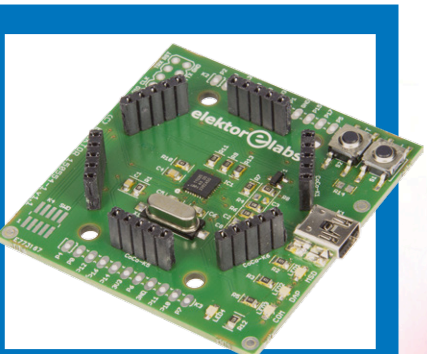
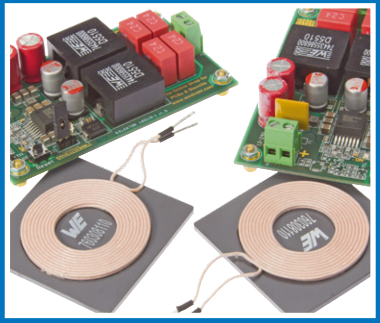
Edition 6/2016 | November & December 2016
www.elektormagazine.com

elektor

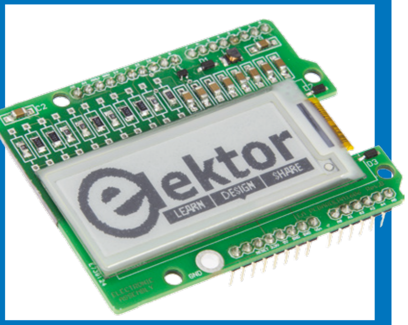
LEARN DESIGN SHARE



Elektor mbed Interface
CoCo-ri-Co & tools
in the cloud



Wireless Power Converter
Transfer 50 watts
through the air

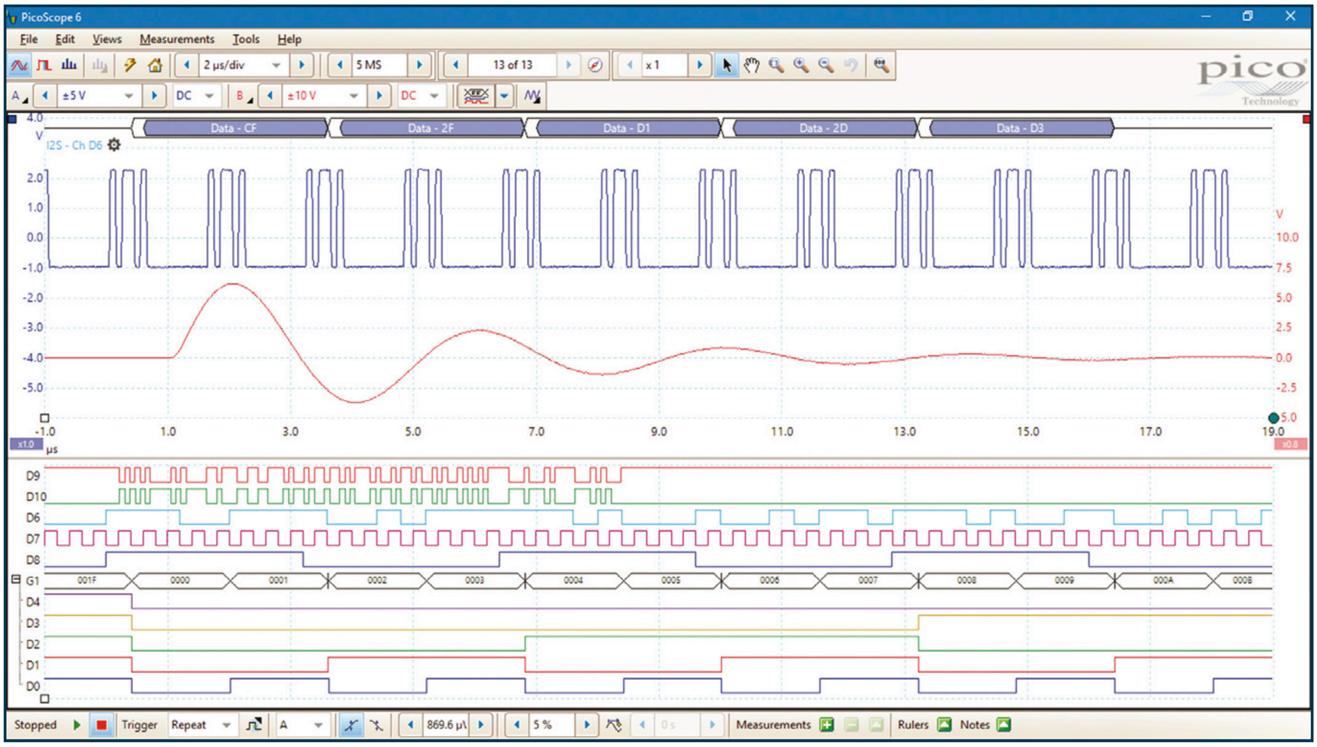


E-paper with Arduino
A sketch for the EPA20-A

Return of the Wi-Fi Controller Board

www-connect objects to your smartphone

Anaren Bluetooth Smart Development Kit • Bandgap Reference • **Bluetooth Controlled Relays** • Capacitive Liquid Detection • **Climate Controller** • **Crimping pinheaders** • **Door Spy using Raspberry Pi** • Elektor Labs Pipeline • Elektor mbed Interface • **Elektor SDR Reloaded (3)** • **ElektorBusiness** • **E-paper** • **Err-lectronics** • **Hexadoku** • Internet Radio with Fluorescent Display • **Minuscule MEMS Microphone** • **PCB Integration of Magnetic Components** • PIC Assembler Course 2.3 • **Programmable LED Circle** • Q & A: CAD • **Retronics: Elektor 36 Years Ago** • **Return of the Wi-Fi Controller Board** • Sensors make Sense (1) • **Simple Class-A Amplifier** • **SmartScope Maker Kit** • **Using E-paper with the Arduino** • Vacuum Fluorescent Displays • **Voltage Tracker for Oscilloscope** • **Wireless Power Converter**


PicoScope 6
File Edit Views Measurements Tools Help
2 µs/div
5 MS
13 of 13
x 1
A ±5 V DC
B ±10 V DC
Data - CF
Data - 2F
Data - D1
Data - 2D
Data - D3
I2S - Ch D6
D9
D10
D6
D7
D8
G1
D4
D3
D2
D1
D0
µs
Stopped
Trigger
Repeat
869.6 µV
5 %
Measurements
Rulers
Notes

Elektor Magazine
Edition 6/2016
Volume 42, No. 479 & 480
November & December 2016

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:
Johan Dijk
Phone: +31 6 15894245
E-mail: johan.dijk@eimworld.com

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

Messe time again

Elektor as a publishing house is often criticized for playing hard to get, being elusive, existing in vapor space, or the editors and lab workers, hiding in ivory towers or at least a Castle. It's true about the Castle (we rent it though) but the working conditions there are sort of mediaeval at times and air-conditioning remarkably is only granted to a few Dell servers in the basement as we discovered the hard way this summer. We do regret the scarcity of personal contact though, so here comes a method to improve that.
At least two times year we are out and about with 27%, 33% or 47% of the publishing staff to meet & greet our readership and talk shop about publications and especially the projects, both of the 'would-be' and the 'all done' variety. It's just a matter of knowing where we are and for that, all you need to do is follow the news in our weekly e-zine, or watch the Calendar on our website. Or turn to page 32 in this edition.
Occasionally we enable you, our reader and member of Elektor community, to have eyeball conversations with many of our staff, including the editors, sales & members' representatives and possibly our CxOs. If you're keen, one occasion not to be missed is *electronica 2016* in Munich on November 8-11 where Elektor has its largest presence ever, thanks to a fruitful cooperation with the Messe Munich organization (where *Messe* is German for trade show and not related to mess(y) in any way). Not being able to find us at the *Messe* I deem a feeble excuse as our stand is definitely among the larger ones there. Everyone's invited — besides free coffee and giveaways we have demos and pitches of our Fast Forward Award nominees, video interviews, tech talk, 5 pm bratwurst, and a complaints desk. Not forgetting the rarest of rare birds to spot there: the makers of this fantastic magazine. They are surprisingly easy to capture in the wild provided the right bait is used like a 470-Ω resistor (see page 32). And amazingly, they talk … electronics!

