ft.) with an accuracy better than 5 mm, and without any wiring between the tank and the display unit at your drawing room.

The indicator is based on the popular ultrasonic ranging module SR04 capable of measuring distances from 0.20 m to 5 m (0.6 to 15 ft.). The ranging accuracy is 3 mm and its aperture angle is 15°. But since we will be measuring levels in integer centimetres our accuracy is a mere 5 mm. Every second eight readings are taken and averaged, allowing 1-cm level changes to be detected very clearly.

Besides the ultrasonic ranging module the transmitter requires an Arduino (Pro) Mini compatible board (an Arduino Uno will work equally well but is bigger and more costly) and a 433-MHz ISM transmitter module. For the receiver an Arduino (Pro) Mini compatible board (again, an Arduino Uno will work equally well) is needed together with a 433-MHz receiver module and a standard alphanumeric liquid crystal display. All these parts can be found online at very low prices. Together with two antennas, and two 5-V / 2-A power supplies I paid about €22 (£19 / $26), shipping included. This leaves a good €8 (£7 / $9.50) for the remaining discrete components and some prototyping board.

Principles of operation

The water level sensor is built on the Arduino platform to which I added an SR04 ultrasonic sensor and a 433-MHz transmitter module equipped with a long-distance spiral antenna (**Figure 1**). The sensor is to be fitted at a strategic location of the tank such that it can always receive the reflected signal from the water surface. The best point is at the centre of the tank for circular tanks or at the intersection of the diagonals for rectangular shaped tanks.

The 433-MHz band part of the open ISM spectrum where anyone is allowed to send as long as the signal strength remains below a certain level and type approved equipment is being used. Note that the 433-MHz ISM frequency allocation differs between countries, so you may have to find your local equivalent. There are no protocols defined such as Wi-Fi or Bluetooth, and it is up to the user to imagine a suitable communication protocol for his/her application. Therefore, in order to segregate the desired signal from the undesired a special code word is transmitted with every water level reading. On detecting a signal, the

Features

- Contactless measurement
- Wireless connection
- Low cost
- Only common parts needed

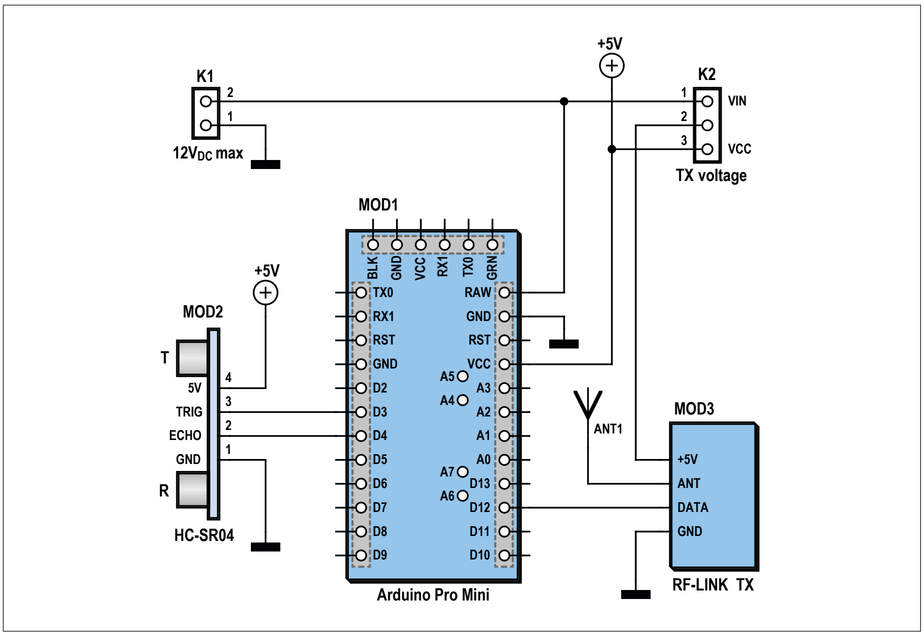


Figure 1. The sensor side of the water level meter.

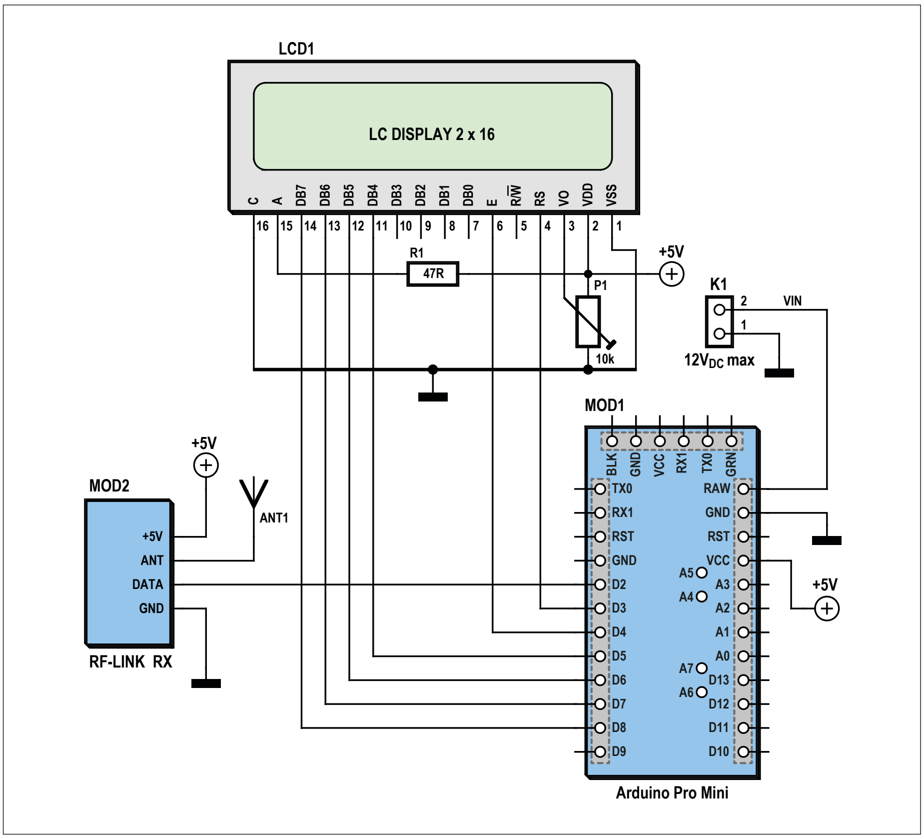


Figure 2. The remote water level display: an Arduino board, a receiver module and a display module.

Figure 3. Badminton rackets are great for quickly building a test setup.

receiver tries to extract the code word to verify it against a known value. If it matches, it prints the received water level on its LCD (**Figure 2**).

COMPONENT LIST

TRANSMITTER

K1 = 2-way PCB screw terminal block, 0.2" pitch
K2 = 3-pin pinheader, 0.1" pitch
MOD1 = Arduino (Pro) Mini or Uno
MOD2 = HC-SR04 ultrasonic transceiver module (or similar, HY-SRF05 can be used too)
MOD3 = 433MHz transmitter module (see text)
Jumper for K2
433MHz antenna (spiral / whip / 173mm stiff wire)

RECEIVER

R1 = 47Ω
P1 = 10kΩ trimpot
K1 = 2-way PCB screw terminal block, 0.2" pitch
LCD1 = 2×16 characters alphanumeric LCD
MOD1 = Arduino (Pro) Mini or Uno
MOD2 = 433MHz receiver module (see text)
433MHz antenna (spiral / whip / 173mm stiff wire)

Since there is no connection between the transmitter and the receiver, the latter has no way of knowing if the transmitter has stopped transmitting, for instance due to a power supply failure. To circumvent this problem a timestamp (continuously counting from 0 to 59 seconds) is included in the transmitted message; its value is shown on the right side of the display. An non-changing timestamp is an indication that the incoming signal has stalled. The LED connected to Arduino pin 13 will stop blinking too.
Together with the small coil antennas which I used and a 5-volt supply the transmitter is capable of transmitting the signal up to 400 metres (1200 ft.) effortlessly. To increase the range the V_{CC} pin of the TX unit may be connected to 12 V as most of these FSK transmitters can operate from 4 to 12 V.

Some words about the software

The software for this project comprises two Arduino sketches, one for the transmitter and the other for the receiver. Both make extensive use of open-source libraries to accomplish their respective tasks, with the result that the sketches have remained pretty simple.

The transmitter sketch

The transmitter requires the NewPing library to drive the ultrasonic sensor. This library takes care of all the dirty work and can output a measured distance in centimetres. After creating a sonar object, all that is needed is a call to `ping_cm`. In this sketch this function is called eight times in a row at 100-millisecond intervals, and the results are averaged. The obtained value is put in a buffer together with the timestamp and the special code (transmitter ID).
The VirtualWire library takes care of sending the data with the 433-MHz FSK transmitter module, which is connected to Arduino pin 12. After configuring the library, sending data is simply a matter of calling `vw_send` and waiting for the transmission to finish with `vw_wait_tx`. The above procedure is repeated every second.

The receiver sketch

On the receiving side things are equally simple. Again the VirtualWire library is used but now in receive mode. The library is configured for listening to a receiver module connected to Arduino pin 2. The function `vw_rx_start` must be called to put the library and module in listening mode. Once in this mode, calling `vw_get_message` is all that is needed. When data is available, it will be read and copied into the receive buffer; when there is no data, it will do nothing.
Supposing that a message has been received, it is checked for validity:

- Does it have the right number of bytes?
- Does it contain the code word mentioned above?
- Is the value equal or larger than zero?

When all these criteria are met, the value is printed on the LCD. The timestamp contained in the message is extracted and printed too.
As has become standard in the Arduino world, the LiquidCrystal library takes care of writing on the LCD. The LED is on while the display is being updated, providing "long-distance" feedback that new data has been received.

Where to put the sketches?

The software (including the libraries) is packed in a folder labelled 'portable'. Simply add this folder and its contents to the folder where the Arduino.exe file is located. If your installation already has a 'portable' folder, put the files in there.

If there is not yet a 'portable' folder, creating it will disconnect the Arduino IDE from your current set-up. In this case, to revert to your current set-up, simply delete the 'portable' folder you created and its contents when you're done. It is of course also possible to put the sketches and libraries in the normal places.

Extending the system

The presented system offers many possibilities for extensions. For instance, adding a second sensor only requires that it gets a unique ID (the special code word can be used for this). The receiver unit must know this ID too and update its display accordingly. In theory the system described above has room for 65,536 sensors (determined by the size of the special code word); in reality it will of course be limited by the available data bandwidth and the receiver's processing power.

Another interesting possibility is to add a serial connection to a computer where the water level values can be logged in a database. Use a spreadsheet to calculate statistics and create nice-looking graphs. Since most of the pins of the ATmega328 are free, various relays and/or contactors can be connected to these pins to control for instance pumps and alarms.

Building and testing

Assemble the transmitter and the receiver units on for instance prototyping board. This should not pose problems since the circuits are not complicated. Note that the ultrasonic transducer comes in several flavours and most can be used. The 5-pin types like the HY-SRF05 are fine too, simply shift pins 2 to 4 one pin up and leave pin 2 unconnected.

When done, program the right sketch in the right unit. If you used Arduino (Pro) Mini boards, you will need a USB/Serial converter cable for this.

Power both units. The transmitter LED should start flashing at a rate of about 1 Hz. The receiver should show some text on its display. If it doesn't, try adjusting the contrast trimmer first before looking for wiring problems.

Figure 4. Most of the world's fresh water supply is frozen enabling penguins to sit on it.

When the display unit receives the signal from the transmitter, its LED will start blinking at a rate of about 1 Hz, synchronously with the transmitter unit. The display will show a value in centimetres and a two-digit counter will start running. Put the sensor in front of a large object or wall at a known distance, e.g. one metre, and verify that the value displayed is correct. The author used a bucket, a plastic ruler and a badminton racket to verify the operation of the sensor in a test system (Figure 3).

That's about it; all that is left to do is installing the units. For best transmission range mount both antennas vertically.

(160626)

Web Links

[1] Article support page: www.elektormagazine.com/160626

[2] Elektor Labs: www.elektormagazine.com/labs/remote-water-level-meter-a-need-of-the-hour

eLABS PROJECT

Platino "Fits-All" Serial Bus Tester

Debug serial comms issues with ease

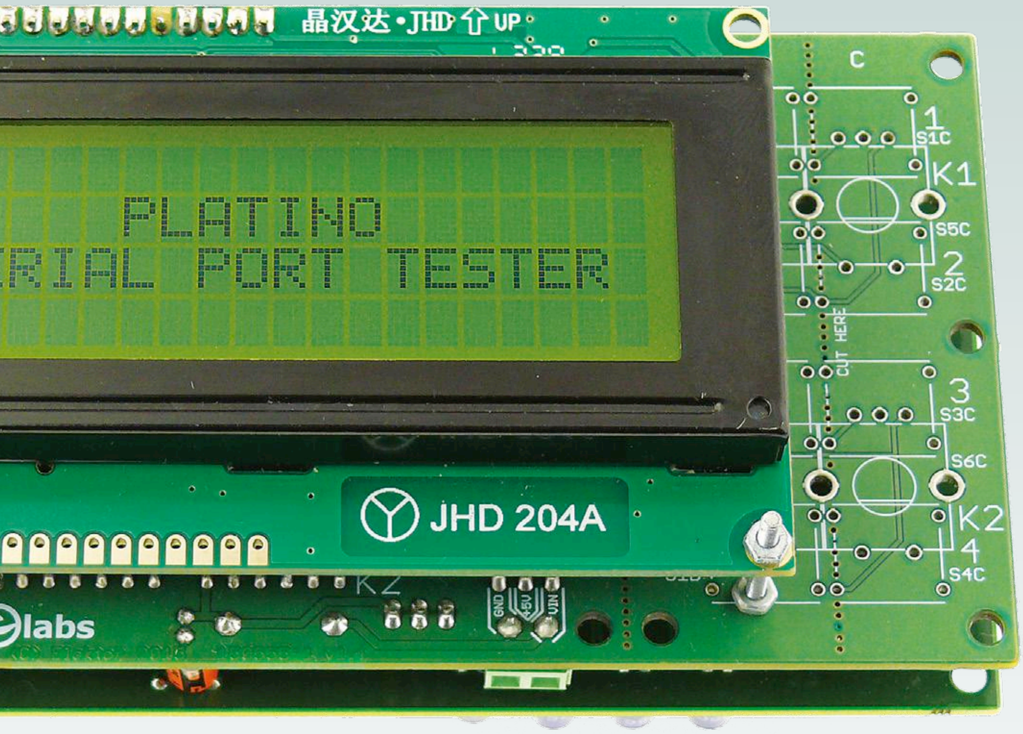
By **Sunil Malekar** & **Clemens Valens** (Elektor Labs)

Here's a microcontroller that's supposed to talk to a device over a SPI bus. You've set up the MCU's SPI peripheral and wrote a test program but the device does not respond. The SPI bus connections got checked several times and were found to be correct so the problem must lurk in the data format. Sounds familiar? Then you need an intelligent tool to visualize what's going on those wires or pins. Enter the Platino Serial Bus Tester.

Even though the more expensive oscilloscopes on the market can interpret certain digital signals formatted in a known way, like SPI, I²C or plain serial, a microcontroller can do this much better. In this article we therefore present a Platino-based system that speaks and understands various serial protocols, and is intended as a debugging tool for serial communication busses.

A word about vocabulary

Let's start by defining some context. A 'bus' is a group of signals. A 'communication protocol' is a set of rules that dictates the way data gets conveyed



Features

- Full control over serial communication busses
- 100% Arduino compatible
- User extendable
- 4 types of hardware interface
- Supports 1.2-V to 5-V digital swing

PROJECT INFORMATION

Platino · Arduino · 1-Wire · Serial Communications · SPI · MIDI · I²C · PMBus · SMBus

entry level
→ intermediate level
expert level

4 hours approx.

(SMD) Soldering iron, AVR programmer, serial-to-USB converter, computer

€75 / £70 / $80 approx.

between computers. A 'standard' is a convention against which something similar can be measured or compared. 'Serial' means one bit at a time. 'Synchronous' serial signals comprise separate data and clock signals; 'asynchronous' serial signals by contrast have data signals only.
In this article we will use the term 'protocol' to indicate how bits are supposed to be ordered on the line (e.g. one start bit, eight data bits, no parity, and one stop bit). 'Standard' will refer to the physical expression of the data (like voltage swings and signal polarities on the line).

Specifications

With the above vocabulary read & absorbed we can write down the following requirements for our test tool:

- multi-protocol;
- multi-standard.

That's not much of a "list" whereas our project is an excellent object for featuritis or list-o-mania — let's add some more detail.

Multi-protocol (one at a time):

- plain-vanilla serial;
- I²C / SMBus / PMBus;
- SPI;
- any others, like 1-Wire?

Multi-standard:

- RS-232;
- RS-485;
- MIDI;
- 5 V;
- 3.3 V.

With these standards we cover a vast range of possible protocols. The popular protocols are present by default, while more exotic ones can be added later. From an operational perspective we specify flexibility.

Multi-standard hardware

As mentioned above we chose the Elektor Platino platform [1] fitted with a backlit alphanumerical display with four 20-character lines, a rotary encoder, a pushbutton, a buzzer, and a tricolour LED as the basis for our project. Equipped with an ATmega1284P microcontroller it has ample memory (128 KB program memory plus 16 KB of RAM) for serial protocols and nifty features. To that we added an extension board with the required interfaces for our serial standards (**Figure 1**). If we load a suitable bootloader into the MCU the whole system becomes Arduino compatible and we can profit from the large number of Arduino libraries available online.

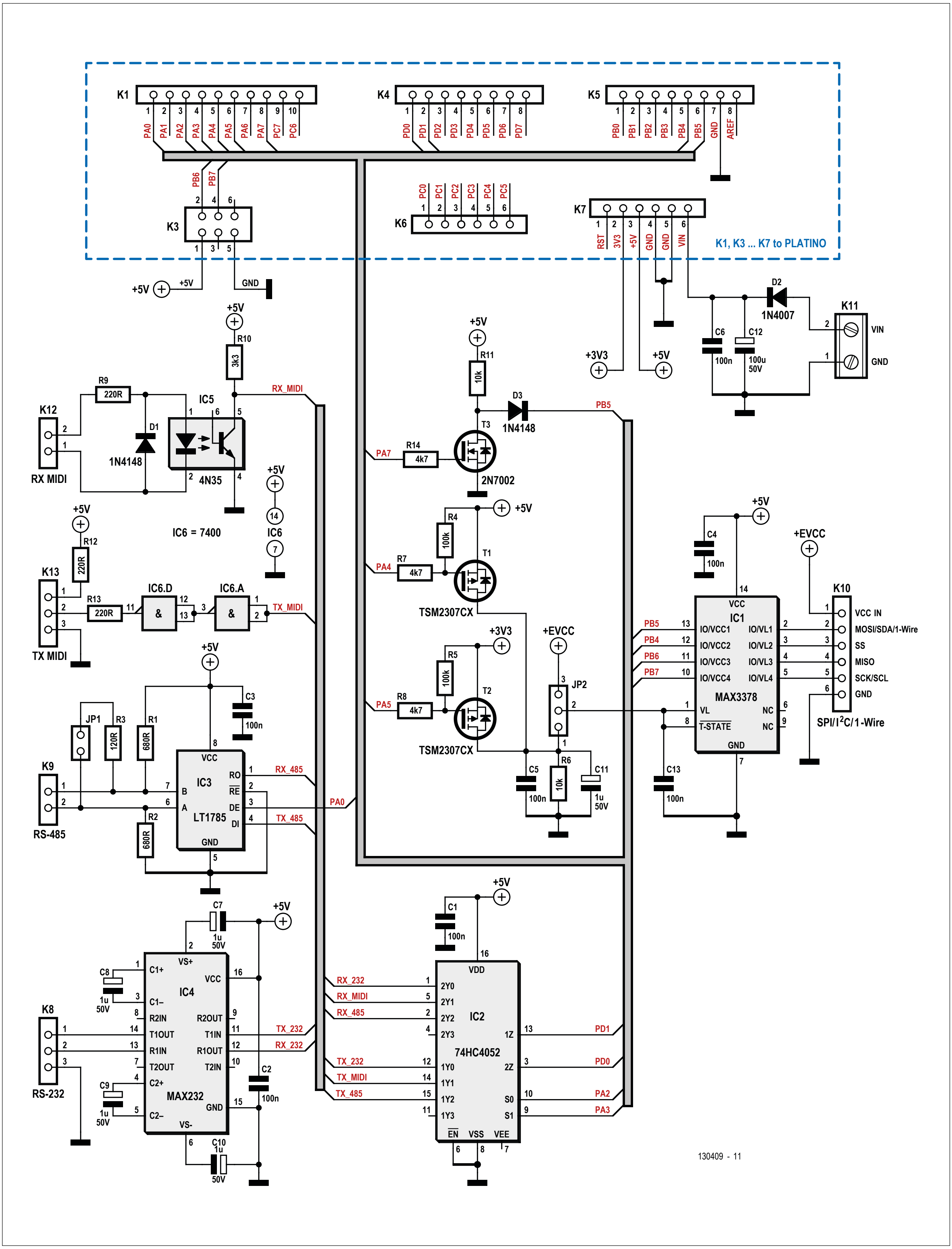


Figure 1. The Platino multi-standard serial interface board has interfaces for RS-232 (K8/IC4), RS-485 (K9/IC3), MIDI (input on K12/IC5, output on K13/IC6) multiplexed by IC2. K10/IC1 provide the interface for I²C, SPI and other 5-volt (T1) or 3.3-volt (T2) level signals where IC1 ensures the required level shifting. Other signal levels are possible through EVCC (1.2 V to 5 V) and by mounting JP2. T3 is intended to act as the 1-Wire output.

Multi-protocol software

Because our space is limited we will not waste it on such futile details as hardware descriptions. Please refer to [1] if you want to know more about the Platino platform.
Even though the ATmega1284P has hardware peripherals for the main serial protocols "UART", I^{2}C and SPI, its fixed pinout prevents us from profiting from all of them together with Platino's user interface elements. For this reason we decided to handle the relatively low-speed I^{2}C protocol (400 kHz max.) in software, giving the SPI bus the opportunity to fully express itself up to 8 MHz. This also implies that only one of the two hardware UARTs is available, as PD2 (RXD1) and PD3 (TXD1) are needed to control part of the LCD instead of the SPI lines PB6 (MISO) and PB7 (SCK). For TTL-level serial data a software UART is required using some of the pins of the SPI bus.
In case it wasn't clear already, our tester will "do" only one serial protocol at a time.

Ergonomics come first

Because the tester's usefulness depends for a large part on its ergonomics, we have spent quite some time on fine-tuning the user interface (UI). The objective was to make using the device as easy and comfortable as possible whilst the display has limited space and cannot do colourful graphics. All the parameters are accessible by rotating and pushing the rotary encoder, and without going through many layers of menu. The result is, paradoxically, a rather cryptic main screen. However, as you will quickly discover, it's easy to master (**Figure 2**). Spinning the encoder will move the little arrow to the next parameter; pushing the encoder will change the parameter's value or enter Edit mode for parameters with a wide range (like serial baud rate).

A suitable menu system

The menu system that made this UI possible was created using the open source Arduino library LiquidMenu [2] that relies on the presence of the LiquidCrystal library (available in every standard Arduino installation). It requires the C++11 standard (or more recent), meaning that the Arduino toolchain (AVR-GCC) must be recent as well. Installing the latest Platino Boards Package [3] will make sure this is the case. We have extended and modified the library a bit (and corrected a few bugs in the process), so please use the download from [4] to make it all work.
LiquidMenu provides a menu system built, unsurprisingly, out of one or more menus. A menu consists of one or more screens and a screen comprises one or more entries.

Menu entries

An entry can be placed anywhere on the display and it can be linked to up to five variables (by default — this is configurable). Variables can be of any type, from floating-point values to text strings, and are automatically converted to readable text when printed. Each entry can also link to up to eight functions (again, configurable) that can be used to activate things or modify related variables. Entries that do not have any functions linked to them cannot get focus, and as such are mere text objects.
Note that inside the library the menu entries are called menu lines (`LiquidLine`), which is a bit confusing as they have nothing to do with lines on the display.

Screens

Menu entries, up to 12 by default, are combined into screens (`LiquidScreen`). An entry can be a member of several screens, like the Back ("<<") entry in our system. A screen "knows" which entry has focus, and calls the linked functions.

Menus

One or more screens form a menu (`LiquidMenu`). Up to 14 screens can be added to a menu. In our system, however, we needed only one screen per menu. The menu keeps track of the active screen. Multiple screens allow for long menus (12 × 14 = 168 entries).

System

Finally, menus (up to 8) can be added to a menu system (`LiquidSystem`), but this is not necessary. **Figure 3** summarises the system. We used one system that we called 'ui' to which we also added other UI functions like beeps and flashes.

What the ...?

If at this point you are wondering why an article about a serial port tester discusses menu structures, please be patient and read on. As will become clear, the communication and test functionalities actually make up only a small part of this

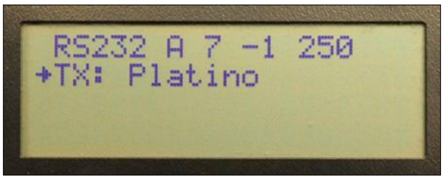


Figure 2. The Serial Bus Tester's main screen shows on the top line the serial protocol and standard being used, the data display format (ASCII or hexadecimal), the number of bytes to be transmitted (up to 7), the number of times to do a transmission (0 to 99, or forever), and the time between transmissions in milliseconds (9,999 ms max). The second line holds the data to be transmitted. Lines three and four are reserved for incoming data.

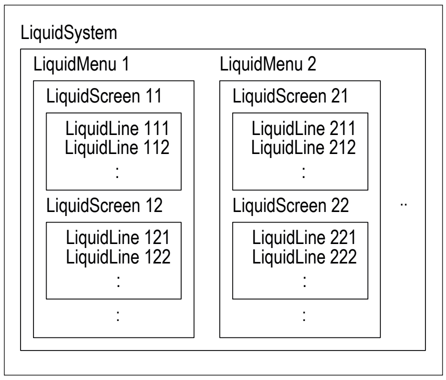


Figure 3. The LiquidMenu library offers a menu system organized not unlike matryoshka dolls.

project. About 80% of the firmware, and maybe even more, revolves about ergonomics. Because this part of the software is so big, it is important to understand how it works in order to make changes to it to add for instance other communication protocols or test options.