CU @ *electronica* 2k016

Jan Buiting, Editor-in-Chief

The Circuit

Editor-in-Chief:	Jan Buiting
Publisher:	Don Akkermans
Membership Manager:	Raoul Morreau
Support Executive:	Cindy Tijssen
International Editorial Staff:	Thijs Beckers, Mariline Thiebaut-Brodier Denis Meyer, Jens Nickel
Laboratory Staff:	Ton Giesberts, Luc Lemmens, Clemens Valens, Jan Visser
Graphic Design & Prepress:	Giel Dols
Online Manager:	Daniëlle Mertens

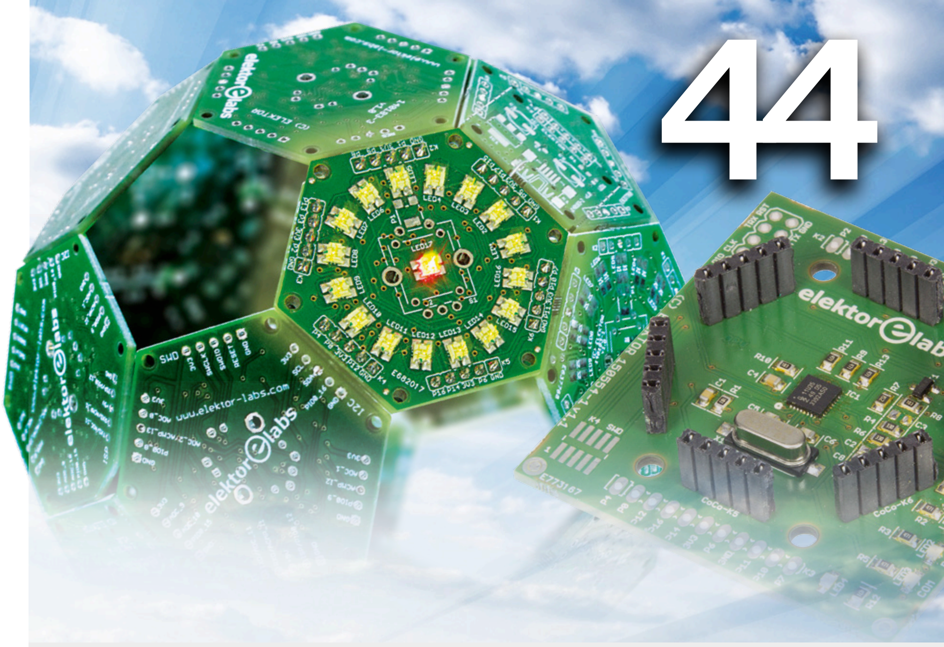
THIS EDITION

Volume 42 – Edition 5/2016

No. 479 & 480 November & December 2016

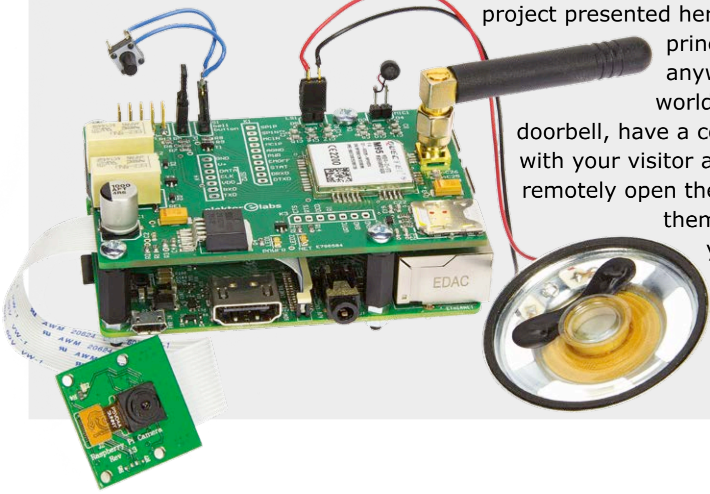
LEARN DESIGN SHARE

LEARN DESIGN SHARE

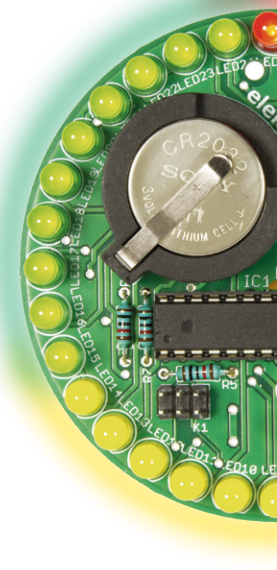


68 Door Spy with Raspberry Pi

An intercom with a camera at the front door is only useful when you are at home. With the project presented here you can, in principle, see from anywhere in the world who rang your doorbell, have a conversation with your visitor and even remotely open the door for them, provided you have your smartphone on you.



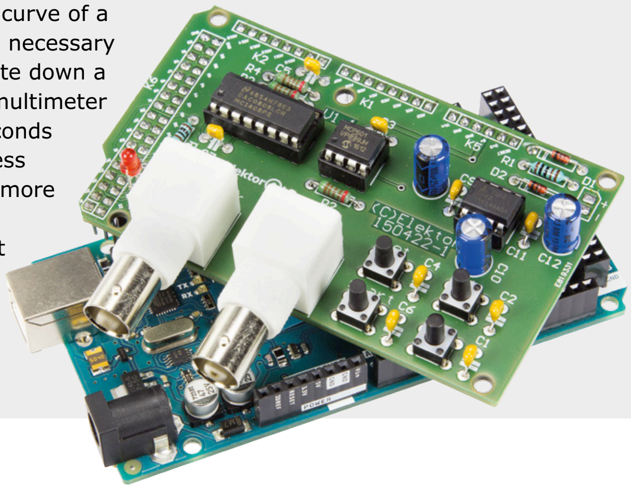
NEW SERIES



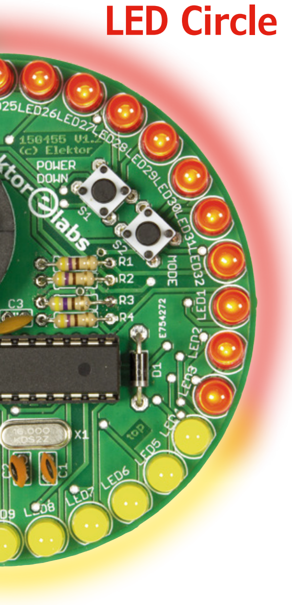
Elektor mbed Interface

An open source library, a free online code editor and compiler, and the mbed fast prototyping platform make excellent tools for developing powerful applications on third-party hardware. But did you know that it is also possible to create your own mbed enabled hardware and add drag 'n' drop programming to your design?

Suppose you want to measure a slowly-varying voltage from a sensor, or perhaps plot the discharge curve of a battery. Is it really necessary to sit there and note down a reading from the multimeter every so many seconds or minutes? It is less tedious and much more elegant to use this Arduino-based unit connected to your oscilloscope.



Programmable LED Circle



Note. We apologize that due to engineering constraints, *Analog Random Numbers* and *My Tiny Radio* could not be accommodated in the current edition.

elektor magazine

LEARN DESIGN SHARE

NEXT EDITION

AxiCat

Many ICs in SMA cases have an I²C or SPI interface, and it would be handy if you could quickly connect them to a Linux or Windows PC to check out their functions before you start using them in an actual design. That's easy with the AxiCat multi-protocol USB adapter, and you don't have to write any code to use it.

ESP8266 on the Elektor Android I/O Board

The Android I/O Board makes it easy to control electronic devices from a smartphone or tablet. There is space on the board for a total of seven different wireless link modules, including the popular and inexpensive ESP8266. In this article we give some background on the module and tell you how to load the firmware.

Chirpie chirpie … Cheepit

Cheepit uses audio files to program ATtiny micros. In this article we describe the programming circuitry to pull off this trick

Elektor Magazine edition 1 / 2017 covering January & February is published on January 10, 2017.
Delivery of printed copies to Elektor Gold Members is subject to transport.
Contents and article titles subject to change.

The Elektor Community

LEARN DESIGN SHARE

Elektor breaks the constraints of a magazine. It's a community of active e-engineers — from novices to professionals — eager to learn, make, design, and share surprising electronics.

Elektor Web Store: 24/7 candy store for every electronics engineer! Permanent 10% discount for GREEN and GOLD Members.
www.elektor.com

Elektor Magazine: Six times per year a thick publication packed with electronics projects, news, reviews, tips and tricks.
www.elektormagazine.com

Elektor PCB Service: Order your own PCBs, both one-offs and larger runs.
www.elektorpcbservice.com

Elektor Weekly & Paperless: Your digital weekly news update. Free.
www.elektor.com/newsletter

Elektor Academy: Webinars, Seminars, Presentations, Workshops and DVDs ... Practice-oriented learning.
www.elektor-academy.com

Elektor Books: Arduino, Raspberry Pi, microcontrollers, Linux and more. Available in our online store with a 10% Member discount!
www.elektor.com/books

Elektor TV: Reviews, timelapse, unboxing and personal journals. Watching is learning.
www.youtube.com/user/ElektorIM

Elektor Labs: Showcasing your own projects and learning from others. We develop and test your ideas!
www.elektormagazine.com/labs

Become a member today!

	GREEN €5.67 per month £4.08 / US $6.25	GOLD €7.58 per month £5.50 / US $8.42	FREE
Elektor Annual DVD	✗	✓	✗
6x Elektor Magazine (Print)	✗	✓	✗
6x Elektor Magazine (Digital)	✓	✓	✗
Access to Elektor Archive	✓	✓	✗
Access to elektorlabs	✓	✓	✗
10% Discount in Elektor Store	✓	✓	✗
elektor weekly e-zine	✓	✓	✓
Exclusive Offers	✓	✓	✓
	www.elektor.com/green	www.elektor.com/gold	www.elektor.com/newsletter

79 Countries

247031 Enthusiastic Members

1034 Experts & Authors

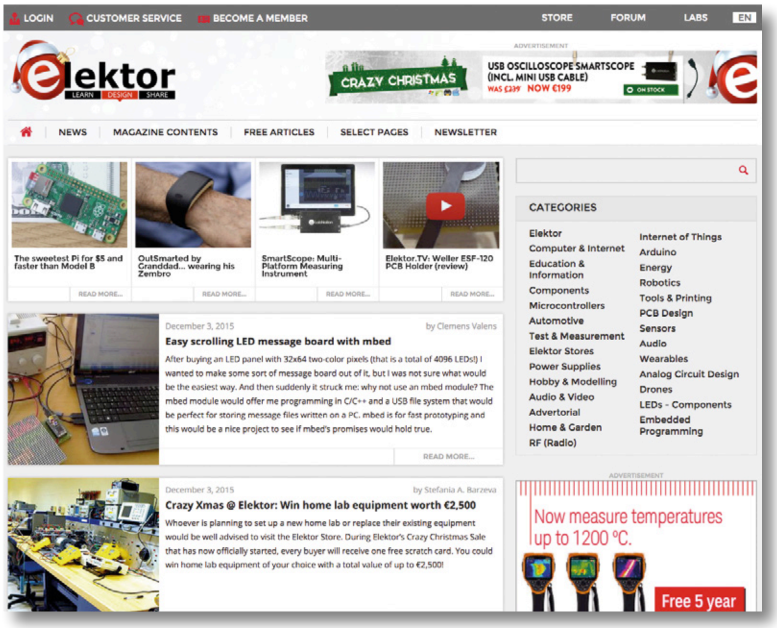
485 Publications

235102 Monthly Visitors

www.elektormagazine.com

A World of Electronics News

Electronics enthusiasts can explore a world of projects, news, and movies on our completely revamped magazine website. Click on the top of the menu to choose the Dutch, English, German or French version, and use the intelligent search tools to find information and articles quickly. Sign up to our community as a GREEN or GOLD Member, and with your personal login details you will have full access to many extras such as special offers and discounts in our online store. You can also manage your account information, including your membership to the printed magazine and the Elektor weekly newsletter.