The main screen

Implemented in the file ui_main.cpp, the main screen features all the different menu techniques that have been employed:

- entering a submenu: Protocol;
- iterate through values: ASCII or HEX data display;
- spinning to a value: TX data size and number of times to transmit;
- editing a number: delay between transmissions;
- editing a string: TX data.

For the description that follows it is assumed that the serial tester is not in test mode, but in menu navigation mode.

A quick word on Platino

Platino is a universal circuit board for AVR microcontrollers in DIP28 or DIP40 packages. It can be equipped with user interface objects like pushbuttons, LEDs, buzzer, and displays. Everything, and we mean everything, about Platino can be found on the Elektor Labs website (see below) and on the GitHub repository of Elektor Labs. Experience has shown us that Platino users find it hard to configure its solder jumpers. This is understandable, as there are many. The table below shows the settings for this project. The Position column corresponds to the labels printed on Platino PCB version 150555-1 v1.4. Note that we recommend placing solder jumpers before mounting any parts.
And while we're on the subject of parts: for this project we need a fully assembled Platino with a 40-pin MCU (ATmega1284P-PU) — so no need for a 28-pin socket —, a 4x20 alphanumerical LCD with backlight, a rotary encoder with integrated pushbutton on position S5A or S5C (depending on your preference) and a pushbutton on position S4A or S4C (again, as you like). The Platino bill of materials can be found here:

www.elektormagazine.com/labs/platino-versatile-board-for-avr-microcontrollers-100892-150555

Jumper	Position	Function
JP1	C	Buzzer on PC4
JP2	-	Non existent
JP3	C	LCD backlight on PC5
JP4	B	Rotary encoder A on PB0
JP5	B	Rotary encoder B on PB1
JP6	B	Rotary encoder pushbutton on PB2
JP7	B	Pushbutton on PB3
JP8	DIP40	Allow PC6
JP9	PB7	Allow PB7
JP10	PB6	Allow PB6
JP11	DIP40	ISP SCK on PB7
JP12	DIP40	ISP MISO on PB6
JP13	DIP40	ISP MOSI on PB5
JP14	C	LED1 blue on PC7
JP15	D	LCD RS on PD2
JP16	D	LCD E on PD3

Also a note about naming conventions: the beginning of function and variable names indicates the file where they are implemented or declared. As an example, the function `ui_bus_type_refresh` lives in the file ui_bus.cpp. (See **Figure 4**)

Entering a submenu

On the main screen, positioning the cursor (i.e. the arrow) to the left of the protocol name by rotating the encoder and then pressing its pushbutton will bring up the protocol's submenu where you can set the parameters specific to the protocol.
Since we are in navigation mode when doing this, pressing the encoder will call the first function attached to the menu entry. In the case of the protocol menu entry this is `ui_bus_enter`. This will load the screen for the submenu and then activate and display the submenu.

Iterating through values

On the main screen, positioning the cursor to the left of the data display format field ('A' or 'H') and then pressing the encoder's pushbutton will change the field's value. This field has only two values, but entries of this type in other menus have more. Pressing the rotary encoder repeatedly will iterate through all the possible values.
As before, because we are in navigation mode when doing this, pressing the encoder will call the first function attached to the menu entry, which in the case of the data format menu entry is `ui_main_data_format`. It will update the value and reprint the TX data in the new format.

Spinning a value

On the main screen, positioning the cursor to the left of either the TX data size or transmit count fields and then pressing the encoder's pushbutton will activate parameter edit mode. Rotating the encoder will increment or decrement the field's value (0…7 or -1…99). Press the rotary encoder again to leave parameter-edit mode.
Here we start in menu navigation mode, hence pressing the encoder will call the first function attached to the menu entry, which in the case of the TX data size menu entry is `ui_main_tx_data_size`. This function activates edit mode and commands the rotary encoder to use the function `ui_main_tx_data_size_increment` to modify the field's value.
In edit mode the encoder's pushbutton acts differently. Pressing it will advance to the next digit or character. When there is no next character to edit, it will switch back to menu navigation mode. The TX data size field has only one digit. The transmit count field has two digits, but since it can be negative which seriously complicates digit-by-digit editing, it is treated the same way as a single digit value.

Editing a number

On the main screen, positioning the cursor to the left of the repetition rate field and then pressing the encoder's pushbutton will again activate parameter edit mode. Because this field can have up to four digits ('0000' to '9999'), rotating the encoder will increment or decrement the first digit. Pressing the rotary encoder again will move the cursor to the next digit. When there are no more digits to edit, the system will switch back to menu navigation mode.
This type of editing is almost identical to spinning a value, except that there is more than one digit. Also some cursor fiddling is required to work around LiquidMenu's display handling. Instead of adding application-specific functions to the library, we preferred to work our way around it.

Editing a string

On the main screen, positioning the cur-

```
_130409_serial_bus_tester.ino
  |__ addon_board.cpp
  |__ test.cpp
        |__ serial.cpp
        |__ spi.cpp
        |__ soft_i2c.cpp
  |__ ui.cpp
        |__ ui_splash.cpp
        |__ ui_main.cpp
              |__ ui_bus.cpp
                    |__ ui_bus_serial.cpp
                    |__ ui_bus_spi.cpp
                    |__ ui_bus_i2c.cpp
                    |__ ui_bus_smb.cpp
                    |__ ui_bus_pmb.cpp
                    |__ ui_bus_midi.cpp
                    |__ ui_bus_1wire.cpp
```

Figure 4. The source code "tree" has the Arduino INO file at its root, all other files are C++ files. The header files are not shown for clarity. Most of the files are concerned with the user interface (UI).

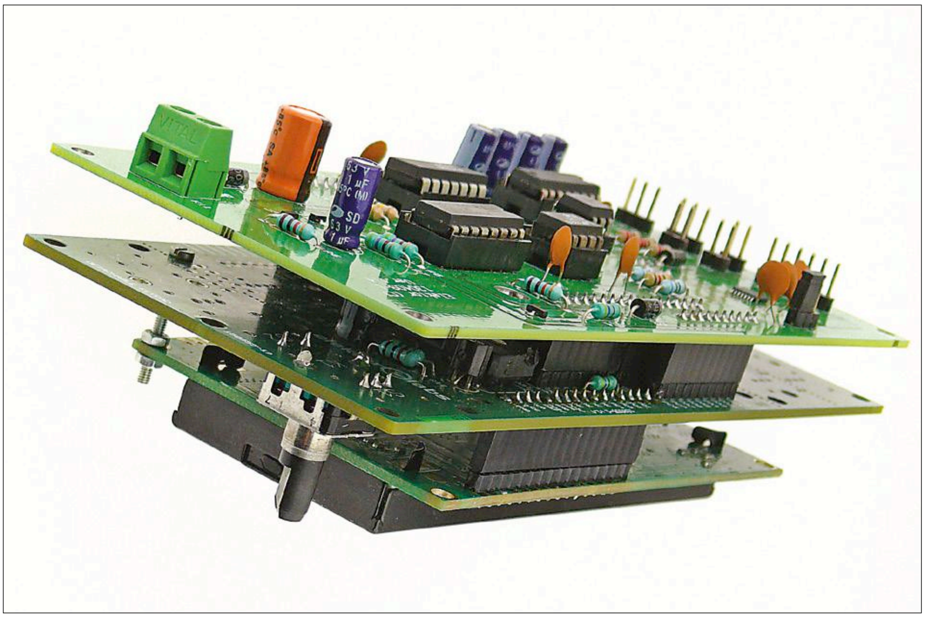


Figure 5. The Platino Serial Port Tester all stacked up. Platino is in the middle, the LCD is at the bottom, the interface board sits at the top.

sor to the left of the 'TX' field (at the beginning of the second line) and then pressing the encoder's pushbutton will activate the most complicated edit mode: string editing. It is complicated because it can work in ASCII as well as in hexadecimal mode. However, in hexadecimal mode a string is twice as long as in ASCII mode. Furthermore, in ASCII mode characters are edited as bytes, whereas in HEX mode they are edited nibble by nibble. For the rest, string editing works in the same way as number editing.

Note that TX data can be up to seven bytes long. This upper limit is imposed by the available space on the second line. When the data display format is hexadecimal there is just enough space to print seven hexadecimal values. The third and fourth lines are reserved for RX data and we did not want to eat into that.

More about protocols and their submenus

After entering the protocol submenu you can iterate through the supported protocols by selecting the protocol field and pressing the encoder. To go back to the main screen, select 'Back' ("<<").

For every protocol the most important parameters can be set:

- RS-232 and RS-485: baud rate (from 1 to 2,000,000 baud), number of data- and stop bits (5 to 8, resp. 1 or 2), and the parity (odd, even or none);
- SPI: Master/Slave, MSB/LSB first, clock phase and polarity, speed and signal level;
- I²C: Master/Slave, address, number of bytes to write and to read, speed and signal level;
- MIDI: channel, status, two data bytes.

> The tool you always wanted to have but never got round to building

At the time of writing these were the implemented protocols, but there may be more when you download the latest version of the software.

From a user interface perspective there is one detail that should be mentioned. When a protocol like SPI or I²C is in Master mode, its name is written in capitals (i.e. 'SPI' or 'I2C'); when it is in Slave mode, it is written in lower case (i.e. 'spi' or 'i2c').

SPI

This protocol has four well-defined signal polarity standards, usually numbered from 0 to 3, and related to the clock phase (CPOL) and polarity (CPHA). The serial port tester supports all four of them. Furthermore, the signals can have 3.3-V or 5-V levels.

The MCU's hardware peripheral is used for SPI which offers eight clock speeds in Master mode: 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz and 8 MHz (when the MCU is clocked at 16 MHz). In Slave mode the clock speed is, of course, determined by the Master.

I²C

Like SPI, the I²C signals can have 3.3-V or 5-V levels. Unlike SPI, I²C is implemented as a software peripheral and as such does not support every feature defined by the maintainers of this communication standard. The maximum speed is 400 kHz, which is feasible with careful programming.

In Master mode the specified address is considered to be the Slave address. Only 7-bit I²C addresses are supported. (Sometimes device datasheets specify 8-bit addresses, one for reading and one for writing. If they do, shift the addresses one bit to the right to obtain the 7-bit address. They will be identical.) The serial tester inserts the Slave address automatically into the data at the right moments together with the read/write bit and does not show it in the TX data on the main screen. In Slave mode the address is the serial tester's own address.

Read, write and write-read transactions are supported by setting the number of

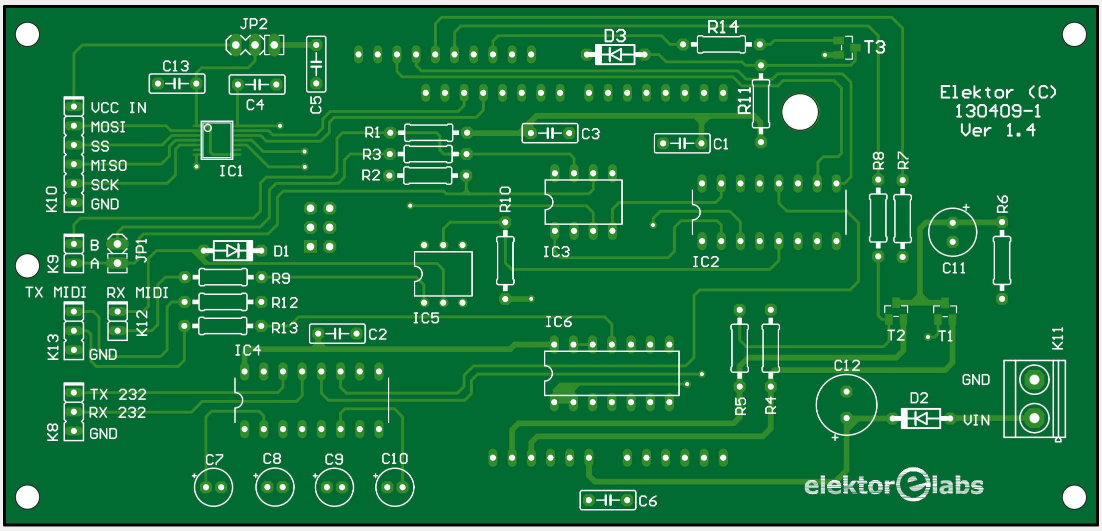
COMPONENT LIST

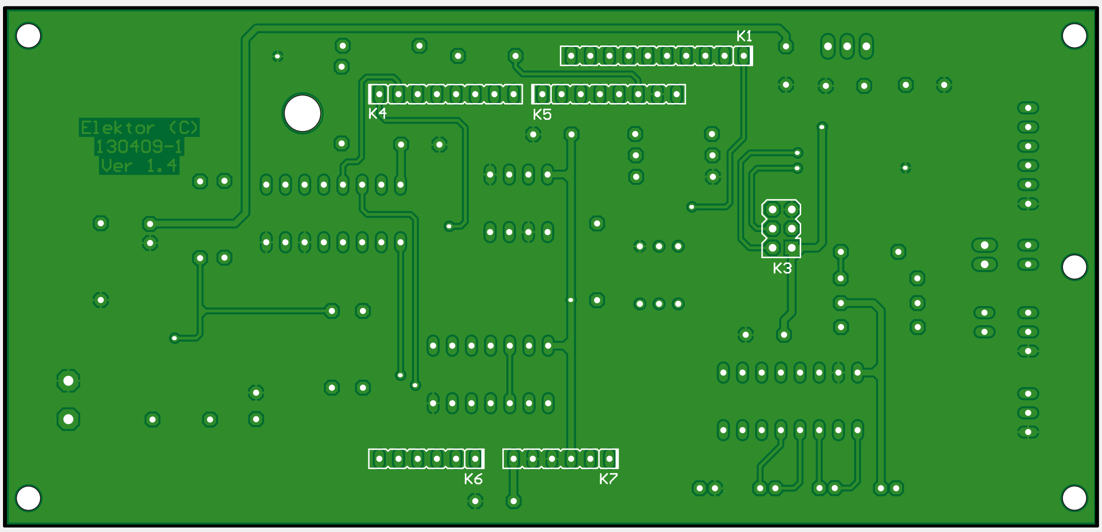
Resistors
All 5%, 0.25 W
R1,R2 = 680Ω
R3 = 120Ω
R4,R5 = 100kΩ
R6,R11 = 10kΩ
R7,R8,R14 = 4.7kΩ
R9,R12,R13 = 220Ω
R10 = 3.3kΩ

Capacitors
C1,C2,C3,C4,C5,C6,C13 = 100nF, 5mm pitch
C7,C8,C9,C10,C11 = 1µF, 16 V, 2.5mm pitch
C12 = 100µF, 16V, 2.5mm pitch

Semiconductors
D1,D3 = 1N4148
D2 = 1N4007
T1,T2 = TSM2307CX
T3 = 2N7002
IC1 = MAX3378EEUD+
IC2 = 74HC4052
IC3 = LT1785
IC4 = MAX232
IC5 = 4N35
IC6 = 74HC00

Miscellaneous
K1 = 10-way (1x10) pinheader socket, 0.1" pitch
K3 = 6-way (2x3) pinheader socket, 0.1" pitch
K4,K5 = 8-way (1x8) pinheader socket, 0.1" pitch
K6,K7 = 6-way (1x6) pinheader socket, 0.1" pitch
JP1,K9,K12 = 2-pin pinheader, 0.1" pitch
JP2,K8,K13 = 3-pin pinheader, 0.1" pitch
K10 = 6-pin pinheader, 0.1" pitch
K11 = 2-way PCB screw terminal block
2 pcs 2-way jumper for JP1 and JP2
DIP-6 socket for IC5 (optional)
DIP-8 socket for IC3 (optional)
DIP-14 socket for IC6 (optional)
2 pcs DIP-16 socket for IC2, IC4 (optional)
PCB # 130409-1





bytes to write and to read.
SMBus and PMBus are a kind of low-speed simplified variants of I²C that can append a so-called Packet Error Checking (PEC) byte to the end of a message. The serial tester calculates it automatically and does not display it as TX data.

MIDI

This is a fixed-speed (31.25 kHz) variant of the standard serial protocol. However, because MIDI messages usually contain only three well-defined bytes they can be easily configured on the MIDI protocol page. The channel and status (the text string after the channel field) are combined into one byte, and the two data bytes are appended. The resulting 3-byte message is shown as hexadecimal TX data on the main screen where it can be edited. Here it is also possible to compose shorter or longer messages (up to seven bytes), useful for e.g. system exclusive messages.

Beeps

The UI uses three types of beeps to provide audible feedback on user actions:
high: enter submenu, test-, or parameter-edit mode;
mid: advance to next editable digit or character;
low: leave submenu, test-, or parameter-edit mode.

Selecting menu entries that iterate through values produce low-frequency beeps because the action corresponds to the sequence of entering edit mode, changing the value and then leaving edit mode again. Since the last action takes the user back to the beginning, a low beep sound is appropriate.

And then, finally, Test mode

The serial tester has a dedicated pushbutton (Platino pushbutton S4) for entering and leaving Test mode. It may be pressed at any time. If pressed while a submenu is being displayed, the system will switch to the main screen.
While in Test mode, it is possible to change the setup of the tester. In some cases this will result in test mode being switched off. A few parameters can be changed on the fly, but not all, and to be sure that a parameter change is taken into account it is best to restart test mode.

LED

The tricolour LED will light up red when test mode is activated, it will be green otherwise (it is orange in parameter edit mode, mainly just because it is possible). Blue is currently not used.

RX data

Any incoming data is displayed on lines three and four of the display. Data right under the TX field is oldest; data in the lower-right corner is newest. There are no separators between bytes.
The RX data field is cleared at the start of each test.

Repeated transmissions

It is possible to not transmit anything either by setting the TX data size to zero or by making the transmit count equal to 0. Setting the transmit count to −1 will result in repeating forever. The delay between two transmissions is set by the repetition rate, in milliseconds.

SPI

Inherent to the operation of SPI, when in SPI Master mode, data can only be received when the Master is sending. A trick to limit to receiving is to continuously send either the value 0 or the value 0xff.
In SPI Slave mode the tester will send the TX data when the SPI Master generates a clock signal. TX data will loop if more than TX data size bytes are requested.
Note that the first byte received usually is junk left behind in the receive data register. This again is due to the way SPI works.

I²C

Similar to SPI, an I²C Master can only receive data when it generates a clock signal. Furthermore, the Slave must have been put in read mode. These conditions will be satisfied when the number of bytes to read in the I²C submenu is non-zero. In Slave mode the tester will send the requested number of TX data bytes. In case more than TX data size bytes are requested, the TX data will loop.

We bit the bullet for you

The Platino Serial Bus Tester is one of those tools that you always wanted to have whenever you ran into a serial communications problem, but never got built because when you needed it you didn't have the time to build it. This is why we, after thirty years of hesitating, finally bit the bullet and built it for you (**Figure 5**).
The design is open and clear, making it easy to understand, extend and/or modify. The software is written in a modular way and is fully Arduino-compatible, allowing the quick addition of new protocols. Less than 20% of the program memory and 10% of RAM has been used. The presented menu system is perfect for use in other (Platino) projects.
All considered we believe that our Serial Port Tester is a nice and useful reference design to build larger (Platino) projects on. ◂

(130409)

Web Links

[1] Platino: www.elektormagazine.com/150555
[2] LiquidMenu: https://github.com/VaSe7u/LiquidMenu
[3] Platino Boards package for Arduino: https://github.com/ElektorLabs/Arduino
[4] Article support page: www.elektormagazine.com/130409

Supercaps

Low voltage but lots of current... or not?

By **Tam Hanna** (Slovakia)

The falling price of supercapacitors means that they can now be considered for homebrew designs and small production runs. We take a closer look at the different types of supercap and consider some of the theoretical aspects of their use. Last but not least we use a new IC designed for supercap charging.

Figure 1. Supercaps from AVX (Photo: AVX).

In the third edition of that well-known electronics bible *The Art of Electronics* by Horowitz and Hill there is a so-called 'Ragone chart' on page 690 which compares the energy density of supercaps with capacitors and batteries. It shows that supercaps are moving into the category of the classic rechargeable battery. This prompted the author to delve a little deeper.
The first thing to be aware of is that the chart has a logarithmic scale, in reality supercaps still have a long way to go before they get close to rechargeables when it comes to energy density. However that doesn't mean that they should be just discounted. Supercaps differentiate themselves from batteries in that they can deliver enormous levels of current per unit weight. Supercaps also boast an almost unlimited number of charge/discharge cycles provided they are not thermally stressed in the process.

Figure 2. Some Panasonic Goldcaps (Photo: Panasonic).

Two types of supercap?

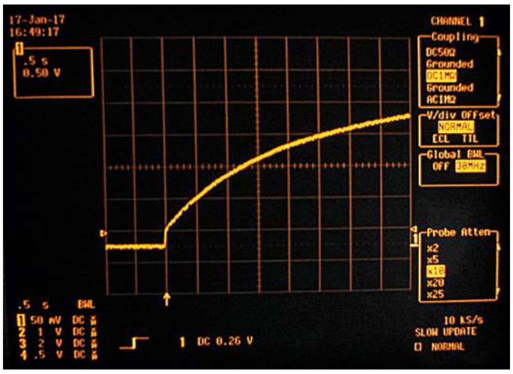
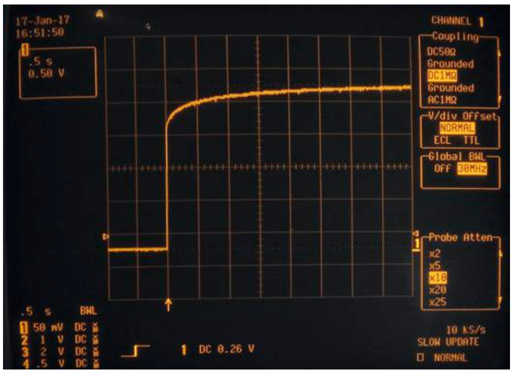
When choosing a supercap for an application you quickly realize that there are in fact two basic types which distributors like to class as 'Supercaps' or 'Ultracaps':

- The first type of supercap has very low internal impedance and is the type we will be looking at here (**Figure 1**).
- The second type, mainly used for applications such as keep-alive memory power, is called a Goldcap. Although these cannot supply high levels of current they have very low self-discharge (**Figure 2**).

The most important data for a Supercap or Goldcap capacitor are shown in **Table 1**. Notice that the relatively high internal impedance of the Goldcap has a big effect on the rate of voltage rise.
To provide the charge current for the supercaps here we used a simple linear voltage regulator such as the LM317. Most regulators will limit the output current to the maximum specified value — this isn't the most efficient way of doing it but in practice it works without problem.

On the limits of measurability

The AVX SCCR12B105SRB supercap used for this test (the significantly cheaper Nichicon JUWT1105MCD was out of stock at the time of testing) was according to an LCR meter an 'Unacceptable DUT', the HP 4262A identified it as a short circuit! So to get an approximate value of capacitance it will be neces-

Table 1. Supercap and Goldcap key features		
Capacitor	SuperCap, AVX SCCR12B105SRB	GoldCap, Panasonic NF Series EECF5R5U105
Specification	2.7 V, 1 F	5.0 V, 1 F
Weight	0.94 g, Long wires (!)	7.92 g
Charge waveform I_{max} **=1 A**		

sary to make use of an old trick. The voltage across a capacitor discharging through a known value of resistance will describe a curve with a time constant $t = R \times C$; after one time constant RC the capacitor voltage falls to 63.2% of its start value V_{in}. To plug a few values in, the capacitor is first charged to 2.7 V and discharged through a 21-Ω resistor. The resulting discharge curve can be easily captured using a digital oscilloscope but we first need to think about calibration. To establish a zero reference the capacitor terminals must be shorted together for some time.

Using this setup and with a little patience it was possible to make measurements of the five supercaps under test (all from the same supplier) and the results are given in **Table 2**. Interestingly it is fairly obvious from the results that we don't need to look much further; just by using the formula $t = R \times C$ it's clear that the values of capacitance are subject to quite a wide tolerance.

Next we take a look at the issue of measuring short-circuit current. We are dealing with fairly low voltage levels and quite high levels of current so things like connections to the load must be carefully considered otherwise they could introduce an unacceptable voltage drop. For example, standard screw terminal strips have a resistance of around 1 ohm; at 2.7 V the maximum current will then be limited to less than 3 A.

A good candidate for switching the current is a relay by the Chinese manufacturer Songle. In its data sheet a maximum contact resistance is specified of around 0.1 Ω. This resistance is not particularly constant and sputtering can occur when high current are switched which increases contact resistance. We will also require a sensing element in the circuit and for this a power resistor (0.22 Ω; 10 W) will be used in series. Now with the capacitor charged we retire to a safe distance and activate the relay. The resulting curve can be seen in **Figure 3**.

Problems with series connections

Capacitors with precise values of capacitance are expensive. While the tolerance of a capacitor used as a charge reservoir in a mains adapter unit is not at all important, for supercaps it is extremely important.

Table 2. Results of five Supercaps from the same supplier (in an RC network).

	RC time, 63.2% of V_{in}
Sample 1	27.6 s
Sample 2	29.9 s
Sample 3	28.4 s
Sample 4	29.5 s
Sample 5	29.6 s

Why is it so important to have an exact value of capacitance? In practice supercaps are rarely used singularly, almost always several are connected in series. This is because of their low maximum voltage rating: the type used here has a maximum rating of 2.7 V. When several are connected in series it forms a capacitive voltage divider. Capacitors with different values of capacitance will produce different values of final charge voltage across each capacitor when they are charged in series. If you find this difficult to believe try measuring the voltage across two different value electrolytics in series plugged into a bench power supply.

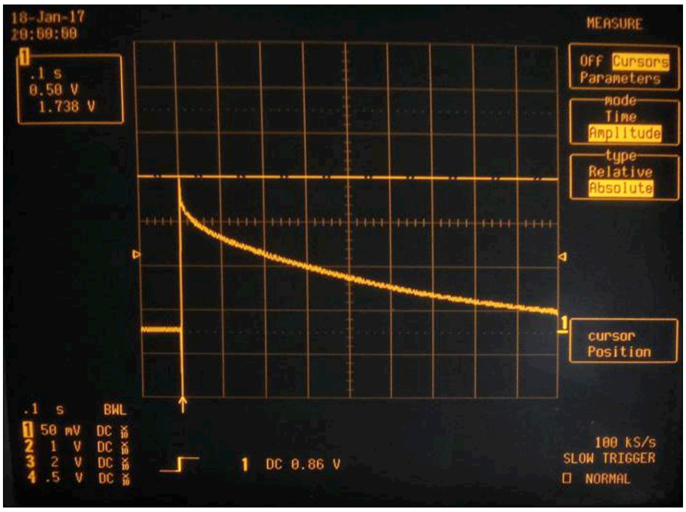


Figure 3. The discharge current waveform has a peak value of 7.9 A, displayed on a LeCroy scope.