The sweetest Pi for $5 and faster than Model B

READ MORE...

OutSmarted by Granddad... wearing his Zembro

READ MORE...

SmartScope: Multi-Platform Measuring Instrument

READ MORE...

Elektor.TV: Weller ESF-120 PCB Holder (review)

READ MORE...

CATEGORIES

Elektor
Computer & Internet
Education & Information
Components
Microcontrollers
Automotive
Test & Measurement
Elektor Stores
Power Supplies
Hobby & Modelling
Audio & Video
Advertorial
Home & Garden
RF (Radio)
Internet of Things
Arduino
Energy
Robotics
Tools & Printing
PCB Design
Sensors
Audio
Wearables
Analog Circuit Design
Drones
LEDs - Components
Embedded Programming

December 3, 2015 by Clemens Valens

Easy scrolling LED message board with mbed

After buying an LED panel with 32x64 two-color pixels (that is a total of 4096 LEDs!) I wanted to make some sort of message board out of it, but I was not sure what would be the easiest way. And then suddenly it struck me: why not use an mbed module? The mbed module would offer me programming in C/C++ and a USB file system that would be perfect for storing message files written on a PC. mbed is for fast prototyping and this would be a nice project to see if mbed's promises would hold true.

READ MORE...

December 3, 2015 by Stefania A. Barzeva

Crazy Xmas @ Elektor: Win home lab equipment worth €2,500

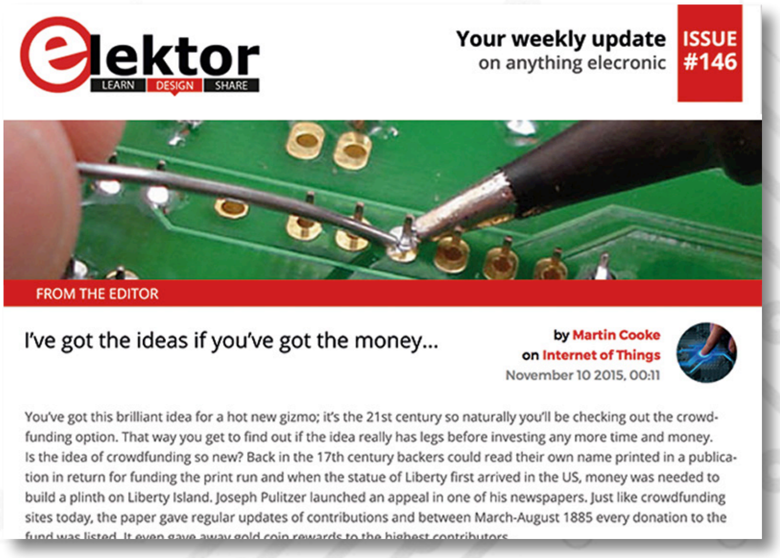
Whoever is planning to set up a new home lab or replace their existing equipment would be well advised to visit the Elektor Store. During Elektor's Crazy Christmas Sale that has now officially started, every buyer will receive one free scratch card. You could win home lab equipment of your choice with a total value of up to €2,500!

elektor Weekly & Paperless

Get a jam-packed Elektor every week

Join the more than 120 K electronics enthusiasts who receive the free & paperless Elektor. Every week you get a selection of news, tips and trends in your email inbox. You will also get special offers and discounts for the online store.

Sign up today:
www.elektor.com/newsletter



FROM THE EDITOR

I've got the ideas if you've got the money...

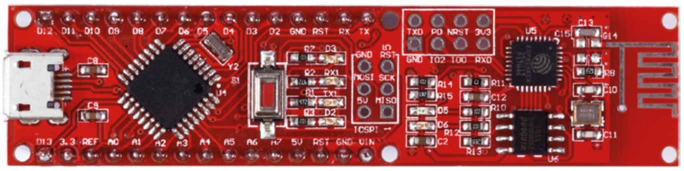
by Martin Cooke
on Internet of Things
November 10 2015, 00:11

You've got this brilliant idea for a hot new gizmo; it's the 21st century so naturally you'll be checking out the crowdfunding option. That way you get to find out if the idea really has legs before investing any more time and money. Is the idea of crowdfunding so new? Back in the 17th century backers could read their own name printed in a publication in return for funding the print run and when the statue of Liberty first arrived in the US, money was needed to build a plinth on Liberty Island. Joseph Pulitzer launched an appeal in one of his newspapers. Just like crowdfunding sites today, the paper gave regular updates of contributions and between March-August 1885 every donation to the fund was listed. It even gave away gold coin rewards to the highest contributors.

Welcome to the LEARN section

By **Jens Nickel**

Finally we are getting there

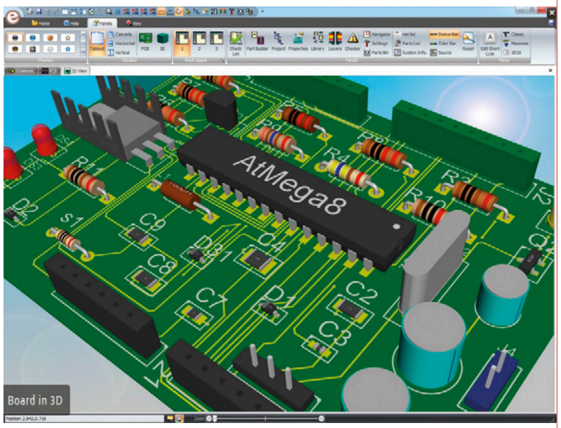


Well I guess I could always blame the unseasonably warm weather in Europe as the reason I have only just managed to submit the eleventh installment of 'My Journey into the Cloud' and it's the third weekend in September already! Anyway enough excuses, you may remember from one of my previous LEARN intros that I managed to write a small program for the PC that controls the ESP8266 WiFi chip on the Pretzel board using AT commands. The chip can establish itself as an MQTT-Client with a Test-broker on the Internet and can also publish messages. However my first attempts at sending a message resulted in a communication disconnect when using the familiar Mosquitto Test-broker and also the HiveMQ-Broker. The reason turned out to be that the ESP8266 automatically appends 'Cr Lf' characters to the end of every message. Without these additional control characters it works fine. After this I managed to change the COM port with a running receive Thread and finally packed up all the software files and made them available online. While I was at it, I almost managed to include my WiFi password in the compilation… With any luck, by the time you read this the 12th installment should also be available online: on www.elektormagazine.com, search: 'Journey'.

The table display arrives

We have already announced in both our magazine and newsletter of Elektor's significant contribution to the hardware development of the *Tabula Project*. The system uses a large touch-sensitive display on which objects or 'Tangibles' (containing electronic sensors) can be positioned. The touch-sensitive display is able to resolve the object's position and orientation on the display surface and the system can serve as a useful teaching aid to help students and pupils understand complex concepts in computing and other science-based subjects. The use of relatively large touch-sensitive screens fitted to table tops will help fully exploit the system's capabilities. The consortium has therefore put on order some of the giant Microsoft Surface Hub-type displays. Colleagues in the HCIC (Human-Computer Interaction Center) at RWTH Aachen, Germany, have already taken delivery of the first of these awesome displays and we will also be receiving one for our lab here in Aachen in the near future.

CAD overload



One of my first jobs here at Elektor — and that was over 10 years ago now — was to translate a really long magazine article. It was a presentation of some CAD programs (mostly demo versions) that we included on a DVD bundled along with the magazine. DVDs are no longer state-of-the-art but mention the words 'free' and 'CAD programs' to any of our readers and you're sure to get an enthusiastic response. In edition 1/2017 my colleagues Harry Baggen and Thijs Beckers will be giving you the low down on the many different software packages available today and publish their findings in the LEARN section. Just as the editors did ten years ago, they will also be looking at the strengths and weaknesses of the different suites on offer. I'm curious.

(160174)

Bandgap Reference

Peculiar Parts, the series

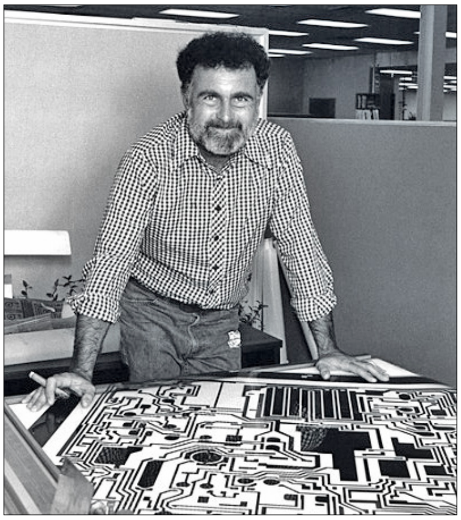


Figure 1. Robert John (Bob) Widlar (1939–1991), fulltime prankster and Silicon Valley's *enfant terrible* for many years.

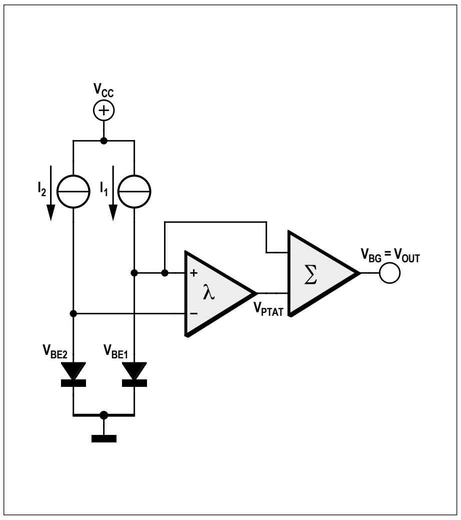


Figure 2. Bandgap voltage reference circuitry (source: Texas Instruments)

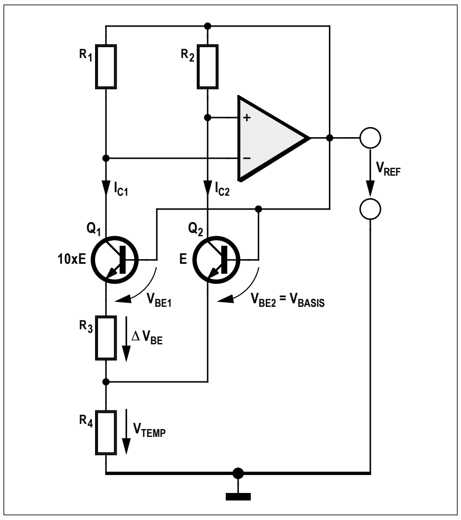


Figure 3. Paul Brokaw's improved bandgap reference.

By **Neil Gruending** (Canada)

Voltage references are a necessary part of voltage regulators, data acquisition circuits and a variety of other analog circuits. A special class called bandgap voltage references is typically used for their temperature compensated high accuracy performance and since they are hidden inside of an IC most people don't give them a second thought. But since this is *Peculiar Parts* I thought this would be a fitting opportunity to take a closer look at them.

One of the early methods for an accurate voltage reference was to use a neon lamp. Once ionized (lit) the voltage across them would be about 48 to 80 V depending on the lamp. They didn't need a lot of current but if the voltage dropped too far then the lamp would stop ionizing and would require a higher (66 to 200 V) voltage to start ionizing again. However, even with these limitations lamps were becoming available that could typically regulate their voltage to about ±0.5 V by 1966. But neon lamps didn't last very long before they were replaced by zener diodes. These shunt regulators achieve much higher accuracies by taking advantage of the breakdown voltage of reverse-biased PN junctions for their regulation. They are also available in a wide range of voltages, making them much more flexible. In the 1970's zener diodes were also being used as the voltage reference to make voltage reference ICs. To help maximize their performance the diodes were actually sandwiched (buried) in the die which helped reduce their intrinsic noise levels.

One very much needed part was a monolithic voltage regulator but Zener references weren't working. In fact, National Semiconductor engineer Bob Widlar (**Figure 1**) even wrote an EEE article arguing that a monolithic voltage regulator was impossible to build because the power transistor had too many negative temperature effects on the die and reference. And then one year later in a twist, Bob published the circuit for the first monolithic voltage regulator, the LM109. His solution for all of the thermal effects on the reference was the bandgap reference like the one shown in **Figure 2**.

The circuit works by biasing the diodes at different currents to create a Proportional-To-Absolute-Temperature (PTAT) current. This is the point where the diode temperature coefficients cancel each other out which automatically temperature compensates the output voltage (1.2 V).

Paul Brokaw tweaked and improved it even further in 1974 by using feedback to improve accuracy like in **Figure 3**. It uses negative feedback amplifier to equalize the transistor collector currents because when the currents are equal the transistor temperature coefficients will exactly cancel each other out. The trick is that the ratio of the transistor emitter areas is what defines this behavior.

Hopefully this gives some insight into bandgap references and how they came to be. They are certainly an important building block in many circuits.

(160074)

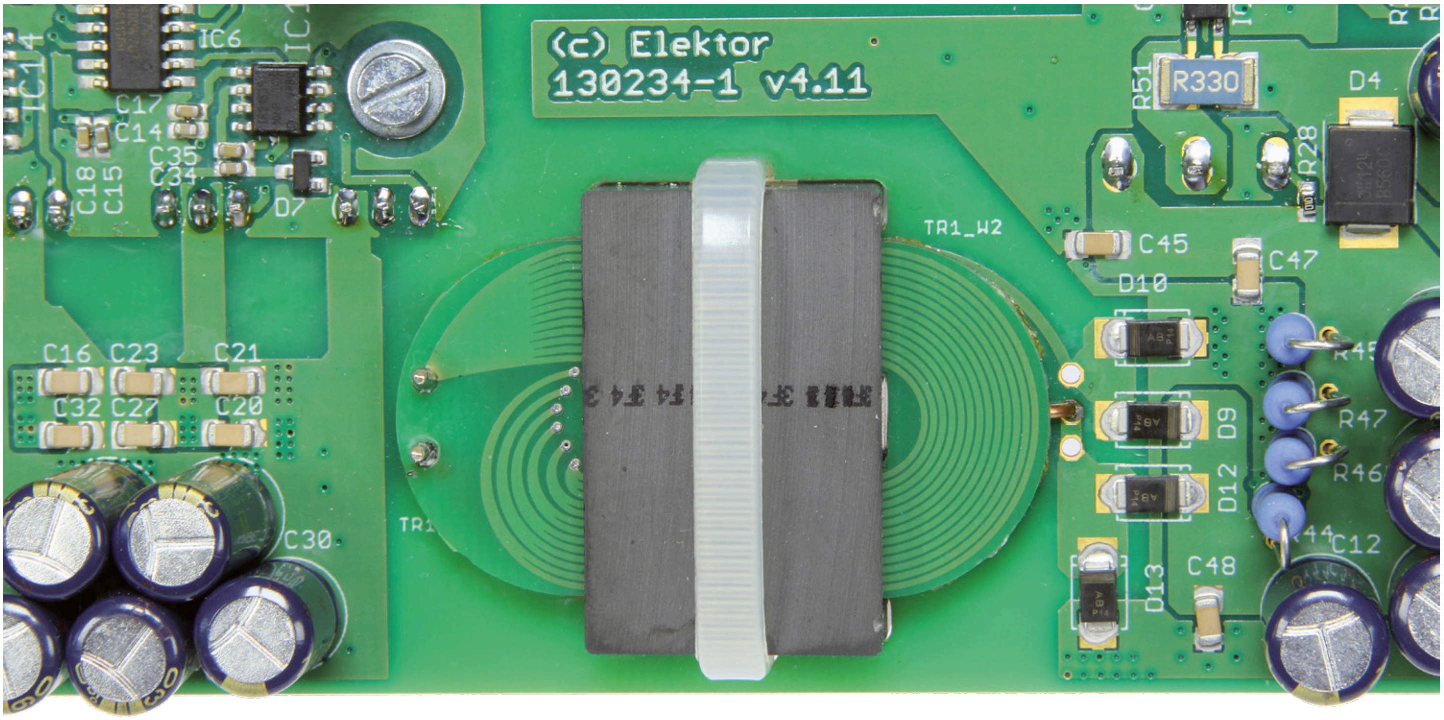
Please contribute your Peculiar Parts article, email neil@gruending.net

PCB Integration of Magnetic Components

Principles and simulation

By **Martin Christoph** (Germany)

Coils and transformers are integral to the field of electronics. The Professional Lab Power Supply project featured in the September 2014 edition Elektor uses a special transformer. It's a planar type where the windings are made from PCB tracks. Here we look at the structure of these electromagnetic components and simulate their performance.



There are already many transformers and inductors available off-the-shelf to meet the most common requirements. It is however often the case where you have a specific requirement and there just isn't a transformer available that meets the specification. The transformer characteristics, particularly power handling, turns-ratio and operating frequency can be very specific. When we get down to subtleties such as the affects of stray inductance on the transformer's performance then there's no alternative but to design the transformer from scratch. This so called 'PCB integrated magnetic component' can be tailored to the application in the same way that its more conventional alternative can be. This method of construction does however place certain constraints on the transformer shape and winding arrangements because of the structure and manufacturing process of the PCB. A typical PCB is a relatively thin flat-plane, not at all conducive to coil construction. We will go on to show how—by careful design—we can overcome some of the shortcomings.

The advantages

As we mentioned earlier, the Professional Lab Power Supply [1] requires a special transformer to be built. The first prototype was quite conventional and wound by hand. Looking at the finished design it was clear that building the transformer would represent the biggest hurdle for constructors. Firstly it's necessary to find the correct gauge of wire and then you need to carefully make the windings in accordance with the prototype in order to achieve the same performance. The main advantage of a transformer with windings etched into the PCB is that it ensures good reproducibility and predictable performance.