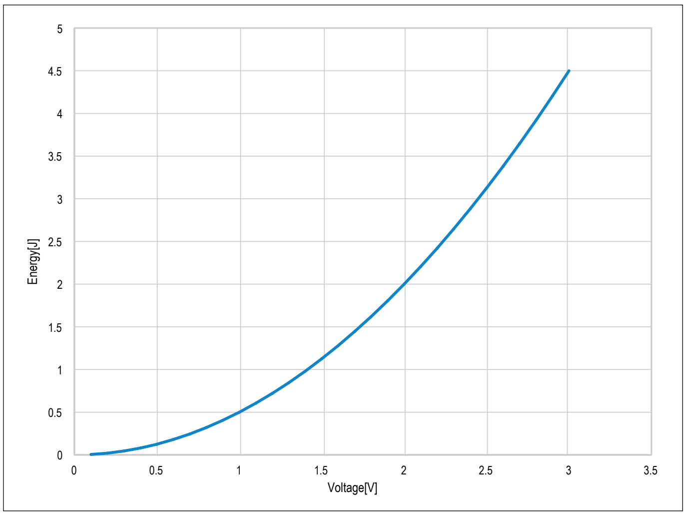


Figure 4. Energy stored and voltage has a quadratic relationship.

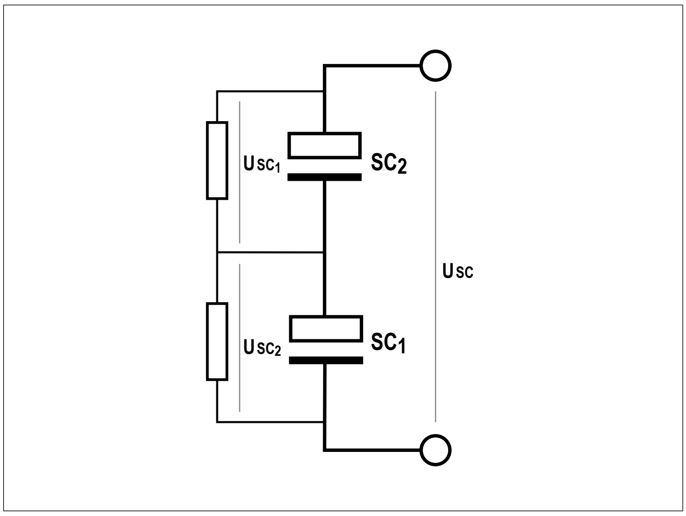


Figure 5. Two resistors tame as well as protect the supercaps.

Table 3. The ALD8100XX series from Advanced Linear Devices.

Part No	Threshold Voltage		Part No	Threshold Voltage
ALD810016	1.6 V		ALD810023	2.3 V
ALD810017	1.7 V		ALD810024	2.4 V
ALD810018	1.8 V		ALD810025	2.5 V
ALD810019	1.9 V		ALD810026	2.6 V
ALD810020	2.0 V		ALD810027	2.7 V
ALD810021	2.1 V		ALD810028	2.8 V
ALD810022	2.2 V			

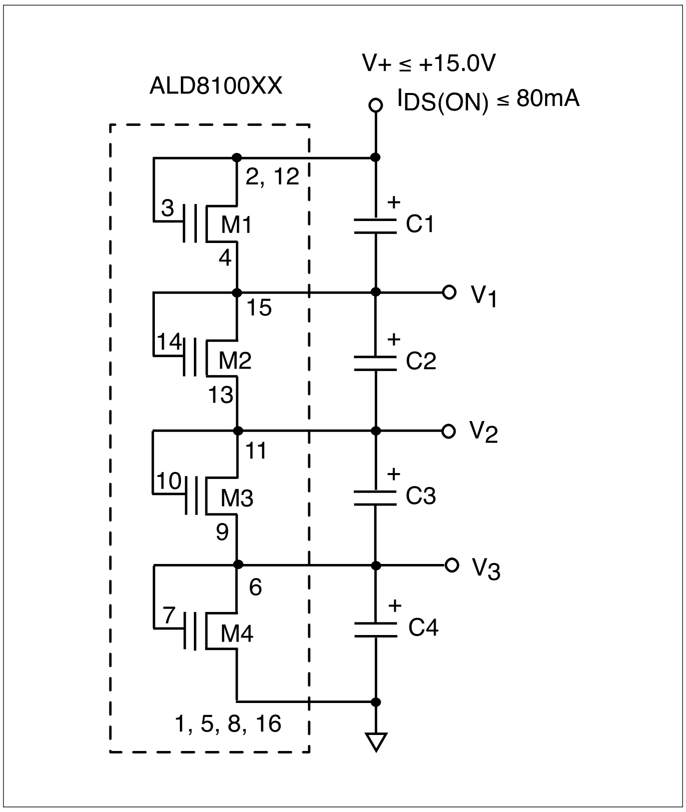


Figure 6. These supercaps are balanced.

Unlike conventional electrolytic capacitors, the internal structure of Supercaps makes them extremely sensitive to overvoltage: with a difference of ten percent you have a choice of reducing the cell voltage and accepting a lower energy storage capability or risking damage to the supercaps. Reducing the supercap maximum voltage has a dramatic reduction on the energy stored (**Figure 4**).

In practice overvoltage stress in a series-connected supercap leads to a progressive decay which is difficult to characterize without more detailed lab tests; usually the capacitor either functions or *Ka-Boom*! The researchers Linzen, Buller, Karden and De Doncker have devised a formula [1], relating capacitor operating voltage to lifetime and although it shows an exponential relationship the units used are in years.

Balancing circuits can be used to compensate for the difference in capacitance between supercaps connected in series. The circuit can be made up of one or more components which in one way or another equal out the voltage across each capacitor and prevent over charging any individual capacitor. The most basic form of protection is provided by two equal-value resistors (ideally, matched) arranged as shown in **Figure 5** connected in parallel across each supercap.

The role of the resistors is simple; the junctions of the resistors form constant voltage points so that the supercaps will all have the same voltage across them (assuming the resistors are well matched). The disadvantage of this simple solution is that the resistors are permanently connected even when the capacitors are charged and therefore dissipate energy. Depending on the resistor values this will be sometimes more, sometimes less important but nevertheless it is not a very elegant solution.

Diab, Venet und Rojat describe in [2] various active balancing methods including one using a series of Zener diodes to dissipate excess voltage developed across any supercap. The zeners will not draw current when the supercaps are charged, others configurations disconnect the capacitors from the charge current to save energy.

Let's add some components!

The American company Advanced Linear Devices caused a sensation in the supercap world when it introduced the ALD series of devices a few months ago. These are ICs containing two or four FETs which become conductive above a certain voltage threshold (in the same way the zener diodes work above) and limit the voltage across each supercap to a certain level. The required maximum voltage level is defined by the type number of the IC. **Table 3** lists the family of ALD8100XX series devices (see [3] for datasheet) and their characteristics (valid at the time of going to press).

It would seem these ALD series of ICs offer the ideal solution; we just need to wire them in parallel to the capacitors as shown in **Figure 6** and bingo, a perfectly balanced network of capacitors. On reading the data sheet however there are some extreme situations that deserve further consideration. Beginning with a series-network of supercaps seriously out of balance because of manufacturing tolerances. We could take the more extreme case where one supercap is completely discharged and the second is fully charged – in this case the first supercap will be charged with a very high level of current in excess of the maximum power rating of the chip (500 mW) which would quickly result in its destruction.

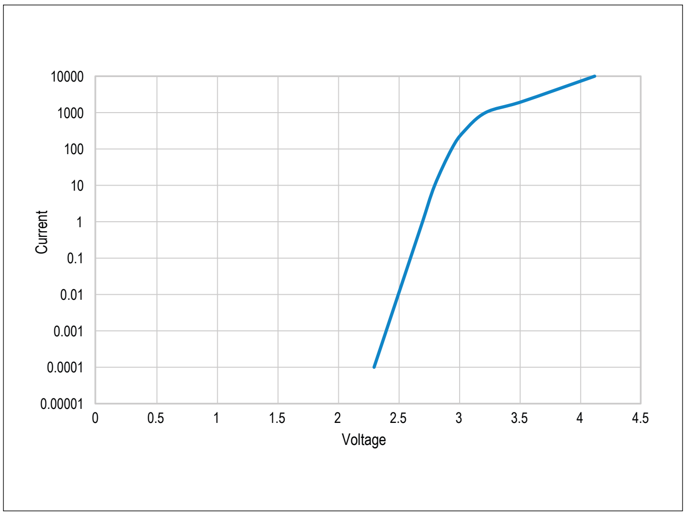


Figure 7. The maximum current flowing through the IC (here an ALD810027) is dependent on the voltage.

As we said this is extreme situation and according to the data sheet from Advanced Linear Devices this operating condition is forbidden. Their recommendation is that the supercaps are all from the same manufacturing batch so that their characteristics are closely matched — the diagram in **Figure 7** shows why this is necessary. Due to the chip's limited maximum current (the Y axis is logarithmic), in the case of massively out of balance supercaps, the IC will quickly give up the ghost and the electrolyte will get zapped before a sufficiently high voltage can be achieved to allow the charging current to decrease (exponentially) to a safe level. By the end of the charging process both capacitors will be at same potential and consequently the balancing currents will be smallest.

In addition to the Linear Technologies chips the semiconductor industry has in the meantime also come up with some alternative chips for use with supercaps. Linear Technologies are the market leader but **Table 4** shows some interesting candidates.

Conclusion

Supercaps are not the average sort of component you are likely to use for general purpose applications. Measuring their properties is an interesting exercise in itself when you find that the component can't be characterized by standard test equipment. For some specific applications such as military equipment design, metalworking and robotics, there is sometimes little alternative, in this case you could say that supercaps are in fact the Gold standard. ⏮

(160098)

Table 4. Alternative Supercap balancer ICs.

IC	Manufacturer	Package	Description
XRP6840	Exar	TQFN	Charge pump for LED flash unit
bq25505	Texas Instruments	VQFN	Energy Harvesting-IC charger
bq24640	Texas Instruments	VQFN	Charge controller, switches between CC and CV.
LTC3225	Linear Technologies	DFN	Low-noise 2-cell charger programmable charge current up to 150 mA.
LTC3350	Linear Technologies	QFN	Step-down 1 to 4 cell plus reverse step-up backup supply.
LTC4425	Linear Technologies	MSOP	Linear 2-cell charger with current limited ideal diode and V/I monitor.

Web Links

[1] Dirk Linzen, Stephan Buller, Eckhard Karden, Rik W. De Doncker, Analysis and evaluation of charge-balancing circuits on performance, reliability, and lifetime of supercapacitor systems, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, 2005, 41(5), 1135-1141, www2.isea.rwth-aachen.de/dataint/alumni/buller

[2] Comparison of the Different Circuits Used for Balancing the Voltage of Supercapacitors: Studying Performance and Lifetime of Supercapacitors, https://hal.archives-ouvertes.fr/hal-00411482/document

[3] Datasheet ALD8100XX-Serie from Advanced Linear Devices, www.aldinc.com/pdf/ALD8100xxFamily.pdf

Powerbank Surprise

By Luc Lemmens (Elektor Labs)

I spotted them for the first time during the Peltier lamp project ('Candle2light', September & October 2017, [1]): these cute little USB LED spotlights from a certain Swedish retailer that we don't need to name. Not that I knew of an immediate application for these, but it is one of those things that continue to stick around in my head. Some time later, in one of the local discount stores, I came across a Li-ion powerbank that had modest dimensions. These are actually intended for an emergency charge infusion of your tablet or mobile phone when you're away from home or when you have forgotten to charge them. But here I recognised these immediately as great power supplies and mounting bases for those USB lamps. "A small, wireless desk/read/bed lamp will certainly come in handy one day", was the thought. And for those few euros it's a bargain… With the packaging promising a capacity of 2000 mAh, such a power LED with a measured current consumption of 65 mA, should run a good 30 hours.

I unpacked the powerbank immediately when I got home and connected a lamp to it, and as expected, it worked perfectly right away. Despite the fact that such powerbanks are not fully charged when bought from the shop, the light remained on for several hours. The experiment was a success, nothing

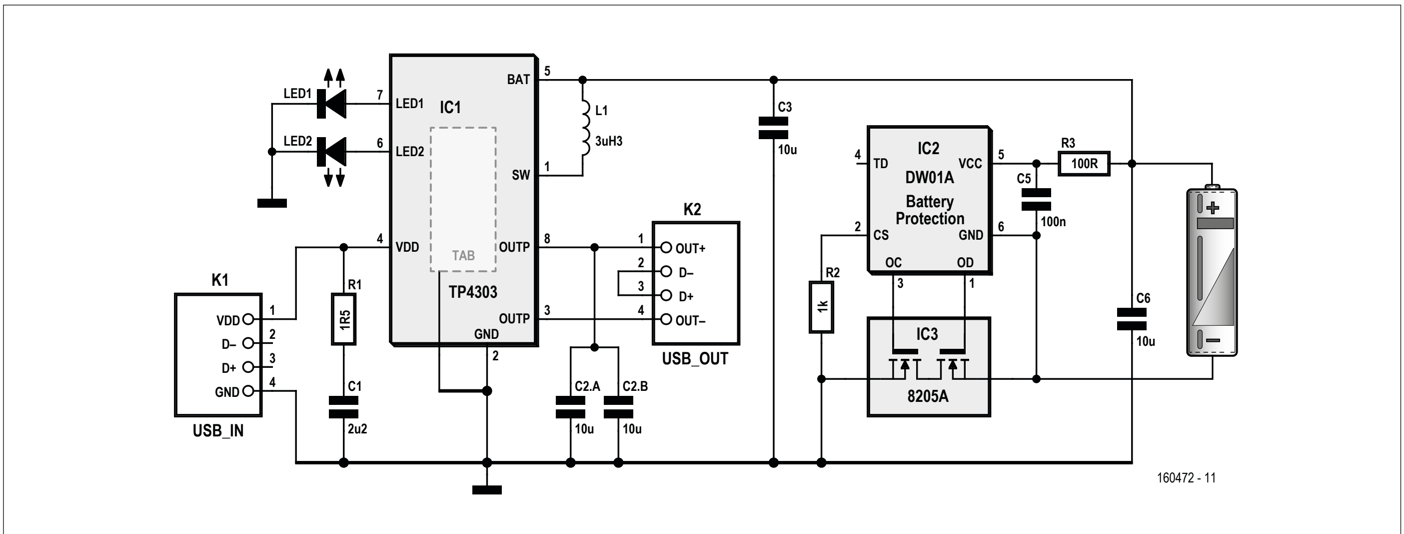


Figure 1. The supposed schematic for the electronics in the powerbank.

particularly startling or anything to brag about, but pleasing nevertheless. The powerbank was subsequently completely charged using the supplied USB cable, to test how long the light would last on a fully charged powerbank. The answer to that question was both surprising and disappointing: about ten seconds!

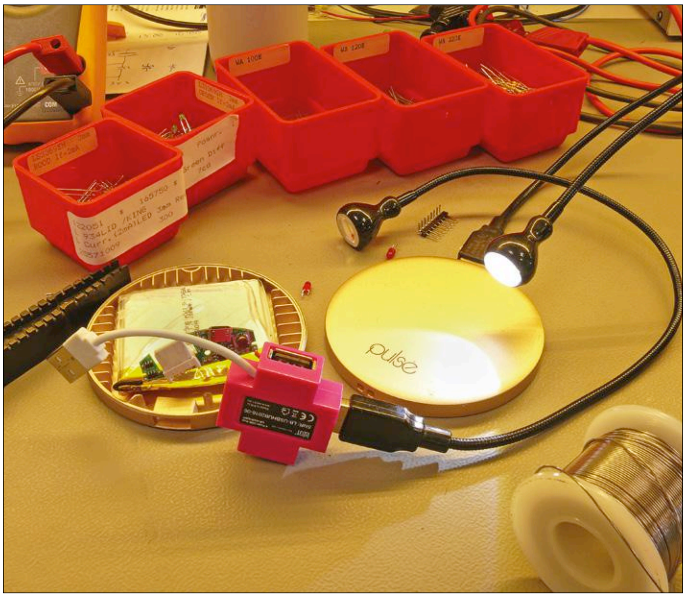
Naturally, I first checked whether the powerbank was actually charged up, and when that turned out to be so, I realised that in that small enclosure there had to be a little more than just a battery and some electronics that ensures that it is properly (and safely?) charged. The very succinct manual — yes, sometimes you will resort to reading those — reports that the powerbank turns off automatically after a few seconds “as soon as the smartphone has been completely charged”. From this it would be fair to conclude that the powerbank turns off when the output current drops below some threshold. Apparently the lamp does not draw enough current and therefore the powerbank calls it quits. And indeed, when two lamps are connected to the same powerbank via a USB splitter cable, the lights remain on. This all sounds logical and convincing. Although… why did it work the first time when it came new from the packaging? It appears to be related to a (too?) fully charged battery, because after discharging for a while it does continue to operate with only one lamp, that is, just like it did in the beginning.

Of course I wanted to find out what was going on here and decided to open the enclosure to see what was inside. This was actually quite a risky job: the metal lid was solidly glued to the Li-ion battery and it can be dangerous if this is damaged or shorted (“Don’t try this at home!”). Fortunately the operation succeeded without accidents. In addition to the battery, there was a small circuit board that contained more electronics than I expected: two LEDs, a number of resistors, capacitors and even a small coil and three small ICs with the markings TP4303, DW01 and 8205A. A brief search on the internet for these part numbers resulted in a schematic that is almost certainly the correct one ([2], **Figure 1**). Not that I put much effort in checking whether all connections and component values are in fact correct, but I don’t expect that much time and money is spent on special electronics for these mass-produced items.

The part around the DW01A and 8205A is clear. This circuit protect the battery from over-charging and deep discharging and offers protection against (too) high currents. The datasheets are easily obtained from the internet ([3] and [4]), and offer sufficient information to show that these are not used for switching-off when the output current is too low. In addition there is no functionality to be found that only operates after a few seconds. So therefore this has to happen in the third IC, the TP4303.

The standard application schematic that is shown in the datasheet for the TP4303 makes it clear what this IC does: it is a step-down regulator for charging of the battery, and at the same time a step-up converter for the output of the powerbank. The Chinese text from the datasheet ([5], I have been unable to find an English version) comes out surprisingly readable from Google Translate. This indeed states that the IC switches off after 16 seconds when the load is disconnected or when the discharge current is too low, but unfortunately it does not specify the current level of this threshold. Strangely enough, the TP4303 also offers several protection mechanisms, including excessive current, deep discharging and battery overvoltage, but the datasheet also mentions that additional protection with a DW01 — as in this powerbank — is also possible. To be honest, this sounds like TPower themselves have reason to doubt the protection that the TP4303 has to offer. And maybe that is where the problem lies: multiple protections that get in each other’s way? It appears that the electronics, after fully charging and with a small load, detects battery overvoltage and that a larger discharge current is just sufficient to reduce the battery voltage below this safety threshold. Once the battery has discharged a little the powerbank does remain on when only one light is connected.

Don’t try this at home



I have to admit that by now my enthusiasm to dig further had dissipated. The explanation for this phenomenon is plausible, but I didn’t continue to measure or check to make sure that I have, in fact, got it right. It can’t be called a design fault. If you use the powerbank for charging a phone or tablet it will work as intended. But I was surprised when my LED lamp turned off so quickly.

(160472)

Web Links

[1] www.elektormagazine.com/160441

[2] www.alexanderb.tk/index.php/2017/02/06/hacking-a-powerbank-to-use-as-a-pi-ups

[3] www.haoyuelectronics.com/Attachment/TP4056-modules/DW01-P_DataSheet_V10.pdf

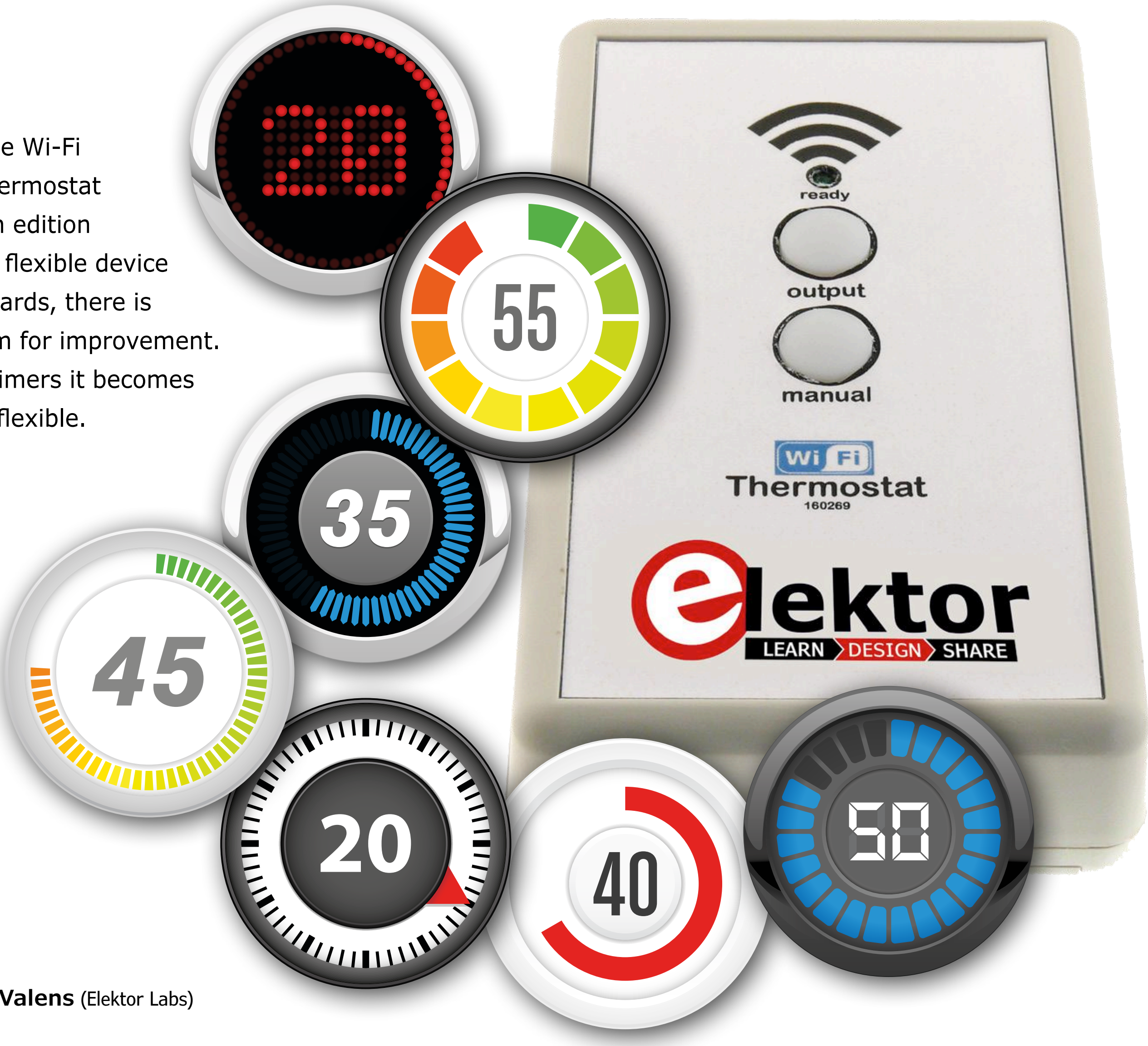
[4] www.maritex.com.pl/product/attachment/91261/8205A.pdf

[5] www.datasheetspdf.com/pdf/949049/TPOWER/TP4303/1

Timers for the Wi-Fi Desktop Thermostat

Seven channels with atomic accuracy

Although the Wi-Fi Desktop Thermostat published in edition 1/2018 is a flexible device by all standards, there is always room for improvement. By adding timers it becomes even more flexible.



By **Clemens Valens** (Elektor Labs)

Features

- New firmware adds seven timers
- Workday & Weekend mode
- Up to seven single-shot events anytime in the future
- Up to seven daily events
- Temperature sensor can be disabled

Indeed, why limit the Wi-Fi Desktop Thermostat [1] to the desktop? As it is, the device is already capable of controlling the heating system in your home but if we add some features to it, it will be even better suited for this task. For instance, a timer, and especially one that can be programmed for every day of the week, is a great extension for controlling the temperature in your home. Besides adding timers we will also add the possibility to disable the temperature sensor to allow the thermostat to act as a pure timer with Internet connectivity.

Get the time from the Internet

Barring egg timers and stopwatches, a timer needs to know the time.

However, looking at the schematic of the thermostat you will not find a real-time clock or any other time-keeping device. Of course, it is possible to add one, but in this era of Internet-connected devices it's no longer necessary to equip them with clocks since the exact time is freely available on the net. All we need to do is add Network Time Protocol (NTP) support to the software. A few years ago this was a lot of work, but since our thermostat software is built in the Arduino environment, adding NTP support (**Figure 1**) is nothing more than copy-pasting an Arduino example into our code and adapt it a little.

Seven timers

Once the application knows the time, it can compare it to a preset time and activate (or deactivate) the thermostat's output. However, considering that the user interface (UI) of the thermostat is a webpage, and because our ESP-12F module has lots of memory left, there is no reason to limit ourselves to just one preset. Like adding NTP support, creating multiple presets is just a little more work than copying and pasting the code for a single preset. For practical reasons we have limited the number of presets to seven.

Timer modes

Each timer can be assigned to a day of the week, and for each timer the start and stop time can be set (**Figure 2**). For a week this means setting up seven timers. However, if your working week is from Monday through Friday and your weekends are Saturday and Sunday, and if you can live with one preset for the working week and one for the weekend, then you only have to set up two timers: one as 'Working Week' (or workweek) and the other as 'Weekend'. For cases where the thermostat has to repeat the same thing every day, a timer can be set to 'Every day'. Finally, by setting the start and stop date and time with a calendar control (mode 'Once', **Table 1**) timers can be programmed to activate sometime in the future with a programmable duration.

Up to seven timers can be programmed and as they run in parallel, they may overlap. Setting several timers to activate on the same day is an option too, as is programming up to seven future events. All in all, the timers are pretty flexible.

More than a thermostat

Being a thermostat its output will only be switched on when the measured temperature satisfies certain criteria as

PROJECT INFORMATION

Wi-Fi | thermostat | ESP8266 | timer | NTP

entry level
→ intermediate level
expert level

1 hour approx.

Arduino IDE,
serial-to-USB converter

€0 / £0 / $0

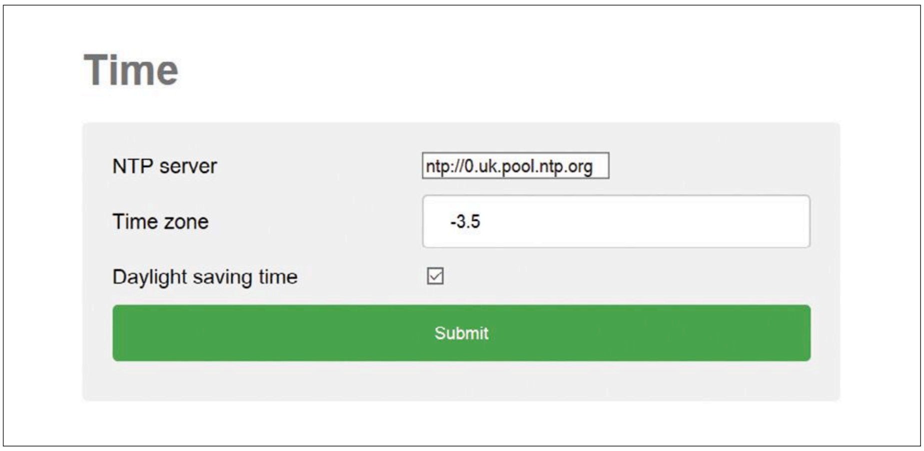


Figure 1. The new user interface section 'Time' is required to set up the NTP server. Note that daylight saving time (DST) must be set manually as it is not easily available from the Internet.

Table 1. Eleven modes make the timers flexible.

Nr	Mode
0	Once
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday
8	Every day
9	Workday (Mon, Tue, Wed, Thu & Fri)
10	Weekend (Sat & Sun)

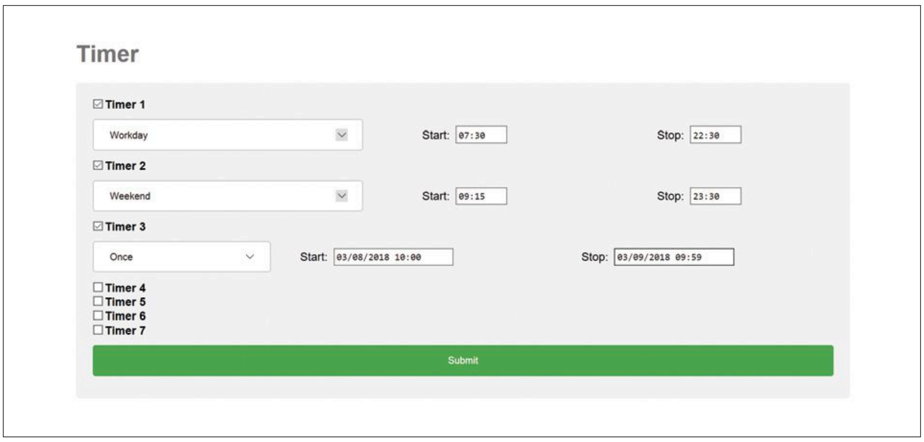


Figure 2. Three timers marked as active and programmed for workday-weekend use together with a one-month single-shot.

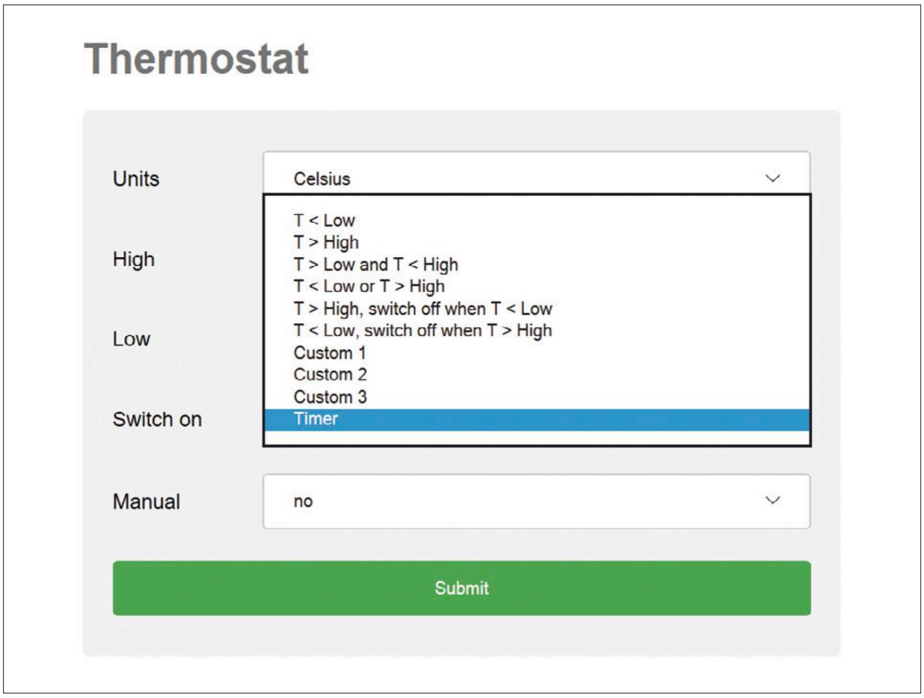


Figure 3. Selecting 'Timer' as 'Switch on' mode will deactivate thermostat mode and turn the device into a pure timer.

defined by the value in the 'Switch on' field of the 'Thermostat' section of the UI. By adding the value 'Timer' (**Figure 3**), the thermostat can be told to ignore the temperature sensor and let its output control by the timers only. This is a great mode for switching lights and window blinds while you are on holidays or to control Christmas decorations. Surely you can come up with other applications too.

Don't repeat yourself

Let's have a closer look at the way the timers are implemented. As you may recall from [1] the UI is written in HTML, JavaScript, and CSS (and talks to an application written in C++). Adding seven timers to it in a naïve way would mean adding almost identical code seven times. In software engineering this is something that is to be avoided at all times as it is a great way to introduce bugs and create hard-to-maintain code. This programming principle, known as "don't repeat yourself" or DRY, is difficult to respect when it comes to HTML, which is why all sorts of wrappers and add-ons have been invented for it like JavaScript and stylesheets. We have therefore defined our timer in JavaScript and put seven placeholders in the HTML page. Only when a timer is marked for use (its checkbox is checked) the HTML code for it is generated and inserted into the page where it behaves like if it had always been there. Consequently, timer customization has to be done in the JavaScript file, while adding or removing timers is done in the HTML file. This also implies that modifications affect all timers in the same way.

The JavaScript code — at the end of the file script.js — is not very involved as it mainly produces HTML code (see the function `timerBuild`) adapted for each timer so that every timer uses unique variable names. This, of course, is necessary for the C++ application to keep them apart. The only complication in the code is the way the start and stop time boxes are changed according to the selected type of the timer. A future event ('Once') needs date and time whereas daily, 'Workday' and 'Weekend' events only need the time. The function `timerChange` takes care of this. Inserting the HTML code into the UI is done by the function `timerShow`.

Updating the thermostat

Compiling the new software and programming the ESP-12F is described in detail in [1]. The only thing not to overlook is to replace the old thermostat files by the new files you can download from [2].

A note on Internet browsers

The HTML code presented has been verified with the W3C Markup Validation Service [3] and was found free of errors and warnings, except for the date/time pickers that are not supported by every Internet browser on the market. It worked fine on our test computer with Google Chrome (version 63.0.3239.84) and Microsoft Edge (version 41.16299.15.0). I failed to get the UI to work in Internet Explorer 11 (11.125.16299.0), even after trying all sorts of solutions found on the net. I therefore do not recommend this browser. If you have a suggestion to fix this problem, please let me know.

Keep in mind that the thermostat's UI uses JavaScript and might conflict with your browser's security settings.

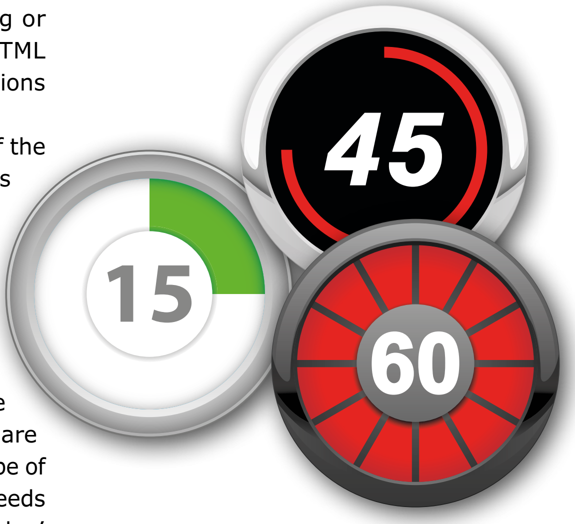
(160631)

Web Links

[1] Wi-Fi Desktop Thermostat: www.elektormagazine.com/160269

[2] Article support page: www.elektormagazine.com/160631

[3] W3C HTML validator: https://validator.w3.org/



A Simple Digital Audio Amplifier

Using a CMOS IC in an analog circuit

By **Hans-Norbert Gerbig** (Germany) (based on an idea from the Elektor book *301 Circuits*)

Analog circuits have had their day; everything is digital now. Here we present a simple digital amplifier built with standard CMOS logic gates, which digitizes the analog audio signal and then converts it back to analog for output to the loudspeaker. Along with being interesting, this project can be used as the basis for countless experiments.

Digital Mini Audio Amplifier

- Single supply voltage +4.5 – 6 V
- Output power <1 W
- Gain ≤45, continuously adjustable
- PWM generator with 4049 or 4069 IC
- Complementary output stage with BS170/BS250

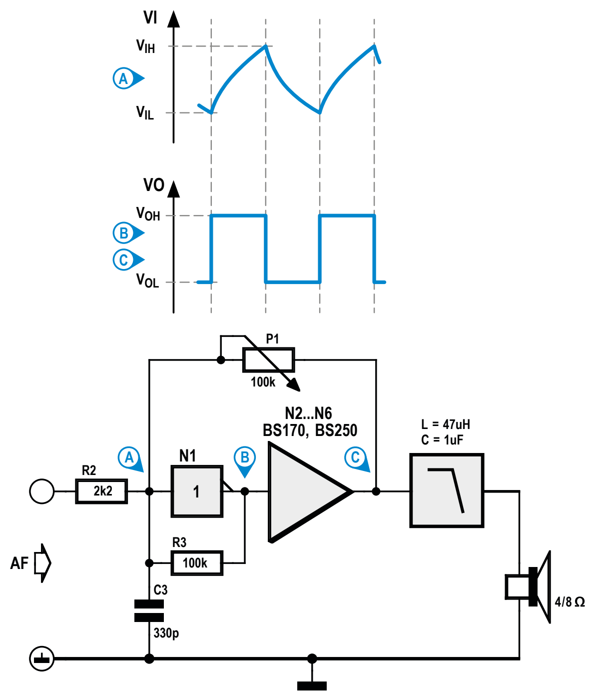


Figure 1. A digital switch-mode amplifier built around logic gates.

The core element of the circuit is a square-wave oscillator consisting of an inverting logic gate N1 and an RC network (see **Figure 1**). When power is switched on, capacitor C3 is not yet charged, so the input of the amplifier is in the Low state and the output is therefore in the High state. The output voltage charges the capacitor through resistor R3 until the upper switching threshold of the gate is reached, at which point the output switches to Low. Now the capacitor is discharged through the resistor until the voltage reaches the lower switching threshold, causing the gate to change back to its original state and start the next cycle.

The voltage waveforms at the input and output are also shown in Figure 1. From them you can see that the input signal is virtually triangular because the charge/discharge characteristic of the capacitor is nearly linear in the region concerned. No matter which IC technology or logic family you choose, the output power of a simple inverter is far too little to drive a low-impedance loudspeaker, so the oscillator output is boosted by a buffer. It consists of the remaining inverters in the IC package wired in parallel, followed by an power stage formed by a pair of complementary discrete MOSFETs. In this way the signal from the oscillator is amplified twice, but it still has the same waveform at the output of the buffer. As you can see from **Figure 2**, this

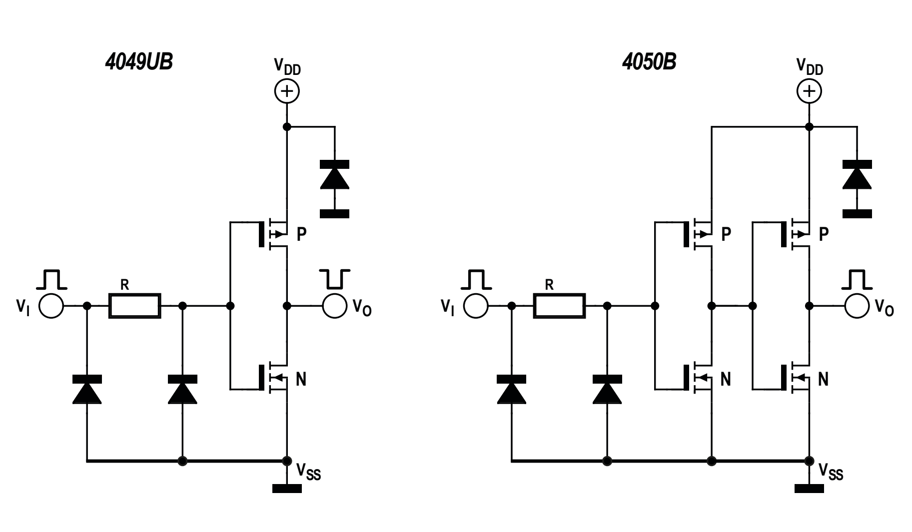


Figure 2. The combination of the inverter gates of the 4049 IC and the buffer stage corresponds to the internal circuitry of a non-inverting buffered 4050 IC (source: National Semiconductor data sheet).

arrangement is equivalent to the internal circuit of a non-inverting buffered gate. Although the BS170/BS250 transistors chosen here have only modest power ratings, they can still source 0.5 A and sink 0.18 A, which is in a totally different ballpark from the output capacity of a logic gate.

The frequency of the oscillator is determined by the RC network according to the familiar formula f = 1/(2RC). The frequency with the indicated component values is a bit more than 15 kHz, which is obviously not high enough for audio reproduction because it would be audible from the loudspeaker output for people with good ears. However, the signal at the output of the MOSFET buffer is fed back to the oscillator input through a potentiometer. This puts the potentiometer more or less in parallel with resistor R3 and raises the oscillator frequency to 30 kHz when the resistance of the potentiometer is 100 kΩ. It rises even higher when the resistance of the potentiometer is reduced.

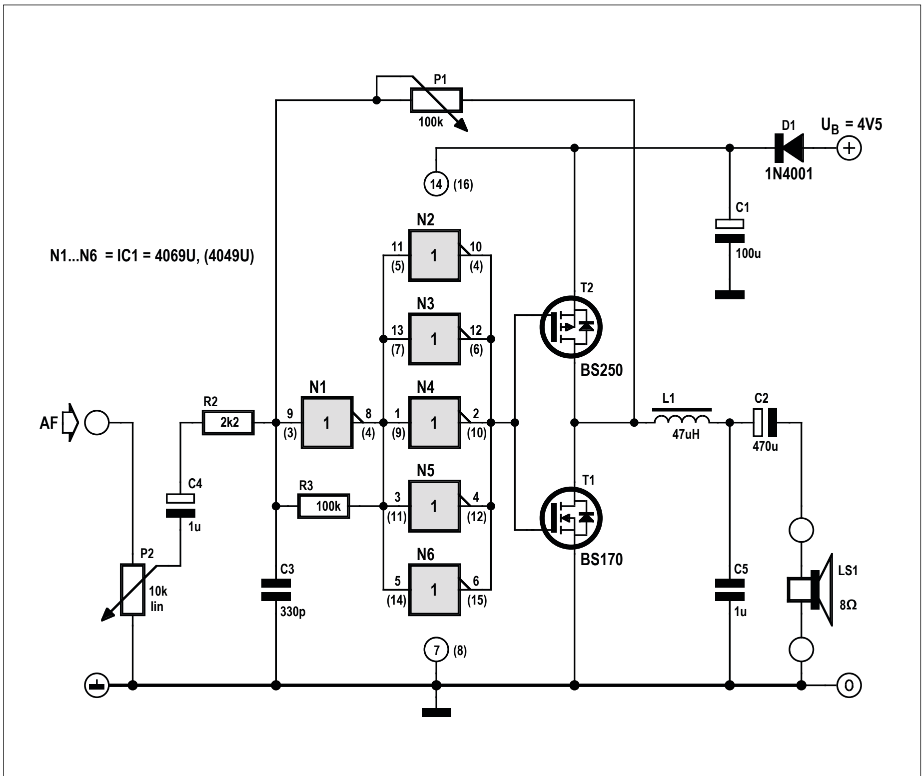


Figure 3. Circuit diagram of the simple digital audio amplifier.

In the analog world

So far we have not considered how the audio signal enters the circuit, what it does in the circuit, and how we can recover an analog signal at the output of the circuit. As can be seen in **Figure 3**, the audio signal is to the input of the oscillator gate N1 through a potentiometer and coupling capacitor C4, which blocks any DC voltage on the input signal.

The instantaneous level of the audio signal affects the charge and discharge rates of capacitor C3. When the level is relatively high, the voltage on the capacitor reaches the upper switching threshold faster and the High portion of the pulse waveform is correspondingly longer. At the same time, the audio signal level slows the discharge rate of the capacitor. As a result, the voltage takes longer to reach the lower switching threshold and the Low portion of the pulse waveform is shorter. However, the fundamental frequency remains the same. In this way the audio signal modulates the pulse waveform to create a form of pulse duration modulation (PDM) or pulse width modulation (PWM).

Just as the audio signal in an FM transmitter modulates the frequency of the sinusoidal waveform while maintaining constant amplitude, here the audio signal modulates the duty cycle with constant frequency and constant amplitude (10% and 90% of the supply voltage) of the pulse waveform.

In an FM transmitter modulation stage, noise spikes on the signal amplitude can easily be eliminated by clipping them with a limiter. The pulse waveform generated here is essentially the same as a PWM signal. The advantage of frequency modulation and demodulation is that it enables noise-free audio transmission, as we all know from FM radio broadcasts. With amplitude modulation and demodulation (as used for broadcasting in the long-wave, medium-wave and shortwave bands), noise cannot be eliminated this way, only attenuated. Noise voltages (for example, from atmospheric noise) can easily alter the amplitude of the signal, and they cannot eliminated by simple clipping because that would adversely affect the demodulation process.

On the other side

Loudspeakers cannot work directly with digital signals. The digital signal is therefore converted back to analog at the output of the circuit. This is done by an integrating network (low-pass filter) consisting of L1 (47 µH) and C5 (1 µF), which effectively averages the value of the pulse waveform. The output electrolytic capacitor C2 (470 µF) blocks the DC voltage at the amplifier output, so the loudspeaker is only driven by the amplified input signal. To be on the safe side — which means to avoid high voltages at the output when no loudspeaker is connected — it is advisable to add a 33 Ω resistor in parallel with the speaker terminals.

The maximum output power is less than 1 W with a supply voltage of 6 V, and the quiescent current is remarkably low at less than 4 mA. The volume can be set to any desired level with potentiometer P2. Potentiometer P1 should be adjusted for the lowest supply current level with undistorted sound from the loudspeaker This digital audio amplifier can replace analog amplifier ICs such as the TBA820, LM386(L), TDA7052 or TDA7233.

The author has designed different PCB layouts for the 4049U and 4069UB ICs, using Lochmaster for a stripboard layout and Sprint Layout for a conventional PCB. The files are included in the free download for this project [1].

(160339-I)

Web Links

[1] www.elektormagazine.com/160339

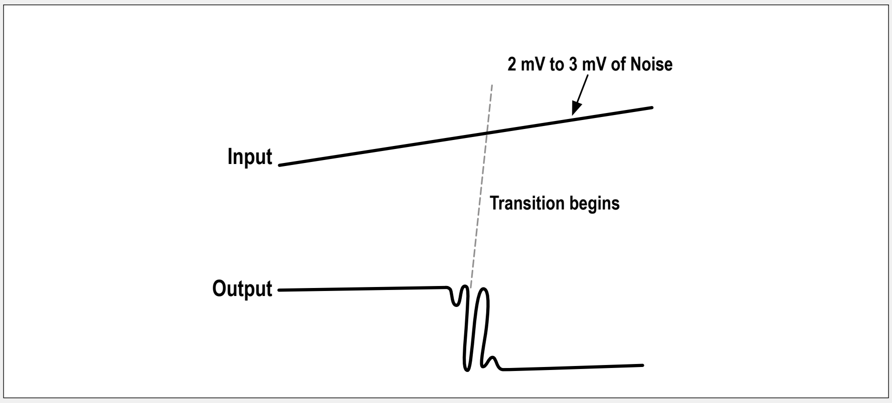
[2] www.ti.com/lit/an/scha004/scha004.pdf

Why unbuffered?

You may have noticed that the circuit calls for a particular sort of CMOS IC with a “U” in its type designation. This “U” stands for “unbuffered” and refers to the configuration of the output stage of this logic IC. The input signal to N1 (leaving the audio signal out of the picture for now) does not have the nice steep edges you are used to seeing with digital signals. Instead, it rises or falls gradually according to the charge/discharge characteristic of the RC network. As a result, the two MOSFETS at the output of the IC remain conducting at the same time (in a sort of short-circuit condition) for longer than with steep signal edges. This causes periodic current spikes which create noise in the output signal and significantly increase the current consumption of the circuit, which is clearly undesirable with battery powered operation.

There is also a second effect. If the input signal has a slight amount of noise superimposed on slow edges, buffered ICs produce short oscillations on the output signal in the transition region.

Both effects occur with buffered as well as unbuffered CMOS ICs, but to a much smaller degree with unbuffered types. Generally speaking, unbuffered CMOS logic ICs should be used in applications with very fast signals as well as systems with very low frequencies (and associated slow signals) and moderate amplification factors, while buffered ICs should be used in noisy environments as well as in low-speed systems and systems with high gain. Buffered logic is also preferable in applications where constant output impedance is important, such as R-2R networks for digital to analog conversion.

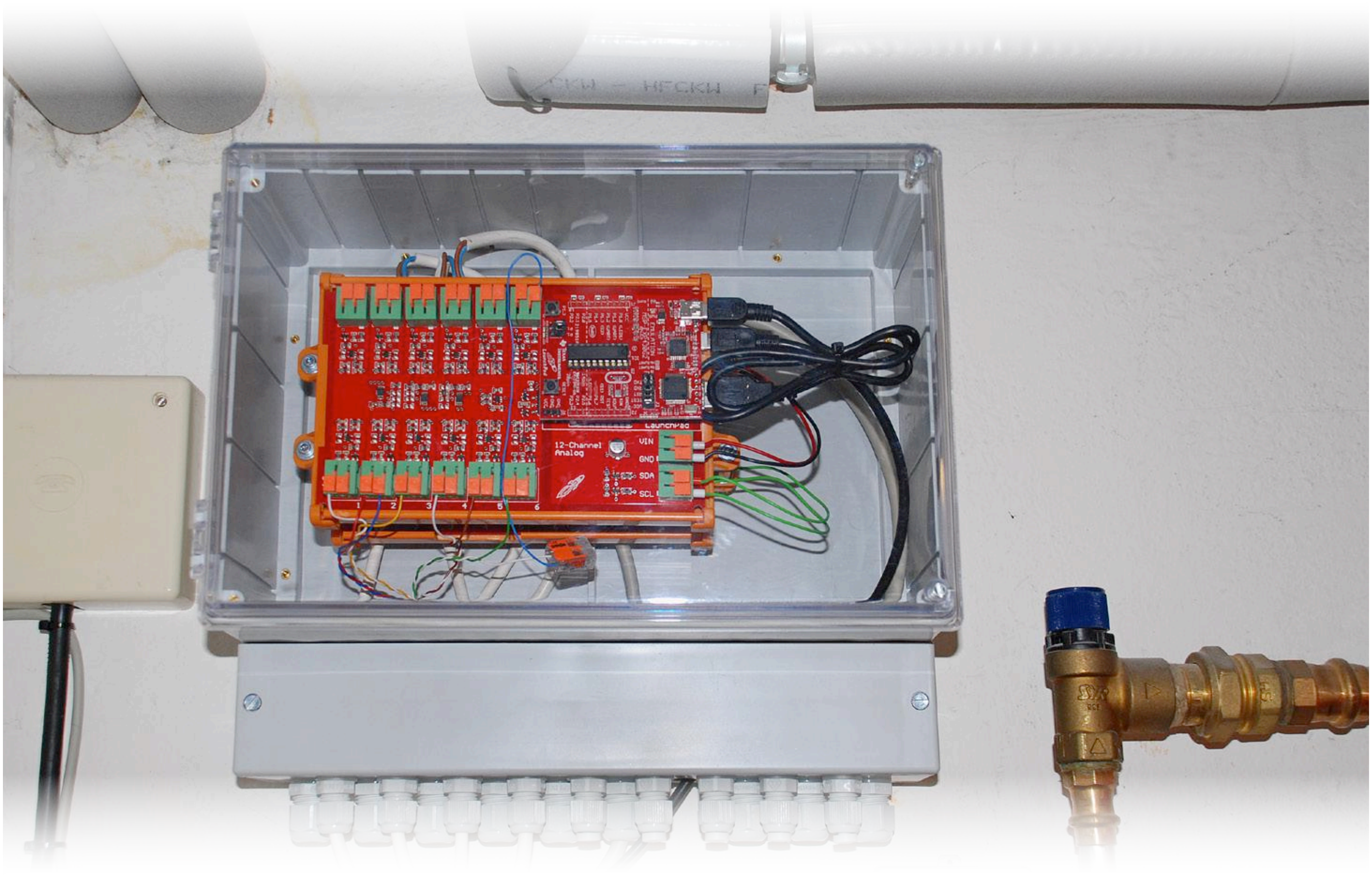


More information about this subject is available in an application report from Texas Instruments [2].