This type of transformer also has lower production costs which is a benefit to both mass produced and homebrew designs. Gone are all the spools of different gauge enameled-copper wire cluttering up your work bench; equivalent results can be achieved with careful PCB design.

A further advantage is improved insulation properties. Multi-layer boards can achieve particularly good levels of insulation: windings can be contained within the inners layers using 'buried vias' to make the interconnections. The outer layers are therefore free of vias and provide the necessary insulation and attendant voltage 'creepage' clearances. Windings contained within the PCB are well insulated from any other windings and also from the core material. Commercial designs using 1.7-kV IGBT switches are built using this technique.

Transformers that use windings integrated into a PCB can produce very low stray inductance; the primary and secondary coils can be built directly over one another. The coil coupling factor is therefore very high. Low stray inductance is an advantage especially when the transformer uses a high switching frequency.

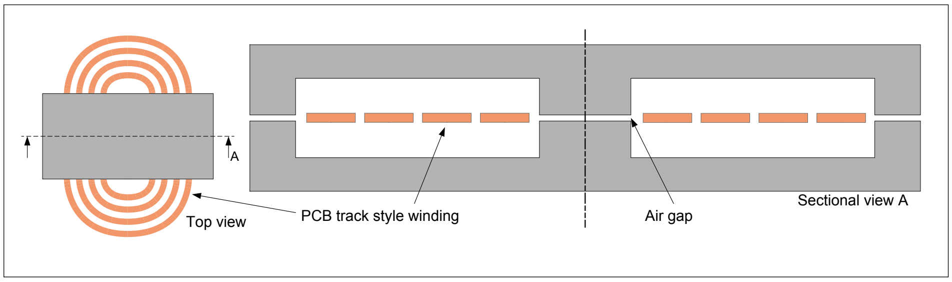


Figure 1. Schematic representation of a PCB-integrated coil.

Some special properties

To illustrate the principle **Figure 1** shows the cross-section through a simple coil. It is made up of two E-core halves with a printed coil on a PCB. The coil core also has an air gap. The coil is symmetrical so we only need to consider a cross section through one half of the windings.

The upper picture in **Figure 2** represents the undesired magnetic flux leakage in one half of the windings which is not contained completely in the core material but leaks out into the coil winding space. It is unavoidable flux leakage from the core (particularly around the air gap region) which can induce current in adjacent wound components and affect inductors on the PCB. The lower sketch in **Figure 2** shows how this induced current flows in opposite directions on the left and right side of the conductor. This current is summed with main current flow through the conductor and produces a net resistive loss. The effect of the current increases with the stray fields and at higher operating frequency which also have a big influence the voltage induced in the conductor.

This problem should not be confused with skin-effect which describes how current flow in a conductor migrates to the conductor surface as the signal frequency increases. This is more to do with the effect of stray fields on the conductor (see Figure 2) and the Proximity effect, i.e. the influence of one current-carrying conductor on a neighboring conductor. The skin-effect has less influence on printed coils because they are relatively thin and therefore well suited to low-loss operation at high frequency.

The losses become greater, as the area of the conductor permeated by the flux increases. The use of narrow but thick conductors as shown in **Figure 3** is an ideal profile to reduce this effect. This would however require a very thick copper layer which would be both expensive and impractical. All is not lost, by careful design; losses in the conductors can be minimized. Using conventional magnetic circuit construction methods the winding in Figure 3 can be reproduced easily with the use of foil tape.

FEM simulation

A FEM (Finite Element Method) simulator allows closer examination of the effects of inductance. While this may sound a little complicated at first simple examples can be relatively easily modeled. As already described the coil arrangement is symmetrical. There is no magnetic flux on axis of symmetry and it is only necessary to simulate one half of the coil. With the conductor in the middle then you would only need one quarter.

Assuming that the relevant area of interest is underneath the core material this results in a planar arrangement of a defined depth. The section through the core can be simulated with the core depth as a parameter. For this we can use the free program called FEMM which can be downloaded, together with the documentation from [2]. The following step by step guide will take you through the simulation procedure. The actual parameters used in this walk through are not so important; it should just give you an appreciation of how the program can be used.

Following download and installation start the program and select *File* and then *New,* a dialog box headed *Create a new problem* will appear where you can select *Magnetics Problem* and click *OK*.

Now we get to the pre-processor page where we can enter the data. Click on *Problem* on the top bar and you will see a dialog box where you can enter the parameters. For *Problem Type* choose *Planar,* then choose millimeters as *Length Units* and for *Frequency* enter 100 kHz for example. For *Depth* enter 20 mm for a E32/6/20 core.

The group of five buttons on the left of the taskbar allows you to specify the type of object geometry (*Node*, *Segment*, *Arc Segment*, *Label* or *Group* – see also the *Operation* option on the top bar). Here a left mouse click places the object and a right click selects the object. When you

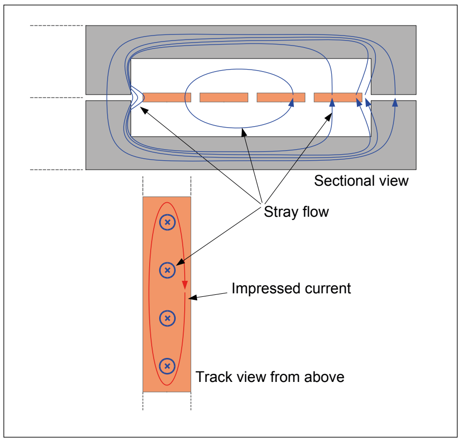


Figure 2. Stray flux and the resulting induced current.

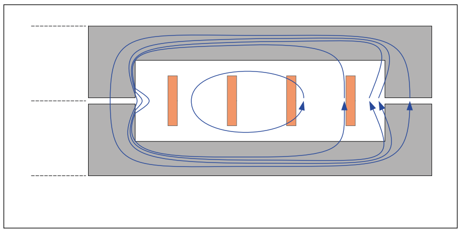


Figure 3. Conductor arrangement to minimize the influence of stray flux.

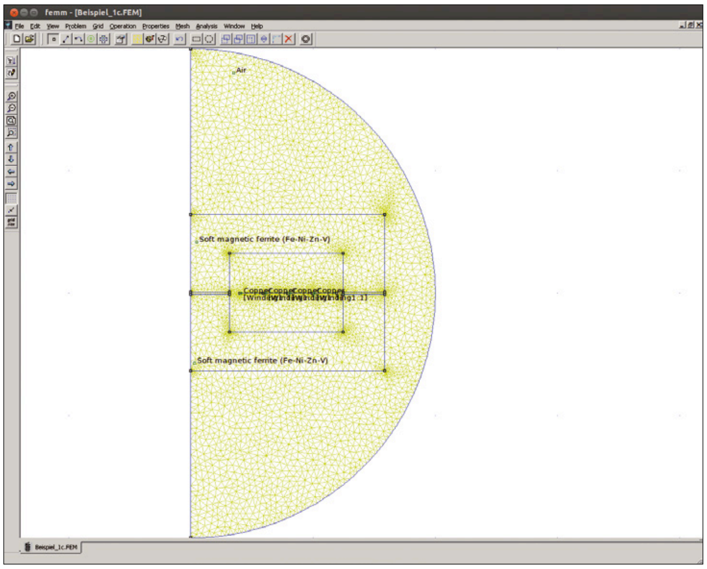

femm - [Beispiel_1c.FEM]
File Edit View Problem Grid Operation Properties Mesh Analysis Window Help
Air
Soft magnetic ferrite (Fe-Ni-Zn-V)
Soft magnetic ferrite (Fe-Ni-Zn-V)
Beispiel_1c.FEM

Figure 4. Represented in FEMM.

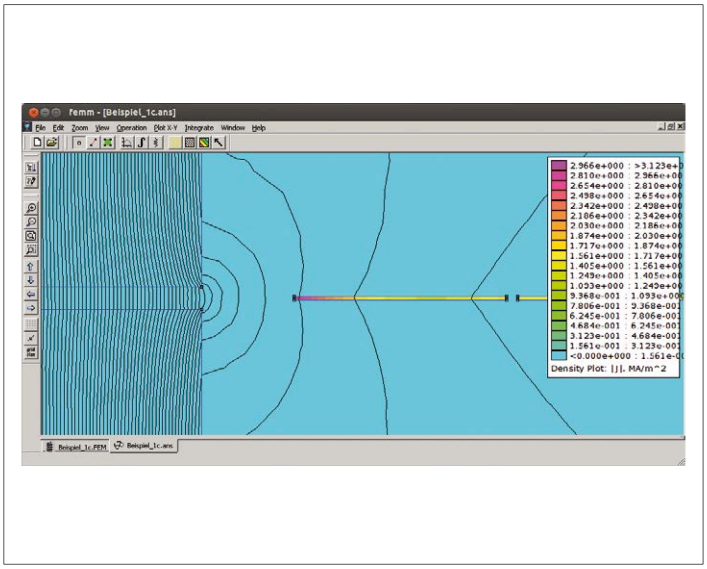


Figure 5. Simulation: Lines = magnetic flux and color = current density.

select an object you can create new copies, select it with a right click and define the properties of the object by hitting the space key or tab (with *Node*, *Label* and *Group* objects) to place an object. The choice of keys in this program is a little random but with practice you soon get the hang of it.

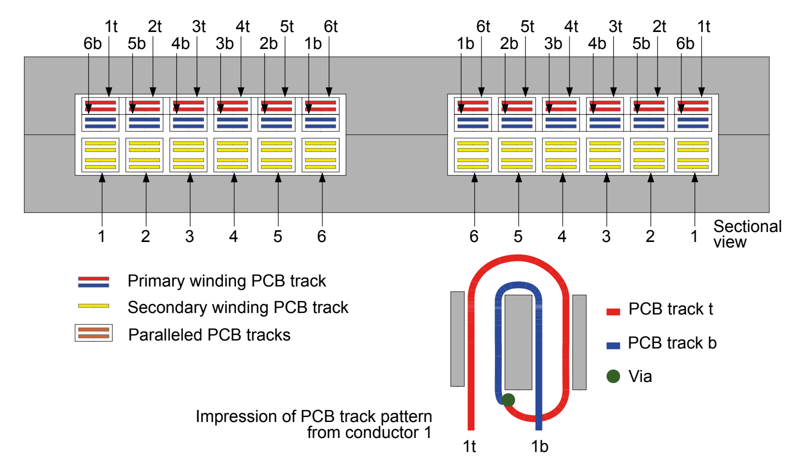
Firstly we need to define a boundary for the problem. Assume that no magnetic flux leaves the core; it will give a false result for the outer air gap. The structure is surrounded by air. Defining a half-circle with appropriate boundary properties avoids the definition of an infinitely large space. Select the *Operate on nodes* button and then bring up the coordination input table with the tab key, enter ‘x = 0’ and ‘y = -20’. Hit the tab key again to define another Node this time enter ‘x = 0’ and ‘y = 20’.

Now select the *Operate on segments* button and place a vertical line first selecting the lower node then the upper node. Next select *Operate on arc segments*, and click both nodes again, this brings up the *Arc segments properties* box. Enter 180 degrees as the *Arc Angle* and click OK to produce the semicircle. A further semicircle can be built up in the same way.

The program can also be controlled by the appropriate script from Matlab or Octave. The user handbook gives more detail of this method.

The winding can be drawn as shown in Figure 1. For the purposes of experimen-

Design of the Lab Supply PCB integrated transformer



The finished PCB for the Elektor Professional Lab Supply is the result of various considerations, discoveries and refinements and has not been subjected to a complete simulation process. The picture shows a cross section through the transformer windings. It can be seen that many of the windings are connected in parallel to cope with the designed current flow. The secondary winding shown in yellow consists of six turns and do not use wide tracks in the four PCB layers but instead use narrow tracks connected in parallel in all four layers. The four parallel paths can be constructed without vias and have their start and end connected together. The narrow tracks help reduce the influence of stray fields. All four tracks are positioned directly over each other and span the same space so they are subjected to the same current level to produce equal voltages in each winding. This ensures that the current flow is shared equally between the windings.

For the primary winding by swapping over the individual conduction paths and connecting their ends in parallel each track consists of the same total area. Each conduction path is subject to almost the same magnetic flux which induces the same voltage and ensures the current is equally shared. Since resistive losses have a quadratic relationship to current, equal current division results in the lowest losses. A close inspection of the finished transformer shows that one turn of the winding divides up and then rejoins. This is due to a limitation in the design process; in the end there wasn’t enough space for all the vias. Electrically not optimal but then again reality is often also less than ideal…

tation precise measurements are not so important but for information: a standard PCB track is 35 µm high.

Using information in the data sheet for the E32/6/20 core [3] you can model (half) of the core geometry. For the air gap between the two core halves you can use a value of 0.2 mm for example. Selecting *Properties → Materials Library* allows you to choose the materials. Now drag and drop properties from the left field into the 'Model Library' on the right. For our purposes here we just need *Air*, *Copper* and *Soft magnetic ferrite*.

Now from *Properties → Circuits* you can define a circuit. With *Add Property* we can define a circuit called 'Winding1' selecting a *Series* configuration with a current of *'0.1'* for 100 mA.

Now *Block Labels* can be placed in all areas enclosed by a line and selected with a right click. The space bar brings up the corresponding Properties-Dialog, where the corresponding Material is selected. For the winding and *in circuit* we select 'Winding1' which we defined earlier and for the *number of turns* enter 1.

Now the boundary conditions need to be set for the semicircle we defined at the beginning. The coil section is symmetrical so the vertical line is defined to pass through the centre of the assembly with 'A = 0'. Under 'Properties' choose *Boundary* and select *Add Property*. Now in the *Boundary Property* the *Name* 'A=0' is entered for example and for *BC Type* 'Prescribed A' chosen. The *Prescribed A parameters* all remain '0'. After closing the dialog the vertical line of the semicircle is selected with a right click in *Operate on segments.* Now using the space bar choose 'A=0' in the pull down list in *Segment Property*.

The semicircular boundary should behave as if there were an infinite amount of air or unbounded. This can be approximated using different boundary conditions than those we assigned previously. Here with the *BC Type* choose 'Mixed' and enter the following coefficients:

$$c_1 = 0; \quad c_0 = \frac{1}{\mu_0\, r_0} = 39788735.77$$

$$\textit{where} \quad r_0 = 0.020\,\mathrm{m}$$

You can find this formula on page 155 of the FEMM handbook.

These boundary conditions are assigned by selecting *Operate on arc segments* then right clicking and hitting the space bar.

Table 1. Power loss comparison

Arrangement	Power loss [mW]	Compared with 1
1	129.634	±0 %
2	126.359	–2.5 %
3	114.508	–11.7 %

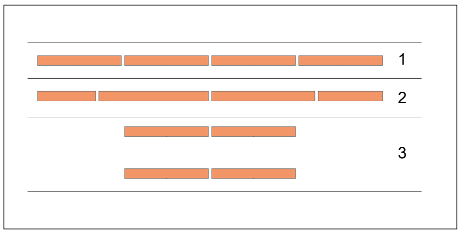


Figure 6. Three track arrangements, each uses the same amount of copper.

Now everything has been specified you can click on the yellow *Run mesh generator* in the tool bar. When no error is generated you can click on *Run analysis*. Any problems encountered are most often caused by some part of the geometry not being closed or by a region where the material has not been defined. All of the closed regions in the half circle must have a material defined, even the air space. The mesh generator should produce the output shown in **Figure 4**.

The FEMM files for the three different arrangements are available for free download from [4].

The Results

With *View results* you can display all the results. Using *View → Density plot* you can, for example with *|Flux Density| (T)* give a representation of the flux density in the core. From this you will be able to see if the core is driven into saturation. Selecting *|Current Density| (MA/m^2)*, shows the current density in the conductor. Here you can see that it is not evenly distributed but influenced by stray fields. This is particularly evident in the region of the air gap. Increasing the operating frequency shows the effect becoming more pronounced. **Figure 5** gives the simulation results in the conductor at a frequency *f* of 100 kHz.

Using the *Contour* mode you can make arbitrary contours in the solution region and show X-Y plots of the field values on the contour. This allows you, for example to place a line through a conductor to show the current gradient.

Also interesting here is *View Circuit Props*. This shows the characteristics of the circuit such as inductance and power losses. Power loss information allows you to optimize a design by indicating which arrangement is more efficient. **Table 1** shows results for the three different conductor arrangements shown in **Figure 6** which all use the same amount of copper. It shows that equal-width tracks are the most unfavorable arrangement. Using the tracks with unequal width the losses fall by 2.5 %. Even better is the configuration using tracks on the underside and topside with a wide unused border. This arrangement however doesn't allow the connection to the center of the winding to be brought out via the other layer on a double-sided board. For a >10 % reduction in power losses the addition of a wire jumper has got to be worth it.

Conclusions

This article gives just a small insight into the optimization techniques of magnetic components. It does show how making even small changes to a design can result in lower losses. It should be clear that magnetic components integrated into the PCB may not be the best solution for every application. When you consider the complete design you can often arrive at a good compromise for their use. They are a good choice where an inductor will be operating with a high level of DC current (such as in a switch regulator with low ripple current or as a filter in a DC supply) because effects from the eddy currents will be so low. Where high levels of AC current are required it is important to pay attention to the coil design. With high levels of stray fields it is not a good idea to have a lot of copper in the region of the air gap. Transformers will generally not have any air gaps so the negative effects will not be so evident. Their very low values of stray inductance make PCB-integrated transformers an interesting alternative for high frequency applications. These compact solutions can be quite easy to make.

(130487)

Web Links

[1] www.elektor-magazine.com/130234

[2] www.femm.info

[3] www.farnell.com/datasheets/1632190.pdf

[4] www.elektor-magazine.com/130487

E-paper

Is it The Future of Displays?

By **Prof Dr Dogan Ibrahim** (Cyprus)

Conventional backlit flat displays work with the principle of emitting light. E-paper or electronic paper (also called EPD: Electronic Paper Display) are display devices that reflect light just like ink on paper. Just like ordinary paper, the characters on an e-paper are visible even in direct sunlight. E-papers are perfect displays since the image remains fixed even after the power is switched off, thus consuming extremely low energy. In this short article we will be looking at the history, operating principles, advantages and disadvantages, and the future of e-paper displays.

The history

The first e-paper, called the Gyricon rotating ball display, was developed in 1970s by a team led by Nicholas K. Sheridon at the Xerox's Palo Alto Research Center (PARC). The idea was to create the concept of paperless office by developing a display that resembles real ink on paper.