Parameter	Buffered	Unbuffered
Propagation delay	Slow	Fast
Noise sensitivity	Very high	High
Output impedance and output transition time	High and constant	Low and variable
AC gain	High	Low
AC bandwidth	Low	High
Output oscillation with slow inputs	Yes	No
Input capacitance	Low	High

Measurement Data Acquisition via USB

For heating system optimization



By **Mattias Ulmann,** Texas Instruments

If you want to optimize a heating system, you first have to record a wide variety of parameters continuously and precisely. Once you have this data, you can quickly assess the impact of even relatively small changes. The data logger described here acquires a large number of analog and digital parameters and outputs them — electrically isolated for safety — over a PC interface.

Modern heating systems have a large number of configuration options for a bewildering array of parameters. In many cases it is not clear or not easy to determine how these parameters affect the behavior of the system. Another complicating factor is the slow response of the system, which means it takes a fairly long time to determine the optimal configuration by adjusting the various settings. In order to truly optimize the system, there is no substitute for continuously recording a wide variety of parameters, such as the water temperatures in the supply and return lines and the boiler and the operating states of the pumps, valves and possibly the burner. Armed with this data, you can quickly see the effects of even relatively small changes.

Although many heating system controllers (including older types) have a bus interface for diagnostic and configuration purposes, the bus protocol is usually proprietary and suitable software is not available to consumers. Most heating system controllers also lack a PC-compatible interface port.

The author therefore came up with the idea of directly acquiring the raw data from analog temperature sensors in the heating system and the states of the connected loads in the heating system. A variety of instrumentation modules and plug-in cards are commercially available for this purpose, but they have only a few channels and/or are very expensive. The author therefore decided to develop a measuring system which can read in analog and digital signals, process them, and send them to a PC over a USB link.

Even if you think your heating system is already working as well as it can or you do not have a complicated heating system, you may find this article useful because the basic concept is versatile. It shows how a large number of digital and analog signals of any desired type can be acquired at low cost and sent via an electrically isolated interface to a microcontroller system and/or a PC for processing. For that reason, this article does not describe the system as a whole, but instead focuses on the details of its modular components.

Features

- Up to 64 digital inputs
- 12 analog inputs
- Electrically isolated sensor inputs
- Electrically isolated supply voltage
- Data processing with a low-cost LaunchPad board
- USB link to PC
- Graphic display of measurement data on PC

The concept

The measuring system needs two or three different interfaces between the input parameters and the PC.

- Several standard types of analog temperature sensor can be used (KVT20, KTY10-6, Ni1000). They are linearized and supplied with 5 V power by a series resistor. These sensors require analog instrumentation inputs with an input voltage range of 0 to 5 V.
- On the digital side the requirement is to acquire the states of the burner, the circulation pump and the solar heating pump, which can only have two states (on or off). To determine whether any of these devices is operating, its supply voltage (AC line voltage) can be measured at the corresponding terminals. Of course, suitable safety measures must be taken for this.
- There should also be additional spare channels available with an input voltage range of 5 to 24 V (DC) for future applications. To avoid potential confusion, these channels are not intended for the acquisition of analog measurement data — the objective is only to be able to acquire digital signals in this voltage range.

All channels must be read cyclically by a microcontroller, and the analog voltage readings and digital states sent to a PC via USB. Conversion and display of the measured data are to be performed on the PC.

This concept has several advantages compared to other solutions. The acquisition process is universal, and no firmware changes are necessary for other applications or using other sensors. The circuit design should focus on robustness and reliability, since the measuring system will be installed for a prolonged period and must operate smoothly and reliably over this period.

Digital inputs

There are two sorts of digital channels: one for AC line voltage (230 V_{AC}), the other for low voltage DC (5–24 V_{DC}). In both cases galvanic isolation is provided by optocouplers (type CNY17F-3), which provide safe voltage isolation. Each sort has eight digital inputs, which are connected through an I/O expander to an I²C Bus.

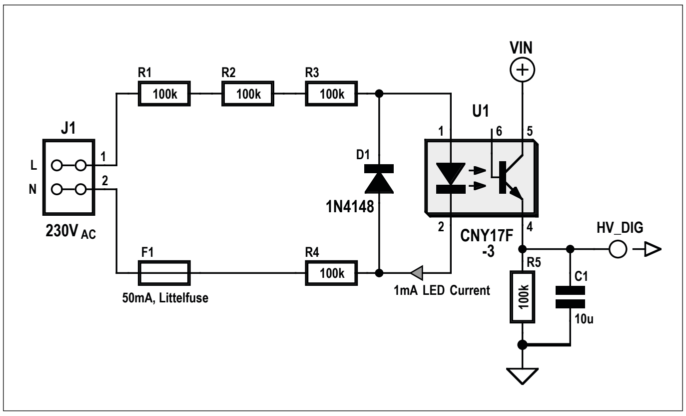


Figure 1. Schematic diagram of the digital inputs for 230 V_{AC} line voltage.

To avoid additional components or conversions, for example with transformers, the inputs for AC line voltage are designed for simple connection in parallel with the load concerned. As you can see from **Figure 1**, four 100-kΩ resistors in package size 1206 limit the current through the optocoupler to about 1 mA. Using four resistors in series divides the voltage drop over each component accordingly, ensuring that the voltage on each resistor does not exceed its maximum rated breakdown voltage.

The reverse-polarity diode in parallel with the optocoupler LED provides a current path for the negative half-cycles of the AC line voltage. This eliminates the need for a diode with a high reverse voltage rating in series with the optocoupler LED to protect it against negative half-cycles. The input is additionally protected by a self-resetting fuse with a rating of 50 mA at 250 V.

On the output side, the optocoupler passes the voltage on its VIN pin (+5 V) when a signal is present on the input. Since the optocoupler output is only active during positive half-cycles, a 10-µF capacitor is provided as a buffer to maintain a sufficient output voltage level during negative half-cycles. When a signal is no longer present on the input, the capacitor is discharged by the 100-kΩ resistor. This arrangement results in a delay of several tens of milliseconds for switch-on and several hundred milliseconds for switch-off, which is not significant for the intended application with its slow response. The optocoupler ensures reliable isolation (5000 V_{rms}) between the input and output. However, the layout must be designed to ensure compliance with the necessary and prescribed clearance and creep distances. This also applies between the individual channels.

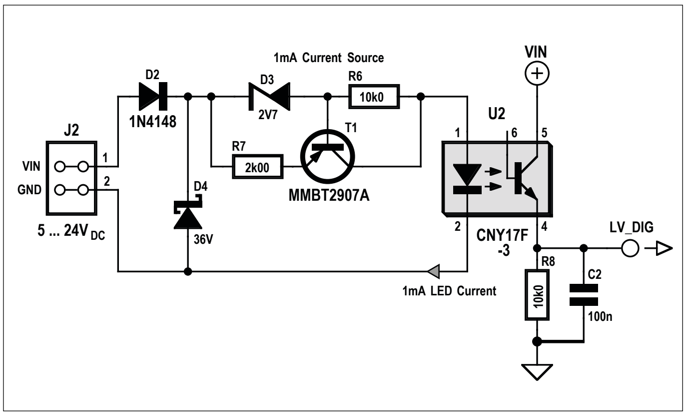


Figure 2. Schematic diagram of the digital inputs for DC voltages from 5 V to 24 V.

The circuitry of the channels for the input voltage range of 5–24 V_{DC} is a bit more complicated. Due to the extent of the input voltage range, limiting the LED current with a series resistor is not appropriate because it would vary depending on the actual input voltage. As shown in the schematic diagram of one channel in **Figure 2**, the input instead has a current source which supplies a constant 1 mA current for the LED in the optocoupler, independent of the actual input voltage. For input protection there is a reverse-polarity protection diode and a suppressor diode to limit voltage peaks. The output structure is similar to the 230 V_{AC} version, but with significantly less delay. Since these inputs are not intended for AC voltages, there is no need for a large buffer capacitor at the output to bridge over negative half-cycles.

The PCA9539 I/O expander connects the 16 digital channels to the I²C bus (see **Figure 3**). The I/O expander can be set to four different addresses with pins A0 and A1, enabling a maximum of 64 digital inputs for data acquisition. That should be enough for even the most elaborate heating systems.

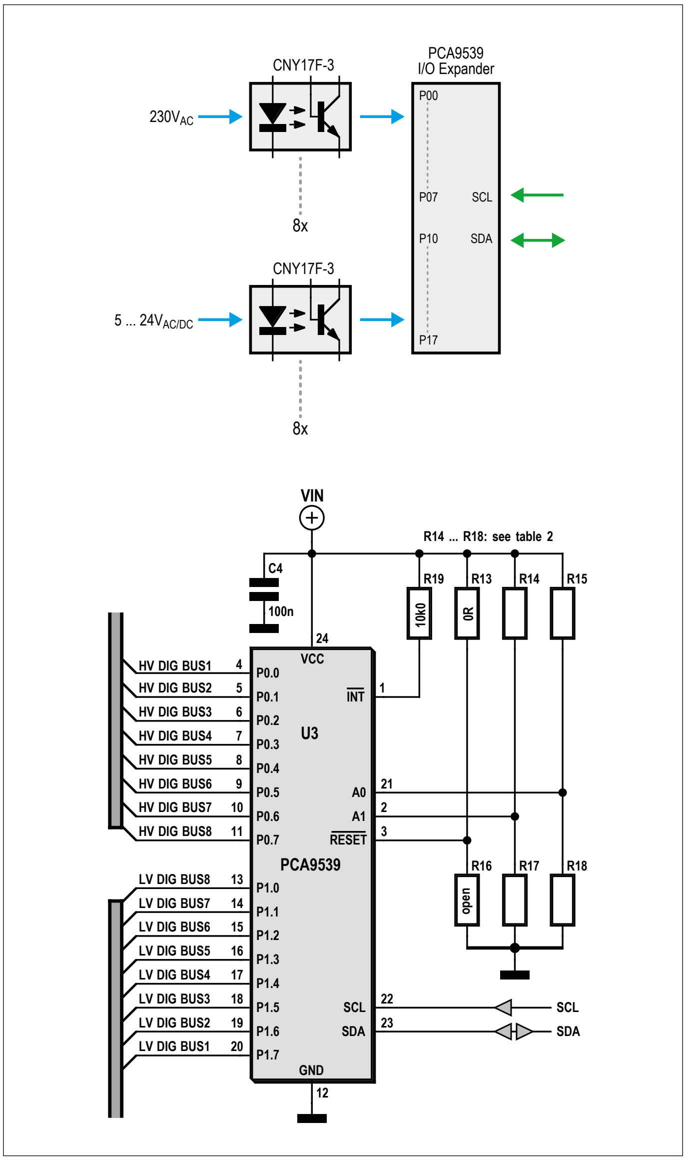


Figure 3. Grouping of the digital inputs and connection to the I²C bus.

Analog inputs

The analog inputs share a common ground and have the same structure. The 12-bit A/D converter ADS1015 with I²C interface has an input voltage range of 0 to 4.096 V and can measure four single-ended voltages sequentially. At the input of each channel there is an opamp to amplify the measured signal if necessary (see **Figure 4**).

The analog signal first passes through an RC low-pass filter (R9/C4) to suppress any high-frequency noise. The dual diode D1 after the filter clamps the input signal to the supply voltage (+5 V) or ground to protect the opamp against out-of-range input voltages. It is important that this clamping occurs *after* the low-pass filter or the input resistor, in order to limit the current through the diode. If the clamp diodes were located directly at the input, the current would not be limited and the diodes could be damaged by overheating.

A type OPA335 CMOS opamp is used here. The high impedance of its CMOS inputs minimizes the load on the sensor. The opamp only needs a single supply voltage, and its extremely low and long-term stable temperature drift of 0.05 µV/K makes it especially suitable for all forms of precision temperature measurement.

The circuit and the PCB layout are designed so that the opamps can be used either as buffers or as non-inverting gain stages. In the former case you simply omit resistor R11 and fit a wire bridge (or a zero-ohm resistor) in place of R10, and in the latter case you choose the values of R10 and R11 for the desired gain, which is equal to 1 + (R10/R11). Resistor R6 is not fitted in either case.

If neither gain nor buffering is necessary, you can omit the opamp and route the input signal directly to the A/D converter through a wire bridge (or zero-ohm resistor) in place of R6.

Circuit boards

There are basically three different types of input to be handled here, each with multiple channels, which the author has grouped onto two PCBs.

Three addressable A/D converters are located on the analog board. The ADS1015 A/D converter can be set to four different addresses (see **Figure 5**). However, a closer look at the schematic diagram reveals an interesting detail: The ADS1015 has only one address pin. How can you set four different addresses with that? The answer is that the address pin can be tied to GND or VDD as usual, but it can also be tied to the SCL or SDA lines to set two additional addresses. However, tying the

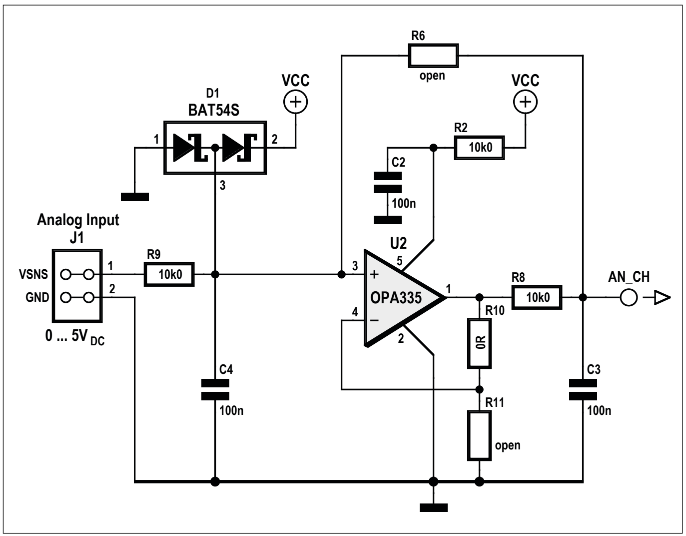


Figure 4. Schematic diagram of the protected analog inputs with optional buffering or amplification.

address pin to SDA requires a bit of extra effort in the software, which we don't really need. Consequently, only GND, VDD and the SCL line are used here for address coding. Overall that yields twelve possible analog inputs with three different addresses (see **Table 1**), which should be more than enough. Along with the analog input section, the analog board houses the microcontroller at a neighborly distance (see **Figure 6**). We have more to say about that in the two following sections. Unlike the analog board, the digital board (**Figure 7**) holds only one of the four addressable I/O expanders. The eight AC line voltage inputs are at the top, and the eight low-voltage inputs are at the bottom. The component and track spacings conform to all necessary and prescribed distances. The channels are routed to the I/O expander located on the right side of the board. The resistors (R14–R18) which determine the address of the expander (see **Table 2**) are located to the right of the IC. Four of these boards are necessary to achieve the maximum possible number of digital inputs (64).

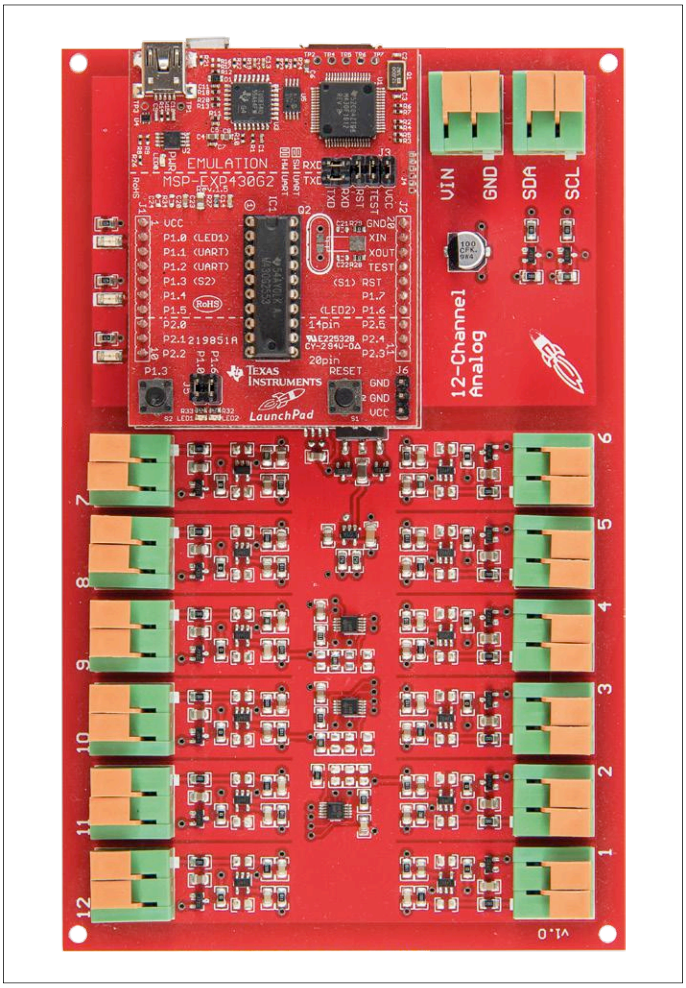


Figure 6. The analog board with the isolated microcontroller section.

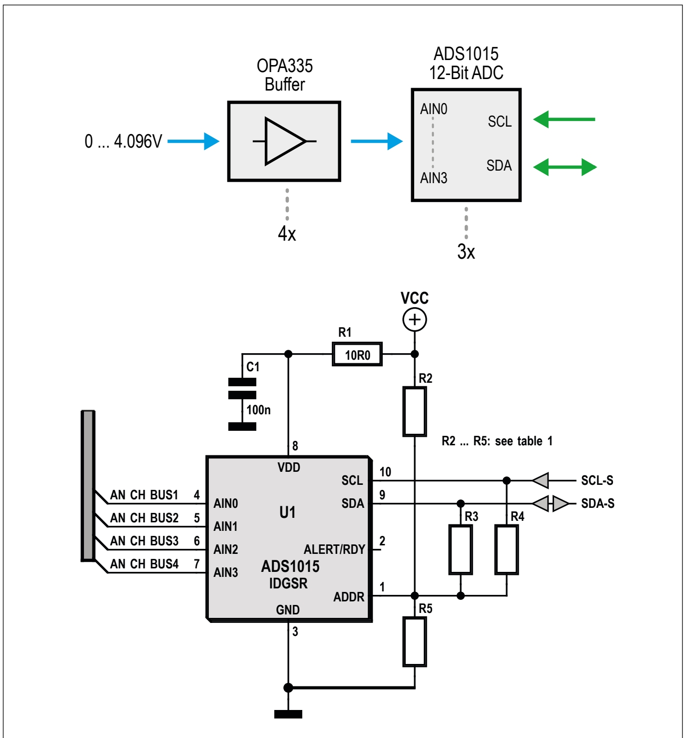


Figure 5. Twelve analog inputs (4 × 3) can be read out over the I²C bus.

Table 1. Address coding of the ADS1015 A/D converter (–: open; •: 10 kΩ).

ADDR (pin 1)	Hex address (decimal)	R2	R3	R4	R5
GND	0x48 (72)	–	–	–	•
VDD	0x49 (73)	•	–	–	–
SDA	0x4A (74)	–	•	–	–
SCL	0x4B (75)	–	–	•	–

Table 2. Address coding of the I/O expander (–: open; •: 0 kΩ).

A1 (pin 2)	A0 (pin 21)	Hex address (decimal)	R14	R15	R17	R18
L	L	0x74 (116)	–	–	•	•
L	H	0x75 (117)	–	•	•	–
H	L	0x76 (118)	•	–	–	•
H	H	0x77 (119)	•	•	–	–

Figure 7. The digital board in Eurocard format.

All boards are in Eurocard format with dimensions of 100 × 160 mm. The mounting holes at the corners of the boards are in the same place on all of the boards, so they can easily be stacked with threaded standoffs to form a sturdy package which can be fitted in a distribution enclosure, as shown in the lead picture for this article. That also provides suitable protection against accidental contact.

Analog isolation

On the digital side, optocouplers provide galvanic isolation for the USB interface and the microcontroller, but on the analog side things are a bit more complicated. As you can see from **Figure 8**, isolation is necessary for both the I²C bus (green) and the supply voltages (red) of the A/D converters and the microcontroller. I²C bus isolation is provided by the ISO1541 bidirectional I²C isolator, and the isolated supply voltage for the input side is provided by the combination of an SN6501 push-pull driver and a transformer. As shown in the detailed circuit diagram in **Figure 9**, there the supply voltage for the analog inputs is stabilized at 5.0 V by an LP2985-5.0 fixed voltage regulator. The rest of the circuitry (on the other side of the isolation barrier) is powered by +5 V from the USB port. No external power supply is necessary, and the connection between the PC and the instrumentation boards is limited to a single USB cable.

All this provides safe isolation between all inputs and the USB interface, so the user and the user's PC are reliably protected against any errors on the input side.

The microcontroller and its firmware

As can clearly be seen from the board layout, the isolated microcontroller module is located on the left side of the analog board. This module consists of a ready-made, very low-cost LaunchPad development board (MSP-EXP430G2 [1]) from Texas Instruments, which is simply plugged into the analog board. This board is intended for developing projects based on the low-power, low-cost microcontrollers in the MSP430G2x series (in old-fashioned DIP packages). There are also some components located around the LaunchPad connector for overvoltage protection, decoupling and indication, which do not need further mention here. The corresponding parts of the circuit are shown in the schematic diagrams included in the download package [2].

The LaunchPad development kit (**Figure 10**) comes with two MSP430 microcontrollers with up to 16 kB flash, 512 bytes RAM, 16 MHz CPU clock rate and integrated peripherals, including an eight-channel 10-bit ADC, timers, and serial interfaces (UART, I²C, SPI). On the board there is a 14/20-pin DIP socket, buttons, LEDs and connectors for a wide variety of modules with additional functionality, such as wireless links or display, as well as a USB programming cable. This means that when you aren't using the heating system data logger for its original purpose, you can develop all sorts of projects with the LaunchPad.

The firmware was generated using Energia [3], a rapid prototyping platform for the TI LaunchPads based on the Arduino IDE. It comes with pre-integrated libraries, making software development a lot faster. The C source code [2] is straightforward and can easily be modified or extended for other purposes, such as outputting or formatting the measurement data for visualization programs other than the one used by the author. The download folder contains two firmware versions – one for Energia 16, in which the project was originally developed, and the other for the current Energia 18 platform version. The C code for Energia 18 is current, but some modifications to the platform are necessary to enable it to recognize the MSP430G2553 microcontrollers. They are described in the program comments.

The software structure is extremely simple. For definition and initialization of the I/O ports, the I²C buses and the USB interface, the microcontroller reads the jumpers on port 2, which is located on the bottom of the LaunchPad board and therefore not visible in the picture. These jumpers set the interval between two successive data sets, since with slow systems such as a heating system it is more than sufficient to make measurements every five or ten seconds.

Then the main loop of the microcontroller firmware sequentially reads the raw data from the A/D converters and the I/O expanders. The variable `channel` is used in `switch-case` structures to differentiate between the various I²C addresses and I²C registers. The software is designed to handle all three A/D converters, but only one I/O expander at the address 0x74. If you want to address additional I/O expanders, you can simply duplicate the relevant code segment and enter the corresponding addresses.

The measured values are then converted and output over the serial interface, which appears as a COM port on the PC. Ana-

log values are represented as integral millivolts, while digital inputs are shown as either 1 or 0. The twelve analog values are sent first, followed by the sixteen digital values. The values are separated by slashes, with a line feed at the end of the output string.
The author uses the RealView curve plotting program from Abacom [4] to display the data on the PC. This program is fairly flexible and very user friendly with regard to integration of user-defined hardware. Formulas stored in RealView are also used to convert millivolt values into corresponding temperature values.

Summary

The data acquisition system described here has been working faultlessly for several months and has made a decisive contribution to the optimization of the heating system settings, and thereby to the efficiency and lifetime of the system. Thanks to visualization of the relevant quantities, such as temperatures and operating states, it can be seen relatively quickly whether changes to the parameters of the heating system controller have the desired effect and whether they are effective in the longer term under different conditions.
As previously mentioned, the project described in this article is not limited to heating systems. We have shown how to route any desired analog and digital input signals to a microcontroller and connect them to a PC for evaluation. This knowledge can be applied to a wide variety of instrumentation tasks. ◀

(160358-I)

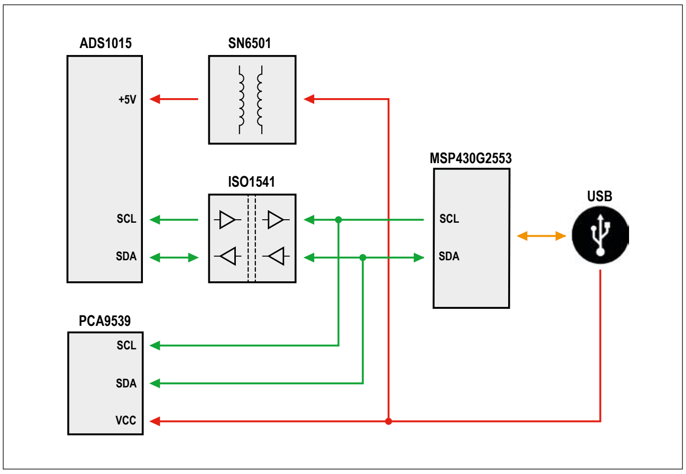


Figure 8. How the microcontroller and its USB interface are isolated from the instrumentation inputs.