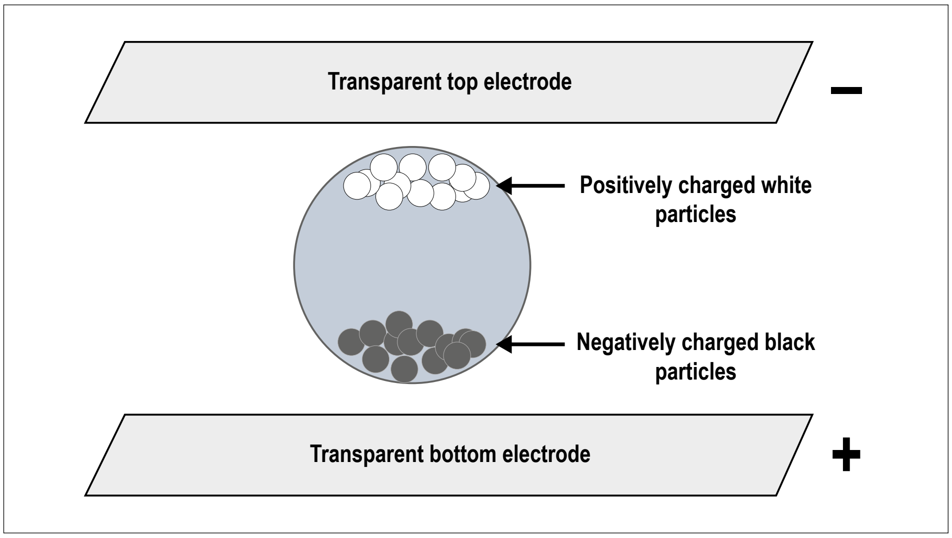

Transparent top electrode
−
Positively charged white particles
Negatively charged black particles
Transparent bottom electrode
+

Figure 1. The principle of e-paper.

The idea of Gyricon was simple as it consisted of a very thin double sided plastic sheet filled with oil pockets. Millions of two colour tiny balls were placed within the plastic sheet. These balls rotated when exposed to an electric field. When rotated fully they displayed as black (or white) and when rotated partially they displayed shades of grey. One very interesting feature of this design was that the displayed text or image remained fixed in position until changed by another electrical field, thus requiring extremely low power for its operation. The image remains fixed when power is switched off, enabling the display to run for years off a single battery.

The initial Gyricon displays were expensive and had poor resolution and were mainly used in message display systems. As a result, the Gyricon project was closed in 2005 by Xerox due to financial reasons. Since then many companies such

> It is expected that the e-paper technology will lead to true paperless offices.

as Sony, Hewlett-Packard, IBM, Fujitsu, Hitachi, Siemens, have worked and developed various forms of e-paper. Today one of the great inventions in this field is the development of true color e-paper.

How does E-paper work ?

The present day e-paper is based on the e-ink technology. So, what is the e-ink technology? E-ink technology consists of millions of tiny ink capsules filled with tiny particles, with the particles inside these capsules having different colors and different electric charges. In an e-paper millions of such capsules are used within a very thin film and electrodes are placed at the top and bottom of this film. As shown in **Figure 1**, when an electric field is applied to the film, the charged coloured particles will move either to the top or bottom depending upon the polarity of the applied field, and cause the surface of the e-paper to display text or an image by changing in color, e.g. different shades of grey in the case of black and white display.

There are two important properties that make e-paper very attractive: the image displayed by the e-paper remains until erased, just like the writing or sketch you draw in a notebook. The e-paper requires power only when a new image is displayed, which means that no power is required to display the image as long as the image is not changed. Thus, for example, while reading a book on an e-paper, power will only be required when you flip a page. The other attractive feature of the e-paper is that the image is reflective and does not require any back lighting, just like the ink on a traditional paper. This feature makes it easier and more comfortable for the eyes as it also provides a wider viewing angle, no glare, and no light pollution to the environment.

Use of the e-paper

Because of the many attractive features, the e-paper is a perfect candidate for use in indoor and outdoor displays. Possible indoor application areas are:

- Mobile devices such as e-readers (e.g. Amazon Kindle, see **Figure 2**)
- Message displays in shops and public places
- Shelf labels in stores
- In hotels and restaurants
- Notice boards in public government offices
- Indoor notice boards such as in museums

Possible outdoor application areas are:

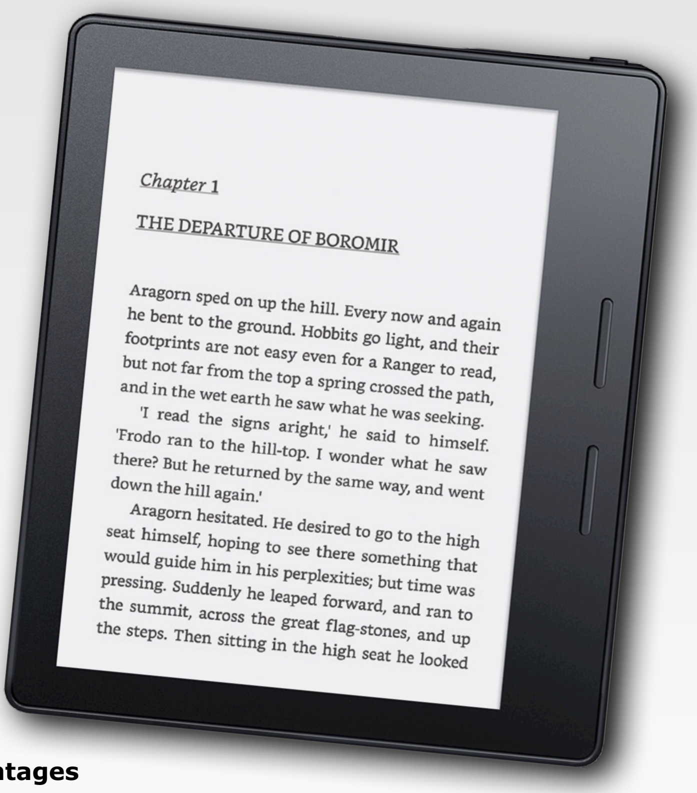
- Traffic signs
- Bus stop and train time table signs
- Parking signs
- Public notices
- Advertisements

Advantages and disadvantages of the e-paper

E-paper display technology has many advantages, some of which are listed below:

- Ultra-low power consumption
- Image like ink on paper
- Reading even under direct sunlight
- Flexible construction

Perhaps currently the only disadvantage of the e-paper technology is its response time.



Chapter 1

THE DEPARTURE OF BOROMIR

Aragorn sped on up the hill. Every now and again he bent to the ground. Hobbits go light, and their footprints are not easy even for a Ranger to read, but not far from the top a spring crossed the path, and in the wet earth he saw what he was seeking.

'I read the signs aright,' he said to himself. 'Frodo ran to the hill-top. I wonder what he saw there? But he returned by the same way, and went down the hill again.'

Aragorn hesitated. He desired to go to the high seat himself, hoping to see there something that would guide him in his perplexities; but time was pressing. Suddenly he leaped forward, and ran to the summit, across the great flag-stones, and up the steps. Then sitting in the high seat he looked

Figure 2. Kindle e-paper based e-reader.

Comparing the e-paper with LCDs

The LCD displays have been in use over several decades in almost all electronic devices where human monitoring and interaction may be required. **Table 1** shows a comparison of the e-paper and LCD displays. Notice that this table considers only the character based LCDs. Basically, the choice between an e-paper

Table 1. Comparison of the e-paper and character LCD displays

E-paper	Character LCD
180º viewing angle	60º viewing angle
Easy to read (even under direct sunlight)	Reading depends on the environment
No glare	Can be prone to glare
Ultra-low power operation	Medium power operation
Excellent contrast ratio	Bad contrast ratio
Very thin	Medium thickness
Can be flexible	Cannot be flexible
Low cost	Medium cost
Can be very large size	Limited size
No backlight	Backlight required
Ideal for e-reading	Not suitable for e-reading
Steady image	Image may be flickering
No eye strain	Can give eye strain

and an LCD display depends on where the display is to be used. In applications where it may be required to stare at the display for long times, such as in e-readers, the choice is definitely the e-paper since the image is sharp, has high resolution, wide viewing angle, and is similar to the print on a paper which is friendly both to the eye and to the environment. In addition, there are many mobile device applications where long battery life is a major requirement. Again, the e-paper is a winner in such applications with a single battery lasting for years. Perhaps another two areas where the e-paper is a winner are in the design of very large message displays, and in different shape and thickness flexible displays (see **Figure 3**).

Coloured e-paper

Perhaps the biggest breakthrough in the e-paper technology is the development of true bright colour e-paper. Color was initially introduced into e-paper by using filter arrays on top of the black and white e-papers, but this had reduced resolution and contrast. E-paper with true bright colours is now available without the use of filters, producing sharp full color reflective displays. Full colour e-paper technology makes use of four different pigments of cyan, yellow, magenta, and white. What is innovative here is that the ink particles differ in sizes and the amount of charge they carry so that different particles move depending on the strength of the applied voltage, thus the combination of different coloured particles produce the required color at the surface. It is estimated that up to 32,000 colors can be displayed with this technology. **Figure 4** shows an example of how the colors of white, cyan, and green can be displayed by combining the available pigments at the surface of the display.

Example e-paper

An example e-paper display is given here to make the reader familiar on how these displays can be used in microcontroller based applications. The display chosen here is the E1271CS021 from Densitron (www.densitron.com). As shown in **Figure 5**, this is an e-paper having dimensions 70.42x45.8x1mm with a resolution of 264x176 and pixel pitch of 117 dpi. The cost of the display is around £18 which is highly competitive compared to LCD displays of similar size.