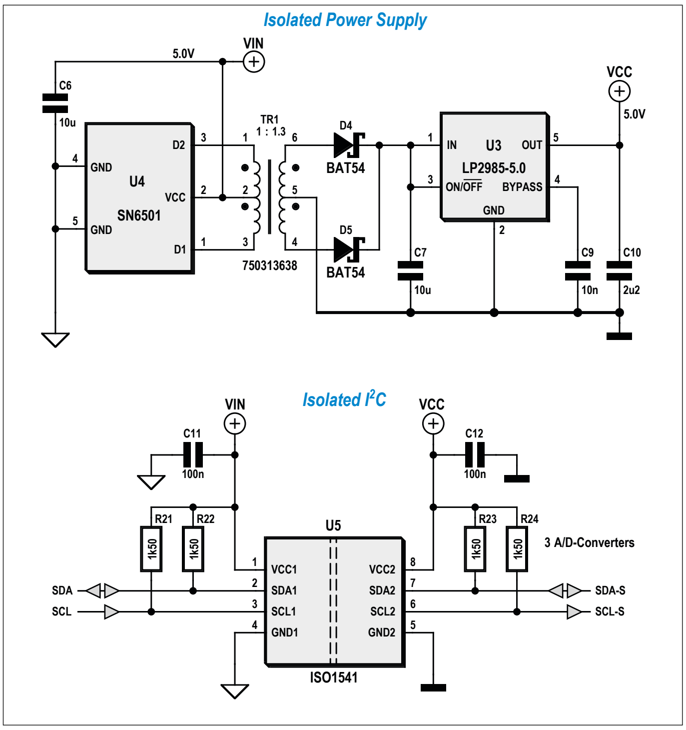


Figure 9. Isolating the I²C bus and the supply voltage.

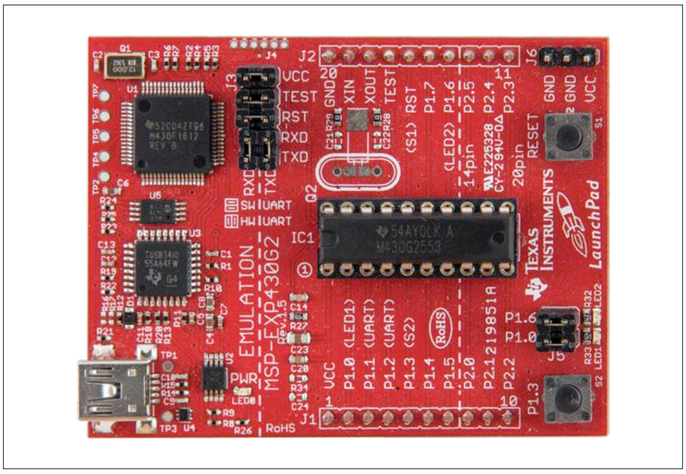


Figure 10. A simple, low-cost LaunchPad microcontroller from Texas Instruments handles the firmware tasks.

Web Links

[1] www.ti.com/tool/MSP-EXP430G2#0

[2] www.elektormagazine.de/160358

[3] http://energia.nu/

[4] www.abacom-online.de/uk/html/realview.html

What’s in the download package

The download package [2] for this homelab project contains:

- The firmware for Energia 16 and Energia 18
- Component lists
- A complete set of schematics in PDF format
- The PCB layout with component overlay in PDF format
- The PCB layout as an Altium project (which can be imported into Eagle)
- Drilling files (Gerber)

Basic and the Embedded World

Set to work with a PICAXE

By **Dr. Veikko Krypczyk** (Germany)

Getting started with microcontrollers can be daunting for the complete beginner. It's not just writing the programs but also getting to grips with some basic electronic principles at the same time. PICAXE is a solution to the problem. It is a complete, simple to use learning platform which offers quite sophisticated capabilities. You can later expand the basic system and transition to professional programming. We took a closer look...

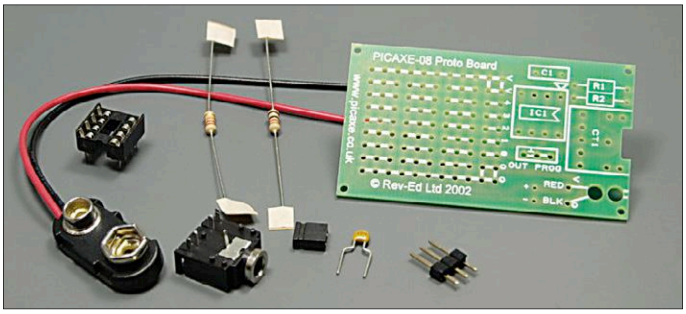


Figure 1. Parts kit for the PICAXE 08 prototype board (photo: www.picaxe.com).

The press and media are keen to point out that the shortage of engineers and technologists is hampering the growth of industry and the situation is likely to get worse if the young fail to engage with technology. The ubiquity of computing and electronics in the 21st century is undeniable and unlikely to diminish but most of us remain passive 'users'. With all the technology hidden behind a shiny exterior it seems almost impossible to hope you could ever understand how it's all put together.
Anyone starting out on the quest to learn more can be sure that any introduction to the world of microcontrollers will not be restricted to software alone but must also touch on the hardware aspects of a design application. Don't ever doubt that any time invested in the pursuit of knowledge will be wasted, as we move to an 'always connected' future more and more devices will need to communicate as the 'Internet of Things' takes off. Where do you start? Whenever you begin to explore any new field of knowledge it's important to use a tried and tested approach which guarantees some early success. Unnecessary complexity and frustration and can cause the student to throw in the towel too soon. PICAXE is a teaching platform with the aim to introduce the student to the secrets of microcontroller programming in a convenient and structured way. The system has resources in the form of documentation, worked examples, projects and the support of an active community to ensure that anyone starting out with the system will not feel out on their own. This article gives a brief introduction and overview of the PICAXE development platform. By the end you should be able to make up your mind whether PICAXE is the right system for you (or for a third party) to start out on the path of microcontroller discovery. As well as an overview of the hardware and software we use the system to work through an example.

PICAXE — an overview

So what is PICAXE? It is in fact a complete and extensive modular microcontroller teaching platform comprising of both hardware and software. The system is based on the popular PIC family of microcontrollers. The complete system has been designed to ensure a problem-free introduction to the technology. Several boards, offering various levels of sophistication are available but for a simple, minimal introduction the starter kit includes a preprogrammed controller together with a few passive components, a data transfer cable and battery holder and is available for around 30 euros. The software can be downloaded for free from the PICAXE home page. It consists of all the important components such as the drivers for the boards and the complete IDE.

The Starter Pack

Whenever you start something new it's a golden rule that everything's got to go smoother with a starter pack! The PICAXE system has a number of boards which offer various classes of power. Basically any one of these boards will be suitable to run the first example. The boards differ in their input/output, interfacing and prototyping capabilities. The type of controller supported also varies depending on which board you choose. The simplest is the PICAXE-08 prototyping board. In kit form you can order it at a cost of around 5 Euro. By selecting the corresponding starter pack (less than 30 euros) you have all the hardware you need to begin. What's include is the PICAXE-08 project board (**Figure 1**), a battery holder to power the system, a PICAXE microcontroller type PICAXE-08M2 and a programming cable which connects to a USB port. Software is downloadable from the project home page free of charge. At first you can use the battery holder as a power source so you won't need an external mains adapter. The system operates at a maximum of 6 V so don't be tempted to connect a 9 V battery to the battery clip, it will damage the chip.
Choosing a kit means that some soldering will be necessary to fit the components. If this doesn't appeal then you can order a fully assembled board with the starter pack. For the more ambitious there is the large AXE091U development board which supports all versions of the PICAXE controller and has features to allow

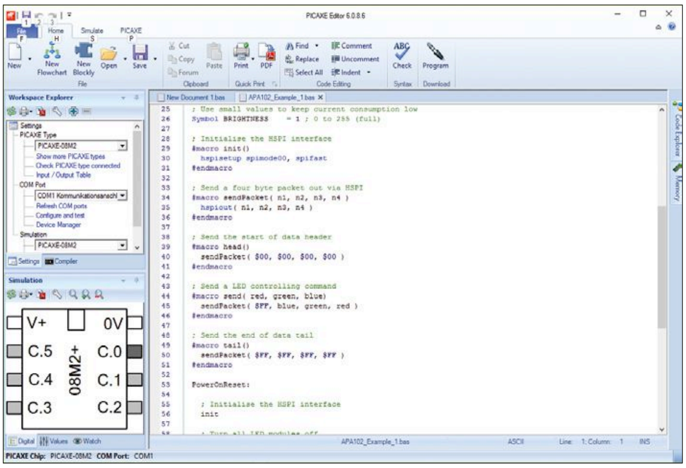


Figure 2. The PICAXE IDE Editor running in Microsoft Windows.

Feature	08M2	14M2	18M2	20M2	20X2	28X2	40X2
Memory Capacity (bytes)	2048	2048 x2	2048 x2	2048 x2	4096	4096 x4	4096 x4
RAM (bytes)	128	512	512	512	256	1280	1280
Byte variables (bytes)	28	28	28	28	56	56	56
Input/Output Pins	6	12	16	18	18	22	33
ADC/Touch Pins	3	7	10	11	11	16	27
Max. Freq. (MHz)	32	32	32	32	64	64	64
Serial In/Out	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Infrared In/Out	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I²C	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Tune (ring tones)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Parallel Tasks	4	8	8	8	1	1	1
Program Slots	1	2	2	2	1	4	4

Figure 3. Overview of the PICAXE microcontrollers [1].

more sophisticated circuit experimentation and development. Together with all the other accessories this works out at just over 80 Euros. Altogether not an unreasonable expense for your first foray into the world of microcontroller programming. The larger starter kit has much more potential for experimenting thanks to all the additional components and devices on board. In detail, amongst other features it includes:

- Support for all the PICAXE range of microcontrollers.
- Power supply from batteries or mains adapter.
- Socket for program download plus cable.
- Integrated prototyping plug board.
- Power-on status LED and three control LEDs.
- Three miniature push buttons, photoresistor, pot to simulate analog signals, temperature sensor, IR LED and TSOP for testing IR remote control, 7-segment display, socket for PS/2 keyboard, socket for DS1307 real-time clock and MAX202 interface chips for serial communication.

Let's go!

Before we get down to any serious play we need to make sure everything is set up properly. Its always best to be working with the latest software version, even though the starter pack includes a CD with all the necessary development environment software there is likely to be a newer version available from the home page [1] so check this out first, it also good for software support. In addition to Microsoft Windows there are also tools for Linux and the Mac OS. We will use Microsoft Windows. So to begin we install the latest version of the development environment PICAXE Editor (**Figure 2**). It runs on Windows 10 and versions all the way back to XP. The system library .NET 3.5.1 is required here; if you are using Windows XP or Vista this must be installed later. Connect the download cable between the PC USB port and the development board. The Device manager in the System option of the Control Panel installs the driver. A serial COM port is emulated using the USB port. In our case COM3 is automatically chosen. This information will be required later on during use and setup of the IDE.

First test...

The PICAXE projects emphasize the interplay between software and hardware. First you need to connect the hardware up according to the schematic then program the system software. For the first test we will just be satisfied to blink an LED on the board. The LED connects to the controller via a series resistor. For complete beginners it will be necessary to get familiar with the pin configurations of the controllers and their capabilities. The range of microcontrollers offer different levels of processing power and control capabilities but they are all based on the original PIC design. **Figure 3** gives an overview (memory size, I/O pins, clock frequency etc.) of the different models available. The chip's pin layout will depend on which model you choose. The three smallest versions use the pinout shown in **Figure 4**. The input/output pins are marked with capital letters (B or C) indicating the port and then a number representing the port pin number. Circuits don't get much simpler than **Figure 5**. Depending on the board you are using, this can either be wired up on the prototyping area or alternatively with short lengths of wire.

The program code (**Listing 1**) for this first program is written in Basic and is about as complex as the hardware. The PIC microprocessor supplied in the kit is preprogrammed with the bootstrap code for program download. Some common routines or language macros are also implemented in the code to reduce the size of the transferred code. For example to set port C pin

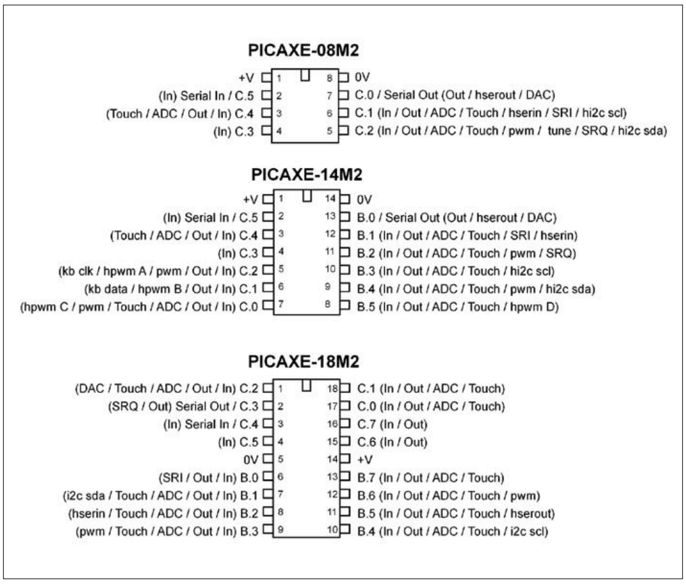
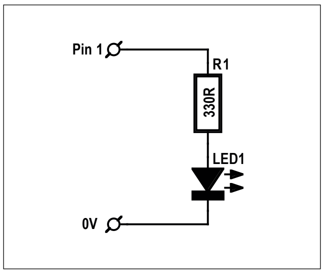


Figure 4. PICAXE microcontroller pin assignment [1].

1 to a logic high we use the command `high C.1` and to set it a logic with `low C.1`. The other commands (`pause, do-loop` for example) are very easy to understand. Even complete novices with no previous experience of programming will soon pick it up and in no time will be writing their first program. The program implements a continuous loop which switches the LED on for 1 second and off for 0.5 second.
In this way you can use Basic dialect to create software and program the PICAXE controller. Only PIC controllers supplied by PICAXE with the preloaded bootstrap code will work with the system. This code is not available for programming into blank controllers. All other essential information (language command set, board specifications and uses, pinouts etc) and a whole host of application ideas are available on the company web page [1]. The PICAXE environment has some unique features that make it an ideal learning tool and we will briefly take a look at what's on offer.



Listing 1.
The LED blinks!

```
do
   high C.1
   pause 1000
   low C.1
   pause 500
loop
```

Figure 5. The LED circuit connects to the PICAXE microcontroller.

From graphics to source code

Graphics tools such as flow charts are an effective aid for newcomers to the dark art of coding. The PICAXE development environment supports their use and also has other options. The entire program or parts of the program can be generated using flow charts (**Figure 6**). The resulting code can then be transferred directly to the microcontroller (referred to here as a Download). More often however you will want to edit the source code. For this you have the possibility to convert the complete flow chart into Basic. This approach allows you to graphically define the program's structure quickly using loops and branches etc. and then to zone in on the detail and expand the code with individual commands and variables. It's not possible to convert back in the other direction i.e. from Basic to a flowchart. Another option is the graphical representation of the code using Blockly. This is reminiscent of the program language Scratch. The program is broken down into a series of interlinked blocks (For example, the body of a loop). Each block can be configured individually and other constructs added such as inner or nested blocks. Whether you choose Blockly or the classic flow diagram is really just a matter of taste. The representation in Blockly is a variant of a Nassi-Shneiderman diagram but both methods are good for implementing algorithms.
Another feature of the PICAXE IDE is its ability to simulate the program on a PC before downloading it to the chip. The simulator is built in to the IDE and shows things such as level changes at the microcontroller pins.

Conclusions and outlook

Approaching the system from the point of view of a complete PIC controller newbie I must say I found the system really easy to get on with. Working with the starter pack made program loading a cinch and the free software made sure there were no unseen additional costs. Even when using the smallest board you get the chance to try out all the features of the PICAXE IDE. Using the PICAXE environment you quickly establish a sequence of copying examples, experimenting with your own ideas and learning. It's easy to lose track of time and that's always a good sign you are engaging with the system and actually learning something. As for the question — what comes next? — the PICAXE system lends itself to expansion and already offers a wide range of expansion boards. There are for example other microcontrollers in the series which can offer more power and extension boards to expand capabilities with LED displays, sensors etc for the development of control systems for real world events. More importantly the system gives a good grounding and familiarity with the PIC range of microcontrollers. Even though in the future you may be developing commercial systems with PIC processors using languages such as C or Pascal the underlying processor will be identical and you will have a good knowledge of its capabilities. The PICAXE learning environment is therefore a good solid entry-level platform with lots of potential.

(150762)

Web Link

[1] www.picaxe.com

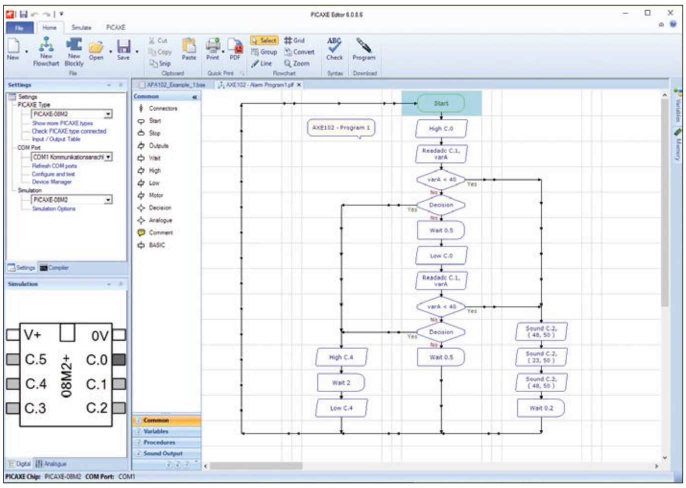


Figure 6. Flow chart programming.

Made in England

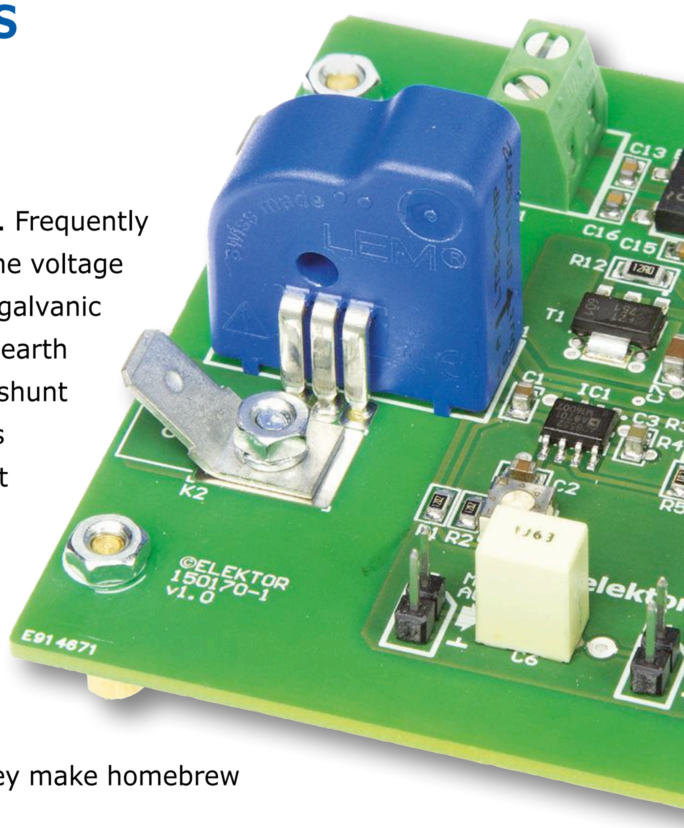
PICAXE products are developed and marketed by Revolution Education Ltd in the UK. Components can ordered from their online store at http://picaxestore.com.

eLABS PROJECT

Current Transformer for Oscilloscopes

Potential-free (floating) measurement using current clamp techniques

By **Karsten Böhme** (Germany) and **Ton Giesberts** (Elektor Labs)



Electric current can be measured in differing ways and processes. Frequently we make use of a shunt resistor, across which we can measure the voltage drop caused by current flow. Actually many oscilloscopes have a galvanic (electrical) connection between signal ground and the protective earth (ground) wire of the 230 VAC line supply and for this reason the shunt method is often inappropriate. Commercial current clamp devices certainly are able to measure current by potential-free means but they are not suitable for capturing rapid alterations in current over time. In addition many low-cost examples of these do not provide an output for connecting to an oscilloscope. If you're looking to measure floating current with an oscilloscope, there is no avoiding the use of an expensive current clamp designed for use with oscilloscopes, which besides can cost you more than the 'scope it's hooked up to. One benefit of these high prices: they make homebrew particularly worthwhile!

Technical Data

- Measurement range: ±25 A nominal effective current (±80 A max. pulse, output voltage clamped)
- Accuracy: 0.2% typical, 0.7% max.
- Bandwidth: 200 kHz (–0.5 to +1 dB)
- Current consumption: 50 mA (at 19.5 VDC on K1)
  44 mA (at 9.2 VDC on K1)
- Power supply (K1): ≥9.2 VDC (below this MOD2 switches off)
  ≤19.5 VDC (max. 18 V on MOD2 input)
  9 to 12 VAC (mains transformer secondary voltage, see text)
- Transducer factor: 100 mV/A at K4 (gain = 4)
  250 mV/A at K4 (gain = 10)
  25 mV/A at K5 and K6
- Output voltage: max. ±2.5 V at K4 (gain = 4)
  max. ±5 V at K4 (gain = 10, limited to ±20 A)
  max. 3.125 V and min. 1.875 V at K5
  max. 625 mVAC effective with 25 A sinewave

Making your own floating current transformer for 'scope use is not only rewarding but also amazingly simple. In fact special components exist for this purpose that do the main work for you. We're talking about a sort of 'current clamp for PCB mounting'. This converts the current flowing in the object under test into a magnetic field and then turns this into a proportionate voltage suitable for any 'scope. All in all, this setup functions just like a 'real' handheld current clamp. The following is a brief explanation of this double conversion of physical quantities.

Current clamp principles

As explained above, we have here a current that is first turned into a magnetic field and from that into a voltage.

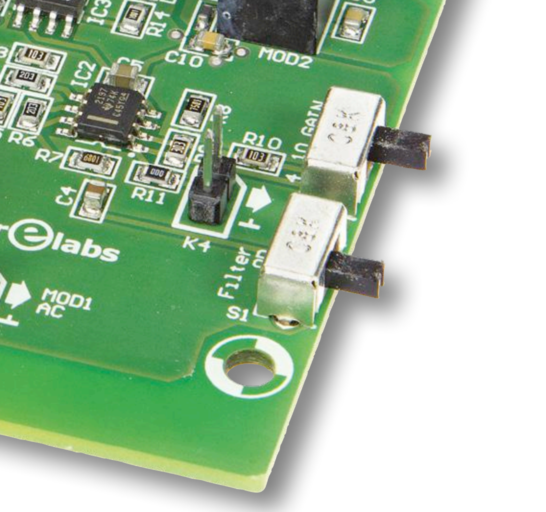



Figure 1 shows the three essential elements in functional form. At the input (left) we have a ferrite toroidal core with an air gap. Passing through this, like a coil winding, is a wire whose current is to be measured. The core is insulated with a plastic coating, which guarantees potential-free operation even when using a bare wire. The current flowing through the wire induces, (mainly) in the core, a magnetic field that is concentrated in the air gap.

A Hall sensor then converts the magnetic field of the air gap into a proportional voltage. The amplifier that follows takes care of level and impedance matching. Quite simple really. A major advantage of this method is that the shunt-free measurement is extremely low-resistance. Using a Hall sensor enables you to measure DC currents as well, which would not be possible if you used the toroidal core as a transformer, whilst not measuring the magnetic field with a Hall sensor would involve the need for a secondary winding. To avoid having to first disconnect the wire and thread it through the core, handheld devices are arranged to be hinged. With an instrument transformer for oscilloscope use this is less important, since 'scopes are not normally used out and about. The key requirements in this situation are the ability to take floating measurements, the bandwidth and the accuracy.

PROJECT INFORMATION

Oszilloscope
measurement current
floating

entry level
➡ intermediate level
expert level

3 hours approx.

Soldering instruments
Drill and drill bits
Lab power supply

€100 / £90 / $120 approx.

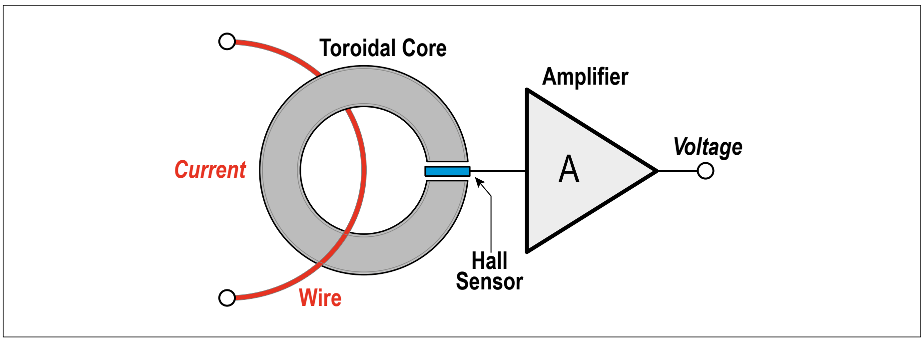


Figure 1. This block diagram shows the functional principles of a current clamp.

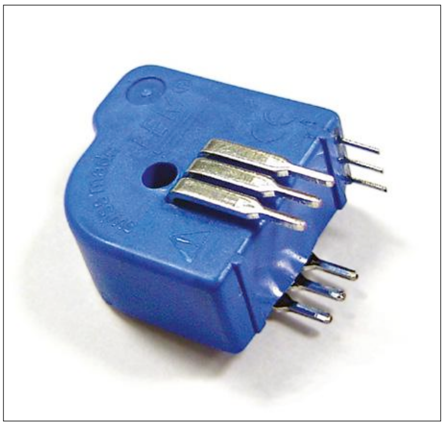


Figure 2. The current transducer module NTS 25-NP from LEM has three 'windings' that can be connected in parallel or series.

Current transducer module

For several decades the Swiss company LEM has been involved with everything to do with heavy currents. Their product line even includes a current transformer that integrates all three of the functions seen in Figure 1 into a single ready-to-use module. The LTS 25-NP 'current transducer' can be seen in **Figure 2**. Its input side is already provided with a 'wire' in which the current flow can be measured without worrying about hinges. To be more precise, we can recognise three wires here. You can connect these in parallel for large current for smaller ones in series, enabling you to use the same module for different ranges of measurement.

With three parallel 'windings' you have a measurement range of 25 A with a precision resistance of just 180 µΩ. Two windings in series produce the higher sensitivity of 12 A for a resistance of 810 µΩ. If all three windings are connected in series, you can record constant currents up to a maximum of 8 A at a resistance of 1.62 mΩ. As you can see, the effective resistance is not linearly proportional to the measurement range. This is because when the windings are in series, the lion's share of the resistance occurs in the tracks (traces) of the PCB. Finally remember you are producing an estimate that is dependent on physical circumstances. For this project we use the 25 A range of measurement, which means you don't need to waste a moment's thought on such subtleties.

The integrated electronics have only three connection pins. Two of the pins carry ground and +5 V from the power supply, with the output voltage available on the third pin. At 'rest', i.e. with 0 A current, half the supply voltage (2.5 V) will be produced here. When current flows, the output voltage will vary by a factor of 25 mV/A. If nothing else, the technical data should not be ignored: the accuracy of typically 0.2 % and bandwidth of 200 kHz (between –0.5 to +1 dB) are pretty good for a magnetic transducer. Further technical data can be found in the datasheet [1].

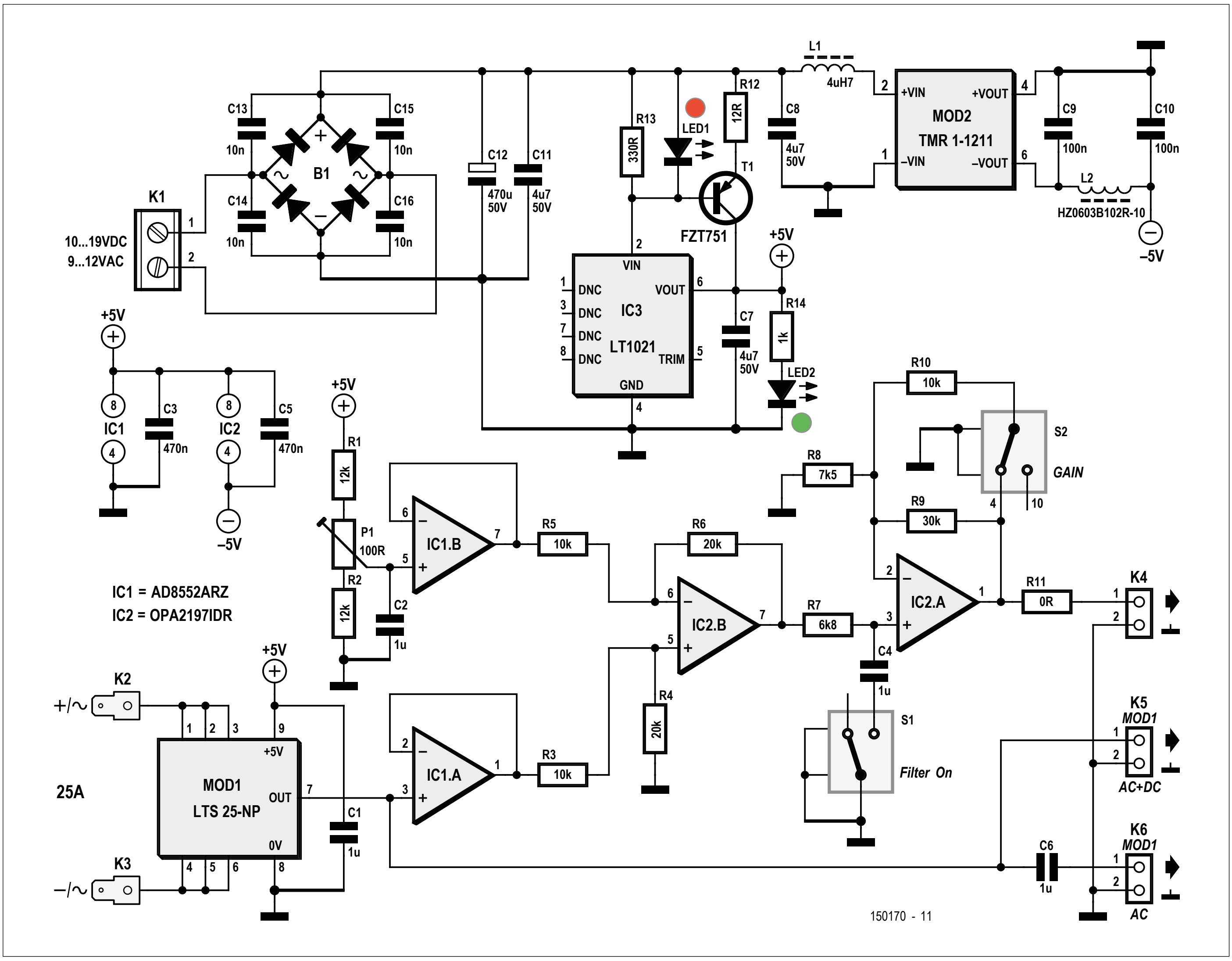


Figure 3. Schematic for a current clamp with toroidal core, Hall sensor and output amplifier.

Circuit

All the important stuff for potential-free conversion of current into a voltage measurable on a 'scope is already included in the NTS 25-NP module, which at a price of around €15 / £13 / $18 is definitely not over-priced. In fact all that's missing is the power supply and perhaps an op-amp for signal conditioning. But life is never that simple.

If you were to power the current transformer simply by using the 5 V from a USB connector, as provided as standard on many digital oscilloscopes, then you would also contaminate the analogue section with all the undesirable noise products and 'shash' of a PSU intended for digital electronics — quite apart from hum loops and other nasties that occur as soon as signal ground and supply ground become linked. For this reason a separate PSU is recommended and if you have got to this stage, you might as well go the whole hog.

Power supply

In the circuit of **Figure 3** this role is performed by a precision reference IC of the LTC1021-5 type [2]. Its output voltage has particularly low noise, low drift and long-term stability and even in its untrimmed 1% version is almost more accurate than necessary. A small problem arises from the fact that the output of the IC has a maximum load capacity of 10 mA. The current converter module alone already takes 28 mA, however; add to this a few more mA for the remaining electronics. The solution is an amplifier circuit using T1. Above 2 mA, which of course has to flow through R13, T1 is activated and the PNP transistor takes over the remainder of the current. But that's not everything.

Because the voltage on the output of MOD1 at 0 A is unfortunately not 0 V but a full 2.5 V, a problem arises if you advance the sensitivity of the oscilloscope, because then the base line (zero line) cannot be shifted properly. For this reason besides the direct output K5 and the output K6 DC-decoupled via C6 we have an additional amplified output K4, where with 0 A produces close to 0 V. The differential amplifier IC2.B subtracts the value of 2.5 V from the useful signal at pin 7 of MOD1 that has been set exactly to half the operating voltage with P1. The result is now symmetrical around 0 V. The voltage swing without amplification would amount to ± 625 mV at ± 25 A. Using IC2.B, gain can now be set to a factor of 4 (S2 = closed) or 10 (S2 = open). Fourfold amplification results in the intuitively plausible relationship of 100 mV/A. At ± 25 A, output K4 is then ± 2.5 V. With tenfold amplification we have a more sensitive conversion ratio of 250 mV/A. However, for an op-amp to deliver negative voltages, it must be supplied with a negative voltage. This task is performed by MOD2 [3], a small DC-to-DC converter. It supplies −5 V because its positive output is connected to ground. Last but not least, R7 and C4 create a lowpass filter selectable by S1 with a cut-off frequency of 24 Hz. This mode serves primarily to measure the DC component more accurately.

Finer details

Another word on the op-amps employed: the two op-amps in IC1 [4] act purely as buffers. Here minimum drift is more essential than everything else. With IC2 the degree of amplification is separated: IC2.B amplifies twofold and IC2.A therefore has a maximum amplification factor of 5. For a gain bandwidth of 10 MHz we still have 2 MHz bandwidth left — around ten times that of MOD1. Apart from that, the IC OPA2197 [5] offers a low offset and rail-to-rail outputs.

The feasible voltage range at K1 is determined by the requirements of MOD2. The minimum value arises from the minimum voltage at the MOD2 input plus the voltage drop at B1. The maximum value is determined by the highest voltage permissible on the MOD2 input minus the voltage drop at B1. You are in the safe area if between 10 and 19 V DC are applied to K1. A 12 V plug-in power supply is consequently well suited. However, direct connection to the secondary winding of a 9 V mains transformer also works. However, a small 12-V transformer can deliver too much voltage because it is lightly loaded. So do your measurement before connecting to the project!

Construction

Ton Giesberts (Elektor Labs) found a project published by Karsten Böhme on the Elektor Labs section of the Elektor website [6] so impressive that he took it under his wing and 'Elektorised' it. Along with some revision, a PCB was developed that fitted directly into an extruded aluminium housing already equipped with

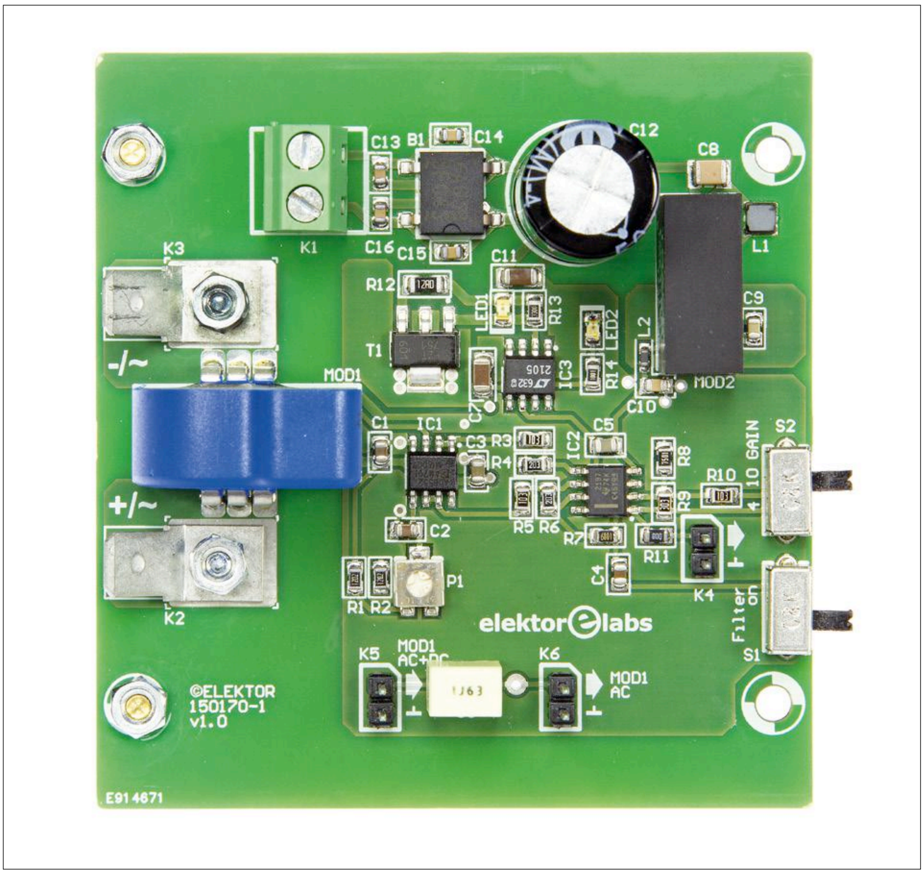


Figure 4. Component side of the PCB of the Elektor Labs prototype of the project.

Figure 5. Prototype constructed at Elektor Labs.

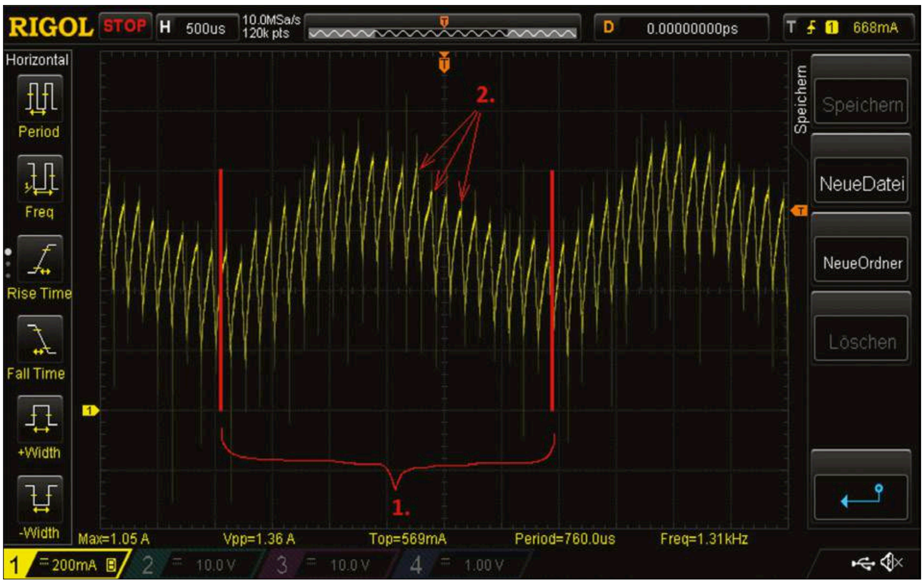


Figure 6. Screenshot (taken by author Karsten Böhme) of current measurement of an electric motor. The sluggish curve 1 illustrates the brush contacts being displaced by the commutator. The speedier component 2 shows the PWM frequency of the motor driver (7.8 kHz).

slots for retaining PCBs. For this reason the board turned out a little larger than otherwise necessary. As always, PCB data for DIY etchers is available to download free from the Elektor web page for this project [7]. **Figure 4** shows an 'aerial view' of the completed PCB of the prototype.

As you can see, it uses quite a few SMD components, but they are all consistently in soldering iron-friendly formats of at least 0805 or larger. If you take a little care, fitting them should not be a problem. As normal, you start with the lowest-profile components and only then apply yourself to the 'high-rise' components such as the potentiometer, switches, screw terminal block, pin headers, MOD2, the electrolytic C12 and MOD1. Last of all K2 and K3 are bolted on. For improved contact you can fix them with some solder on their pads — after the bolting operation.

Next some mechanical work is called for but fortunately not much. The recommended and exactly matching Hammond aluminium profile enclosure requires four holes for M3 mounting screws drilled in the bottom of the case. In addition, there are two elongated openings for the two slide switches and one end piece needs a hole for a BNC connector, which is screwed directly onto the aluminium housing. This ensures that the screened metal case is at the same ground potential of the 'scope input when connected using a coaxial cable with BNC connectors. Anyone who uses an insulated BNC connector will need to provide an extra connection from the output ground to the body of the metal case. The other end panel needs two openings for a pair of banana sockets plus one hole for the coaxial DC power supply connector. It goes without saying that these three connectors need to be insulated from the housing, otherwise your measurements will be 'somewhat' less accurate...

Testing

A first test with no connection to the input should produce a few mV at K4 and more or less exactly 2.5 V at K5. If this is the case, your soldering efforts must have been successful. Now come the adjustments: P1 is set to exactly 0.000 V at K4. A moment's humour here: don't short-circuit K2 to K3 for this — the input is sufficiently low-resistance ;-). With this dealt with, a test using 'real current' is recommended. A lab power supply with adjustable current-limiting will handle for this task well. With 4 A on the input you should have precisely 1.0V at K4 with S2 in the Gain = 4 setting. Now switch S2 to Gain = 10; 2.50 V should be observed. If you allow the current to flow in the opposite direction, −1.0 V and respectively −2.50 V should be indicated at K4. At K5 you should be reading 2.60 V or 2.40 V independent of how S2 is set. If no negative voltages are triggered at K4, first check whether MOD2 is producing 5 V (measured straightforwardly across C9). If this was not the problem, make the usual search for short circuits, imperfect solder joints and components connected back to front.

If everything is in order so far, the PCB can be placed inside the housing and fixed to the base with the four 10mm standoff pillars so as to provide good access for operating the knobs of the two slide switches from outside. After this the wiring can be carried out as in the prototype seen in **Figure 5**. One final test, with the cover in place and fully complete. To give you an idea of how well the current transducer works in practice, **Figure 6** shows a screenshot of the current flow in a DC motor, which Karsten Böhme produced with his prototype. You can see clearly how even rapid alterations in current can be indicated accurately. ⏮

(150170)

Web Links

[1] http://goo.gl/HDB7kb

[2] www.linear.com/product/LT1021

[3] http://goo.gl/uqW1vw

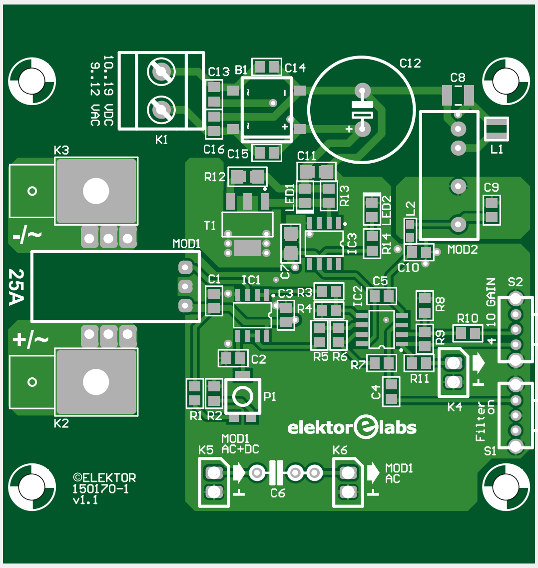
[4] http://goo.gl/FSiAoX

[5] www.ti.com/product/OPA2197

[6] http://goo.gl/JziUKZ

[7] www.elektormagazine.com/150170

COMPONENT LIST



Resistors

Default: SMD 0805, 1%
R1,R2 = 12kΩ
R3,R5,R10 = 10kΩ, 0.1%
R4,R6 = 20kΩ, 0.1%
R7 = 6.8kΩ
R8 = 7.5kΩ, 0.1%
R9 = 30kΩ, 0.1%
R11 = 0Ω
R12 = 12Ω, 0.25W, SMD 1206
R13 = 330Ω
R14 = 1kΩ
P1 = 100Ω, SMD, e.g. Bourns 3314G-2-101E

Capacitors

Default: SMD 0805, 50V, 10%
C1,C2,C4 = 1µF
C3,C5 = 470nF
C6 = 1µF 63V, PET, 5 or 7.5mm pitch
C7,C11 = 4.7µF, SMD 1206
C8 = 4.7µF, SMD 1210
C9,C10 = 100nF
C12 = 470µF, 20%, 5mm pitch, 13x21 mm
C13-C16 = 10nF 100V

Inductors

L1 = 4.7µH, 20%, 1.2A, 0.12Ω, e.g. Bourns SRN3015TA-4R7M
L2 = chip bead, 1kΩ@100 MHz, 200mA, 0.6Ω DC, e.g. Laird HZ0603B102R-10

Semiconductors

B1 = DF04S, bridge rectifier 400V 1.5A, SMD SDIP 4L
LED1 = red, SMD 0805
LED2 = green, SMD 0805
T1 = FZT751, PNP, 60V, 2W, 3A, SMD SOT-223
IC1 = AD8552ARZ, SMD SOIC-8
IC2 = OPA2197IDR, SMD SOIC-8
IC3 = LT1021DCS8-5#PBF, SOIC-8
MOD1 = LTS 25-NP, wired module, LEM
MOD2 = TMR 1-1211, SIP module, Traco Power

Miscellaneous

K1 = 2-way PCB screw terminal block, 0.2" pitch
K2,K3 = 6.35mm FASTON terminal, 3.3mm hole, e.g. TE Connectivity 42822-2
K2,K3 = 6.35mm FASTON female connector, blue, for 2.5 mm² stranded wire
K4,K5,K6 = 3 pcs 2-pin pinheader, 0.1" pitch
K4,K5,K6 = 1 (only) two-way female pinheader, 1/10" pitch
S1,S2 = slide switch, horizontal PCB-mounting, SPDT, 100 mA, e.g. OS102011MA1QN1
BNC socket for single-hole mounting (see text)
DC power socket for single-hole mounting, 2.1 mm, insulated
Banana socket, red, 4mm, insulated, 30A, e.g. Cliff Electronic FCR14461
Banana socket, black, 4mm, insulated, 30A, e.g. Cliff Electronic FCR14460
Enclosure, Hammond 1455K1201, 78x123x43 mm
Stranded hookup wire, 5cm, 2.5mm², red
Stranded hookup wire, 5cm, 2.5mm², black
Stranded wire for K1 and BNC connector
4 pcs flat washer, steel
2 pcs standoff pillars, steel, 10mm, M3 thread
Plug-in power supply, 12VDC, ≥100mA
PCB # 150170-1 v1.1

Err-lectronics

Corrections, Updates and Feedback to published articles

Jammer Alarm

Elektor Magazine 5/2017, p. 18 (160453)

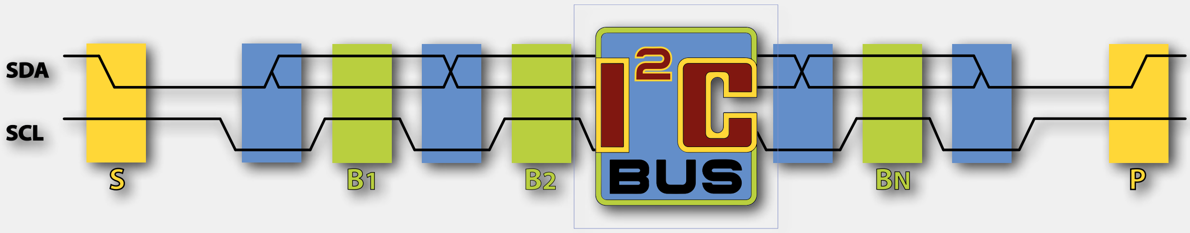
UPDATE. In order to improve the jammer alarm sensitivity increase the distance between the transmitter and receiver section. I mounted the transmitter module along with its pulse generator in a separate enclosure and powered it from a separate 9-V mains adapter (9 V is adequate for the module). Now you can experiment to find the optimum distance so that the receiver still reliably receives the transmitted pulses. The greater the distance, the more sensitive the circuit will be to detect weaker jamming signals. I also increased the transmitter pulse repetition rate to around 7 s.

Walter Meyer (Author of the article)

The I²C Bus

Elektor Magazine 4/2017, p. 42 (160095) – Elektor Magazine 5/2017, p. 38 (160418)

I recently noticed, the Raspberry Pi (or Raspbian) has a default maximum “clock stretching” duration of around 640 µs (http://raspihats.com/i2c-clock-stretching-timeout-on-the-raspberry-pi/).



I programmed an ATmega (with a DHT22 sensor attached) as an I²C slave and thought that I could just read the data from the DHT22 as required in my own time. The I²C controller in the ATmega keeps the clock line low until the TWINT-bit is reset but it takes too long! This can be accommodated by changing the timeout-register value in the BCM2835 (see URL above), but this is not a general solution. Instead I split access into two parts: Firstly a write transaction (the data value is ignored) triggers a read out operation and the five bytes are stored in a buffer in the ATmega. About a second later, I can then retrieve the data without any delay.

Josef Möllers (Author of the article)

Dimmable Outdoor Lighting

Elektor Magazine 4/2017, p. 74 (140574)

FEEDBACK. I tried to recreate the project using an Arduino to provide the control signal. In principle it works but the lamps are not dimmed progressively. I think the problem may be that some modern LED spotlights, along with a built-in rectifier also include a reservoir capacitor. This stops the lamp from flickering but the capacitor means the LEDs remain at the same brightness level when the lamps are dimmed in range from 20 to 100%, only when the duty cycle drops to less than about 10% do they start to dim noticeably. It also makes the transformer whistle! Which type of spotlights did the author use?
My LED lamps (three different types, all described as dimmable) exhibit the same behaviour.

Peter Richter

The author replies:
Not all LED lamps will be suitable for this form of control, as described in the article, I first tested the lamps to see if they could be progressively dimmed using a PWM signal.
The lamps I use in the article are: 21 LED LARA ROUND brushed chrome 2.8 W, part no: LED55019WW.

Andreas Meyer

RETRONICS

HP650A Test Oscillator (ca. 1948)

From 'boat anchor' to pristine sine-wave

Figure 1. Front view of my HP650 test oscillator in actual operation.

By **Dr Martin Beusekamp, MSc.** (Netherlands)

My collection of test equipment includes a Hewlett-Packard HP650A Test Oscillator. That model was launched in 1948, but the relatively high serial number in this case (5793) means it's an open question who's older: the device or its owner (model year 1954). Here I would like to tell you about how I acquired the device and how I brought it back to life and even updated it.

I bought the oscillator back in 1979 in defective condition for five Dutch guldens from a company where I was on a work placement stint. The original new price was US\$475 in 1948, rising to US\$550 a number of years later (with fairly successful sales figures). Unfortunately my unit came without documentation, so I wrote to the company to ask for help. I simply sat down and typed a letter, put it in an envelope with a stamp, and dropped it in the post box. Nowadays you need an old dictionary to find out what those words mean. I was more than willing to pay a reasonable price for a good service manual, but clearly HP liked the idea that someone wanted to breathe new life into an old device from their stable and kindly sent me an original manual free of charge. Nowadays hobbyists can simply search the web for a suitable manual.

The photo in Figure 1 shows the test oscillator in its present state. Read on to learn how it got there.

Switch it on and see what happens

My first measurements — nearly 40 years ago — quickly revealed that the power transformer had given up the ghost. And it was not a garden-variety example, with two 435 V secondaries for the anode voltage and four filament voltage windings — three separate 6.3-V windings rated at 2.5 A, 2.5 A and 1 A, plus a winding for 5 V @ 3 A. The latter winding was dedicated to a 5U4G dual rectifier, whose task (in combination with an electrolytic capacitor) was to convert the full-wave high voltage from the transformer to a supply voltage of 450 V_{DC}. This unregulated anode voltage then passed through a 6 H choke to a second hefty electrolytic capacitor. After that the voltage was regulated down to 180 V_{DC}, at a current of over 100 mA, by two 6L6GB tubes (similar to the European EL34) wired in parallel. They were driven by a 6SQ7 that took its reference voltage from a type 0A3 neon-filled cold-cathode glow discharge tube. The power dissipation of the anode voltage regulator alone was a tidy

$$(450 - 180) \times 0.1 = 276 \text{ watts}$$

augmented by more than 28 W of filament power for the previously mentioned tubes.

Finding a replacement for such a specific power transformer is anyhow difficult, let alone rewinding it yourself, and particularly considering the high power dissipation of the power supply I went looking for a different solution. By chance I had two identical power transformers from salvaged tube radios, which in combination could provide the desired voltages and the necessary power. However, I switched to BY127 silicon diodes to rectify the secondary high voltage. The hefty 6-H choke remained, but anode voltage regulation is now provided by an LM723 IC (an old stalwart) and four BU426A

Figure 2. Top view of the HP650A signal generator in its original state. The power transformer, which in my case was defective, can be seen at the upper left. To its right is the 6-H choke (that's right, six henries). The six tubes in the back row are all part of the regulated power supply. Note: This photo was taken a long time before there was any thought of writing an article for Elektor.

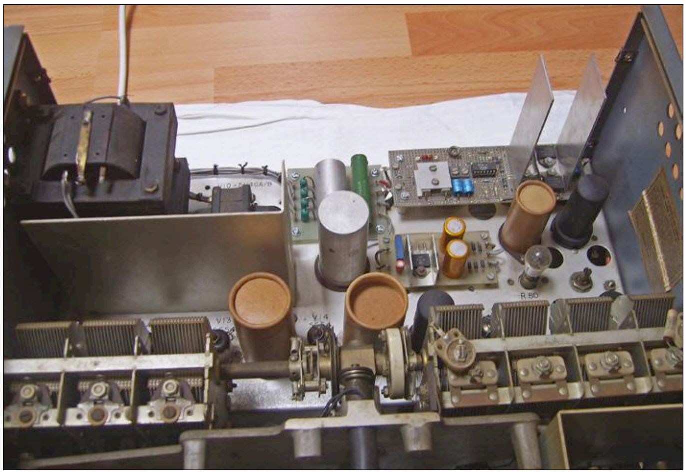


Figure 3. Top view of the signal generator after some modifications. At the top left are the two power transformers salvaged from radios. The original 6-H choke is still there, but the row of six tubes visible in Figure 2 has been removed and replaced by power supply circuits using transistors and an IC. The famous 10-W incandescent lamp for amplitude stabilisation (see the inset **'HP's roots'**) is located to the immediate right of the smallest circuit board.

transistors wired in parallel.
In the original circuit, the filaments of five 5654 tubes (similar to 6AK5/EF95) were powered from a sort of regulated filament voltage obtained from a 12.6 V_{AC} source (two of the three 6.3-V windings in series) in series with a type 12-4 ballast tube, which has a characteristic similar to a PTC resistor. If the filament current rises, the resistance of the ballast tube also rises; the result is a regulated filament current. I solved this issue by powering the filaments of the five tubes concerned from a regulated 6.3-V_{DC} supply, which also enhances the hum characteristics of these tubes. After this overhaul of the power supply, the test oscillator worked nicely right away, so I could turn my attention to the rest of the circuit.

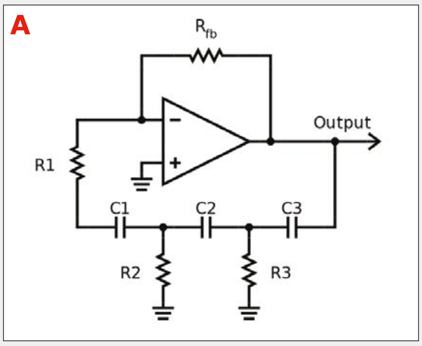
Two oscillators

The HP650A is a pure sine-wave generator, which means it is not a function generator that derives a sine-wave signal (or something that looks like one) from a triangle wave or square wave signal. More precisely, the generator comprises two completely separate oscillators. The state of the art and the available components at the end of the 1940s were not up to the task of building a signal generator that could work properly and within tight specifications over a range of six decades (10 Hz to 10 MHz), which is nearly twenty octaves. A Wien bridge oscillator is used for the lower four decades (10 Hz to 100 kHz), while an oscillator with three RC phase-shift stages (see the inset **'3 × 3 × 3 and a bit'**) handles the upper two decades (100 kHz to 10 MHz). The frequency range selector switch also switches the supply voltage for the two oscillators.
For frequency adjustment in a modern signal generator, we would use a (multi-turn) potentiometer to set the frequency within each decade range, and capacitors with values a factor of 10 smaller for each increasing decade. The approach used in the HP650A is exactly the opposite. The frequency decade switch selects fixed resistors, and tuning is performed using rotary variable capacitors (visible in the foreground in **Figure 2** and **Figure 3**). Here it should be mentioned that Figure 2 shows the test oscillator before the start of the renovation, and this photo was made a long time before Jan Buiting asked me to write this article. The resolution of this photo is accordingly much lower than the current standard for publication in a magazine.
The three capacitor sections on the left in the photos — with 603 pF per section according to the manual — form the capacitive part of the phase-shift stages of the high-frequency oscillator (see the inset). The four capacitor sections on the right are connected in parallel as two pairs. With 535 pF per section, that creates two 1.07-nF variable capacitors for

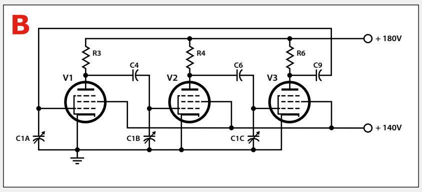
3 × 3 × 3 and a bit

The oscillator for the two upper frequency decades (100 kHz to 10 MHz) of the HP650A is a conventional phase-shift oscillator, which consists of an inverting amplifier and three RC phase stages, each contributing a phase shift of 60 degrees, see **image A**.
However, the amplifier in this sort of oscillator needs to have a gain of -29, in this case all the way to 10 MHz, and that was not really feasible at the end of the 1940s.
HP therefore opted for three amplifier stages, each with a gain of slightly more than 3, see **image B**.
An additional benefit of this circuit is that the load on the output of each stage of the circuit is minimal.

A



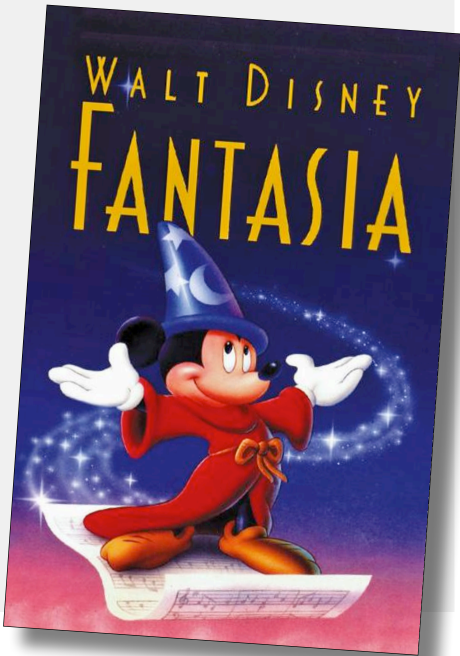
B



HP's roots

The 'HP garage' in Palo Alto is commonly known as the birthplace of Silicon Valley. Photo: Jan Buiting.

A sine-wave oscillator for the audio frequency range was the first product made by William ("Bill") Hewlett and David ("Dave") Packard in 1939, working in a rented garage in the village of Palo Alto, California. They tossed a coin to decide whether the company would be known as 'Packard-Hewlett' or 'Hewlett-Packard'.
The key innovation in their very first model, the HP200A, was the use of an incandescent lamp as a temperature-dependent (and therefore voltage-dependent) resistance for amplitude stabilisation of the generated signal. As a result, the HP200A with a selling price of less than US$90 was more stable than significantly more expensive competitive instruments. The HP650A also uses an incandescent lamp for amplitude stabilisation, as do most more recent oscillators. By the way, the first model number (200) was chosen arbitrarily, but it was intentionally not a low number. Otherwise it would have been apparent that Hewlett and Packard were newcomers in the industry. For more information about the HP200A, see the Retronics article in the April 2014 issue [2].
Several minor improvements led to the HP200B model, which was good enough for Walt Disney's studio to buy eight of them for sound effects in the film *Fantasia*. The film was no blockbuster, but that wasn't due to the HP sine-wave oscillators. Overabundant use of classical music, loss of the European market due to the Second World War, and the requirement for cinemas to have a surround-sound audio installation for proper appreciation of the sound effects all hampered the popularity of the film.



the lower four frequency decades.
The two oscillators feed their signals to an output amplifier composed of three tubes, which produces the maximum output signal amplitude of 3 V_{rms} at the output terminals. Even now, building an amplifier with a perfectly flat frequency response from 10 Hz to 10 MHz (within 1 dB according to the specifications) is not easy, and a bit less than seventy years ago it was virtually impossible. Nevertheless, the parasitic impedances of the tubes in the output stage are compensated as well as possible over the frequency range by implementing all anode and cathode impedances as networks of resistors, inductors with a value of several microhenries and small capacitors, in some cases in the form of trimmers. With a maximum output power of 15 mW (3 V into 600 Ω), it's apparent that most of the 165 watts of power drawn from the AC line (in the original version) serves to keep the vicinity of the device pleasantly warm.

Voltmeter included

Naturally, the user wants to know the amplitude of the output signal from the sine-wave oscillator. The HP650A is therefore equipped with a true vacuum tube voltmeter (VTVM) with scales in mV, V and dB. Due to the structure of the output attenuator, the voltmeter readings are only accurate when the oscillator is connected to a load that matches its characteristic impedance of 600 Ω. For that reason, I normally have a pair of resistors (330 Ω and 270 Ω) connected in series over the output terminals.

Let's see you do this with DDS

Following the revision described above, for 38 years now I have the privilege of using a sine-wave generator that was designed, developed and manufactured in the tube era, but now operates with a combination of tubes (there are still eleven left after getting rid of the tubes in the original power supply), a handful of transistors and a single IC. They cooperate in perfect harmony and produce sine-wave signals up to 10 MHz with significantly less distortion than many function generators.

(160621-I)

Web links

[1] www.hparchive.com/Manuals/HP-650A-Manual.pdf

[2] www.elektormagazine.com/130423

EST^D 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

ELEKTOR ETHICS

Internet Censorship during the Catalan Referendum

And how it was circumvented

Demonstration in Barcelona during the general strike on 3 October 2017. (Photo: Ernerst CS. CC BY-SA 4.0 licence)

On the day before the independence referendum in Catalonia, the ICT centre of the Catalan government was raided and occupied by the Spanish police. With the shutdown of the IT centre, the official Internet infrastructure intended to facilitate the referendum was largely rendered ineffective. Nevertheless, voting was possible on 1 October 2017 thanks to an alternative system set up and maintained by volunteers with technical expertise, along with the wireless Internet connections of thousands of citizens.

By **Tessel Renzenbrink** (Netherlands)

In the weeks leading up to the referendum in Catalonia, the Spanish federal government engaged in large-scale Internet censorship. With this censorship the government restricted free access to information by limiting what could be published and viewed on the Internet. The various censorship methods, and how they were circumvented by citizens and the Catalan government, have been charted by a social activist and Linux programmer named Matthias (surname unknown). He presented his findings at the 34th Chaos Communication Congress (CCC), the annual convention of German hackers, on 27–30 December 2017 in Leipzig [1].

.cat domains removed from the Web

According to Matthias, the censorship started with the seizure of the official website for the referendum, *www.referendum.cat*, on 13 September. The site content was no longer visible; visitors only saw a message saying that the domain name had been taken over by the Spanish authorities. In the following days, other *.cat* sites related to the referendum were also seized, including *www.ref1oct.cat*. (The Internet domain *.cat* is not dedicated to cat pictures, but instead to the Catalan language and culture.)

The *.cat* domain is administered by the foundation Fundació puntCAT. The foundation received three court orders in succession, with increasingly longer lists of websites they were supposed to take down. Along with specific websites, the foundation was ordered to block ‘all domains that contained any information about the referendum’, which meant that Fundació puntCAT was tasked with active monitoring of all *.cat* websites. That is an extraordinary and probably unlawful demand because the foundation, as a top-level domain operator, only handles registration of *.cat* domain names. It is like holding Verisign, which administers the *.com* domains, responsible for

the content of every *.com* website on the Internet.
On 20 September the online fight was extended to the physical domain. The police raided the offices of Fundació puntCAT. They seized computers, and the Technical Director was arrested. Among other things, he was charged with perverting the course of justice and disobedience.

Message on seized websites (public domain).

Demonstration during the national 'Yes Day', Barcelona, 11 September 2017. (Photo: Medol. CC BY-SA 4.0 licence)

Let a hundred sites bloom

In response to the seizure of *referendum.cat*, mirror sites sprang up everywhere on the Web (mirror sites are sites that display the same content but have different IP addresses), according to Matthias's presentation. The Spanish government also tried to block them on the Web, and several *.cat* sites were seized. That was more difficult for mirrors with domain extensions such as *.eu* or *.org*, because they are often administered outside of Spanish jurisdiction. They were blocked by large Internet service providers (ISPs), such as Telefónica, Vodafone and Orange. (Some small independent ISPs did not cooperate with the blockade.) Sites were blocked by means of DNS tampering, which prevents the domain name system (DNS) server from translating domain names into correct numerical IP addresses. Telefónica also employed deep packet inspection, a surveillance method in which data packets are analysed and filtered out if they meet specific criteria.
These blockades could be bypassed relatively easily. Using social media, Catalan citizens and politicians explained how information could be accessed via the Web despite the blockades. For example, DNS tampering can be avoided by using a public DNS server instead of the ISP's server.

Airplane mode! Airplane mode!

With the raid on the regional IT centre on 30 September, censorship was also imposed on the physical Internet infrastructure. The Centre for Telecommunication and Information Technology (CTTI) is tasked with providing Internet connections in public buildings such as schools and town halls, which are buildings where polling stations are typically located. On the day of the referendum, many polling stations reported problems with their Internet connections. Some of them had no connection at all, while others reported that their Wi-Fi or cable connections did not work. There were also stations where some parts of the Internet were not available.
Here again, citizens joined forces to enable voting. Neighbours opened their WLANs so that the stations could access the online voting system. Hundreds of citizens stood in front of the doors of polling stations to prevent the police from shutting them down. They repeatedly shouted "Airplane mode!". That was a call to avoid using the 4G networks of mobile phones, so that all available bandwidth could be used for voting.

Underground digital infrastructure

A few hours before the start of the referendum on 1 October, the Catalan government announced a 'universal census', which meant that eligible voters could vote at any polling station instead of being limited to the station in their neighbourhood. It also proved to be necessary, because the Spanish police forcefully closed a number of polling stations.
To make a universal census possible, there must be a system in place that prevents individuals from voting more than once. That system consisted of a simple database with the names of everyone who had voted. The universal census was a well-kept secret that had been in preparation for some time. Hackers had been working for weeks already to build the digital infrastructure necessary for the census.
In an interview with the Catalan news site Vilaweb, one of the hackers (who wished to remain anonymous) described how that was done: "Writing a voting program is normally relatively easy. But avoiding the censorship and repression of the Spanish government made it much more difficult. However, we were able to disrupt the efforts of that formidable opponent thanks to Tor, Signal, anonymous phones, Linux, Bitcoin, and open-source software. Not to mention the efforts and creativity of a large number of hackers who did everything they could to make this possible."
At 8 a.m. the system went live at *registremeses.com*. "The domain was blocked within 15 minutes", said the anonymous hacker. "They even blocked all the IP addresses — every one of them — of a well-known European provider. That affected thousands of services that had nothing to do with the referendum. That is arbitrary censorship in its purest form. But each time they did something, we came up with a response. For every IP address they blocked, my colleagues launched two new ones."
Despite the opposition, the census system remained intact during the day. Ultimately 2.2 million of the 5.5 million eligible voters actually voted. Over 90% voted in favour of independence. "The most important thing", as Matthias said in the conclusion of his presentation at the Chaos Communication Congress, "is that the Internet censorship and the repression were not able to stop the referendum." ⏮

(160623-I)

Web Links

[1] A video *of Matthias' presentation is available at* https://media.ccc.de/v/34c3-9028-internet_censorship_in_the_catalan_referendum

[2] https://www.vilaweb.cat/noticies/the-hackers-who-made-possible-a-universal-electoral-register-for-the-referendum/

welcome in your ONLINE STORE

EDITOR'S CHOICE

As long as it's on the bench, a Raspberry Pi can live belly-up, but sooner or later most users will think about putting it in a case. For this purpose there are a multitude of cases available, mostly very similar. At first look, the Pi Desktop case looks like the others, but it offers more advantages than just a simple box to put the famous microcomputer in. It's a little kit that turns it into a proper computer for the (corner of the) desktop. The purpose of the Pi Desktop kit is to build, with a standard Raspberry Pi board, a real versatile computer, with up to 1 TB of mass storage and an on-off button controlled by software. RPi 2 or 3? Doesn't matter, the board is not included in the kit, it's up to you to choose the Raspberry Pi which will be the motherboard for your desktop computer.

Denis Meyer
Elektor Labs

www.elektor.com/rpi-desktop-case

Elektor Bestsellers

1. HDMI/AV Digital Microscope ADSM302
   www.elektor.com/adsm302

2. The Official ESP32 Book
   www.elektor.com/esp32-book
3. Anet A6 3D Printer
   www.elektor.com/anet-a6
4. Electronic Circuits for All
   www.elektor.com/electronic-circuits
5. 10" Touch Display for RPi
   www.elektor.com/touch-display-for-rpi
6. Motor Control Projects
   www.elektor.com/motor-control
7. Python 3 Programming and GUIs
   www.elektor.com/python3

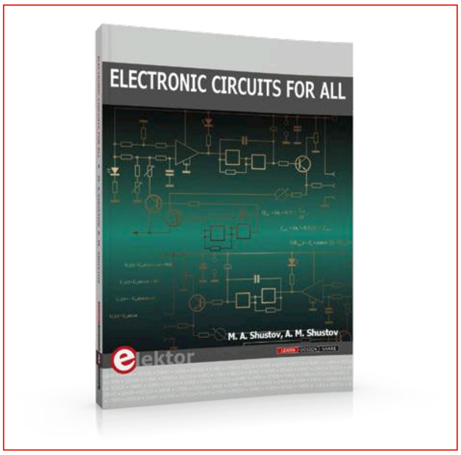
Acoustics in Performance

This book helps those responsible for providing good acoustics in performance and worship spaces to understand the variables and choices entailed in proper acoustic design for performance and worship. Practicing acoustical consultants will find the book a useful reference as well. The level of presentation is comfortable and straightforward without being simplistic.

member price: £19.95 • €22.46 • US $27

www.elektor.com/acoustics-in-performance

Electronic Circuits for All

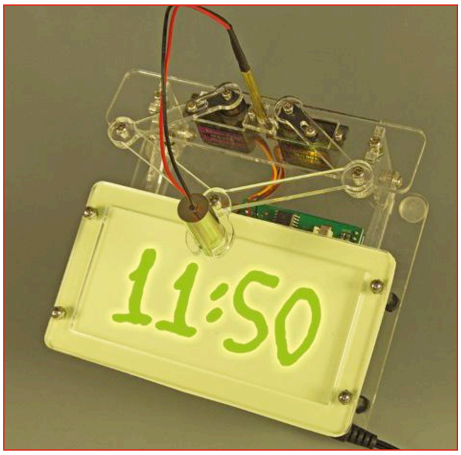


This book includes 400 new and original radio electronic multipurpose circuits. The technical solutions presented in the book are intended to stimulate the creative imagination of readers and broaden their area of thought. The chapters are devoted to power electronics and measuring equipment and contain numerous original circuits of generators, amplifiers, filters, electronic switches based on thyristors and CMOS switch elements.

member price: £31.95 • €35.95 • US $44

www.elektor.com/electronic-circuits

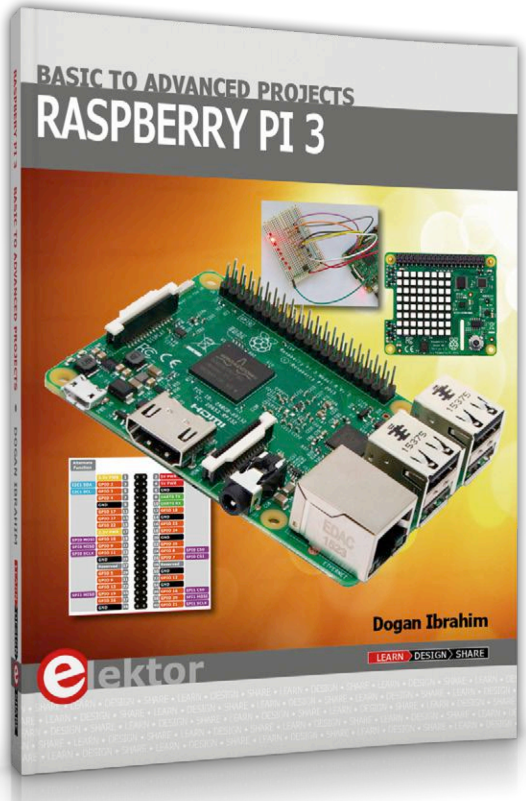
Laser Time Writer



This is an upgrade kit for Elektor's earlier published sand clock. Instead of writing the time in a layer of sand, it uses a laser module to plot the time onto a piece of glow-in-the-dark sticker material. The upgrade-kit comes with all parts required to transform your sand clock into a laser writer: all acrylic parts assembled in a precut plate, glow in the dark sticker material, laser module, pushbutton and wire, brass spacer and small mounting hardware.

member price: £35.95 • €40.46 • US $49

www.elektor.com/laserclock



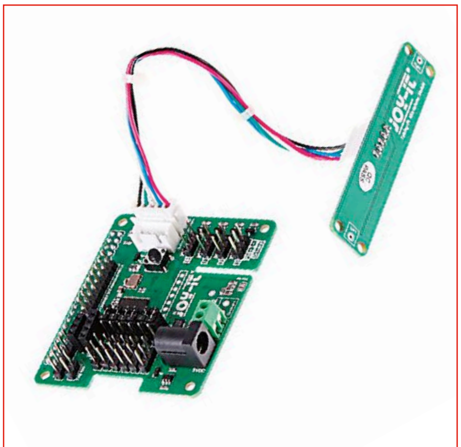
Raspberry Pi 3
Basic to Advanced Projects

This book is about the Raspberry Pi 3 computer and its use in various control and monitoring applications. The book explains in simple terms and with over 30 tested and working example projects, how to configure the Raspberry Pi 3 computer, how to install and use the Linux operating system, and how to write hardware based applications programs using the Python programming language. The nice feature of this book is that it covers many Raspberry Pi 3 based hardware projects using the latest hardware modules such as the Sense HAT, Swiss Pi, MotoPi, Camera module, and many other state of the art analog and digital sensors.

MEMBER PRICE: £26.95 • €29.95 • US $36

www.elektor.com/rpi-basic-advanced-projects

Talking Pi

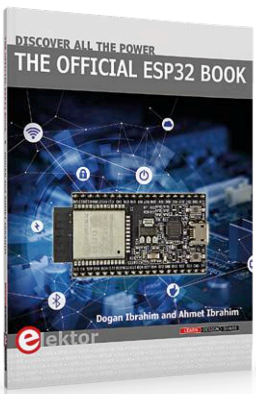


Talking Pi is your intelligent, universal open source speech control assistant for the Raspberry Pi. The extension module is compatible with Google Home / AIY project. Simply use voice commands to control home lighting, switch your power outlets on and off or activate your coffee machine when you open the front door. There are hardly limits to your imagination!

member price: £27.95 • €31.46 • US $38

www.elektor.com/talking-pi

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 book teaches the reader how to use the ESP32 hardware and software in practical projects. 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.

member price: £27.95 • €31.46 • US $38

www.elektor.com/esp32-book

DVD Elektor 2017

This DVD-ROM contains all editorial articles published in Volume 2017 of the English, Dutch, French and German editions of Elektor. Using Adobe Reader, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article.

member price: £21.95 • €24.75 • US $30

www.elektor.com/dvd-elektor-2017

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 March 23, 2018, 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 1/2018 (January & February) is: **7164C**.

The €50 / £40 / $70 book vouchers have been awarded to: Artigue Francis (France); Joe Young (Canada); Jean-Paul Winberg (Belgium); Mguel Án. Gimeno (Spain); Ursula Kronberger (Germany).

Congratulations everyone!

C	3				E	8			9	2		B	A	F	
6	7	9	F				1				4		E	0	2
	8			2		C	F	D						6	9
	1			9	6				F						C
		F	7	1	D				C					3	
A			8	3	4				E						0
1		6					C			A	3	E			F
	C	E				0	5	7					2		
		0					2	8	A				7	E	
B			2	8	F			3					6		5
E						7				9	1	2			3
	F					5				7	0	9	C		
F						1				6	7			D	
2	E						0	A	1		9			C	
5	D	1		E				B				4	0	8	6
	6	8	C		9	4			3	E				2	A

6	7	9	B	C	2	A	0	5	8	D	4	1	F	E	3
8	3	4	A	1	F	5	B	E	0	6	C	9	2	7	D
1	2	5	C	3	8	D	E	7	B	F	9	0	4	6	A
D	F	E	0	4	6	7	9	1	A	2	3	5	8	B	C
9	5	F	D	2	7	1	6	4	C	8	B	3	0	A	E
7	8	B	E	5	A	C	D	9	1	3	0	4	6	F	2
0	A	C	2	8	3	B	4	6	D	E	F	7	1	9	5
3	1	6	4	9	E	0	F	A	7	5	2	8	D	C	B
2	B	0	7	6	9	4	1	8	3	C	E	A	5	D	F
A	4	3	1	E	5	F	C	D	9	7	6	2	B	8	0
C	9	D	F	7	0	8	A	2	4	B	5	6	E	3	1
5	E	8	6	B	D	2	3	0	F	A	1	C	7	4	9
B	6	A	5	F	1	9	2	C	E	4	8	D	3	0	7
4	D	1	3	A	C	E	8	B	2	0	7	F	9	5	6
E	0	7	9	D	4	3	5	F	6	1	A	B	C	2	8
F	C	2	8	0	B	6	7	3	5	9	D	E	A	1	4

Using a 300 MHz 32-bit MCU with 2 MB of buffer memory, the MPLAB® ICD 4 programs at twice the speed of its predecessor. Speed and flexibility are the most important factors when selecting a debugging tool. The MPLAB ICD 4 reduces wait time—and in turn—improves debugging productivity. With speed, compatibility, durability, comprehensive device support and the award-winning MPLAB X IDE, the MPLAB ICD 4 is sure to help you win with your design.

- Debugs twice as fast when compared to the ICD 3
- Robust metal enclosure with easy-to-read indicator light
- Wider target voltage range than the ICD 3
- Optional 1 amp of power to target
- Programmable adjustment of debugging speed for optimised programming
- 4-wire JTAG compatible

MPLAB ICD 4
(DV164045)

microchip DIRECT
www.microchipdirect.com

MICROCHIP

www.microchip.com/ICD4


DO YOU WANT THE BEST
ELECTRONICS DESIGN SOFTWARE
The Alternatives
PROTEUS
User Friendly
Comprehensive
Integrated
Affordable
Danger
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