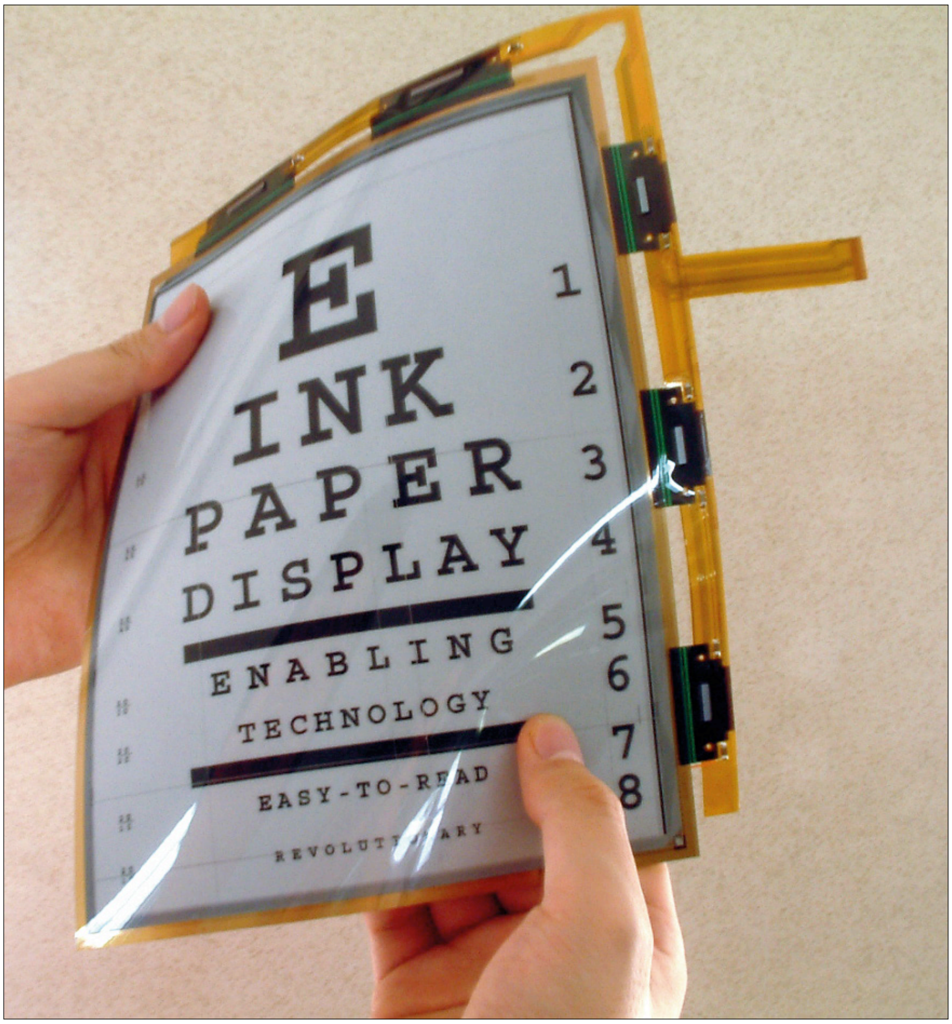


Figure 3. Flexible e-paper.

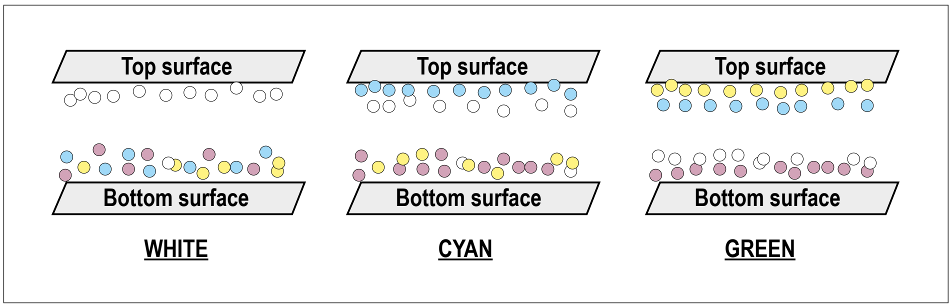


Figure 4. Displaying colour on e-paper.

Web Links:

www.waveshare.com/4.3inch-e-paper.htm

www.pervasivedisplays.com/kits/epd_xpro

www.embeddedartists.com/products/displays/lcd_27_epaper.php

www.adafruit.com/products/1316

www.seeedstudio.com/depot/eink-display-shield-p-1374.html?cPath=132_134

www.eink.com/esl_tags.html

www.conrad.com/ce/en/product/1311174/PCB-design-board-Renesas-YLPDSKRL78EINK

Color Inkjet All-In-One
$179.90
• Display panel Smart 7.8" touch panel & 3.5" LCD
•Built-in wireless and Ethernet networking Quickly
• print from your mobile device

Figure 5. E1271CS021 e-paper display (www.densitron.com).

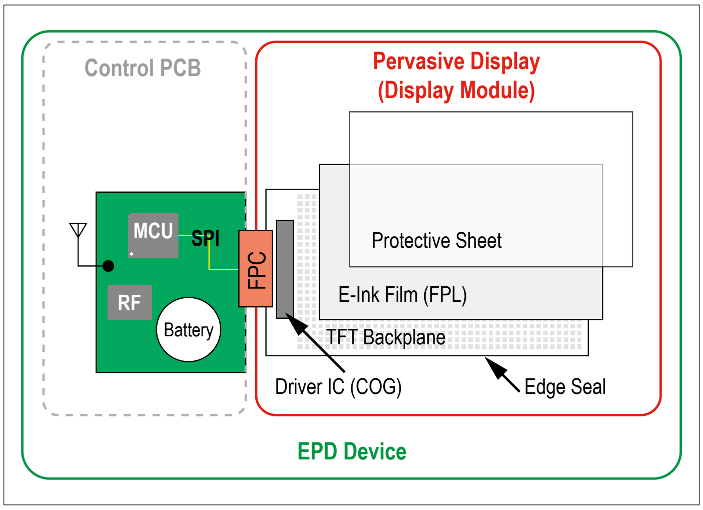


Figure 6. Structure of the E1271CS021 e-paper display (www.densitron.com).

As shown in **Figure 6**, the display consists of an e-ink film on top of a TFT backplane and a protective thin cover. The timing and data transfer to and from the display is controlled with an MCU mounted on a printed circuit board at its back. The display is interfaced to the external world using Wi-Fi wireless communication. Configuration of the display is over the Wi-Fi link using the supplied SmartConfig tool.

The Future

It is estimated that the market for e-paper displays is growing constantly and will reach over $9 billion by the year 2025. It is also expected that the future developments and the reduction in the cost of the e-paper technology will lead to true paperless offices. The fact that e-papers have very low weights and can be flexible are great advantages in the display technology. This for example means that a display can be folded and carried along easily, or a wearable display can be constructed and for example worn as a watch (e.g. the Seiko e-ink watch). Research is ongoing to integrate touch-screen technology with e-paper displays. It is the author's opinion that considering the great advantages and features offered, the e-paper is here to stay and with the recent development of the coloured e-paper, most of the future displays will probably be based on the e-paper technology.

(160196)

Development kits

Many companies are now offering e-papers at competitive prices:

- Waveshare offers a 4.3", 800x600 resolution Arduino compatible e-paper at a cost of $55, operating with a serial data port.
- Atmel introduced an e-paper development kit called the EPD Xplained Pro at a cost of $72 which operates with the SPI interface and supports screen sizes up to 2.7 inch.
- An e-paper kit is available from Embedded Artists at a cost of €29, designed to support 2.7 inch e-paper displays.
- Adafruit sells a trio of e-paper development kits names rePaper Project Board with prices around $35 including a screen. The kits operate with 5 V and a 20-pin cable is provided so you can plug into Arduino header.
- SEEED offers an e-paper display shield designed to plug into an Arduino board. The shield costs $50 and has a 2.1-inch screen.
- E-ink sells an e-paper development kit with two displays at a cost of $70. The kit plugs into the USB port of a computer and includes all the necessary hardware and drivers.
- Renesas Electronics sells an e-paper development kit through Conrad. The kit includes a small screen and includes all the necessary hardware. It sells for €79 and is interfaced to the host computer through the USB port.

Vacuum Fluorescent Displays

Peculiar Parts, the series

By **Neil Gruending** (Canada)

I've written about quite a few different types of vacuum tubes in previous installments so let's continue that theme and look at vacuum florescent display (VFD) tubes. We've all seen them and probably haven't given them a second thought as they work day after day in all sorts of conditions.

The first seven segment VFD was designed in the 60's to be a low power and long life replacement for nixie tubes for numeric displays. LED and LCD technology has since replaced VFDs in many applications but they are still widely used today in test equipment, home electronics and other applications where a long life and wide viewing angles are important. VFDs also excel in automotive applications because they are very rugged and can operate over wide temperature ranges.

The dead giveaway for vacuum florescent displays is their characteristic blue-green color like in **Figure 1** although they can be other colors as well. A VFD resembles a triode tube and has a cathode filament, one or more control grids and phosphor coated illumination anodes like in **Figure 2** (taken from [1]). The filament generates electrons which are either accelerated by a positive grid voltage towards the anodes or are blocked with a negative grid voltage. A positive charge on the anodes will attract the electrons which then illuminate the anode phosphor coating.

This triode construction of a VFD means that they generally need more complex drive circuitry compared to other display technologies. One challenge is to equalize the filament voltage across the display so that the display will have a uniform brightness. It's possible to drive it a DC voltage but voltage drop across the filament will also reduce the display brightness in that portion of the display. The recommended solution for this problem is to drive the filament with an AC voltage from a center tapped transformer. Grounding the center tap halves the intensity variation in the display which is necessary for larger displays.

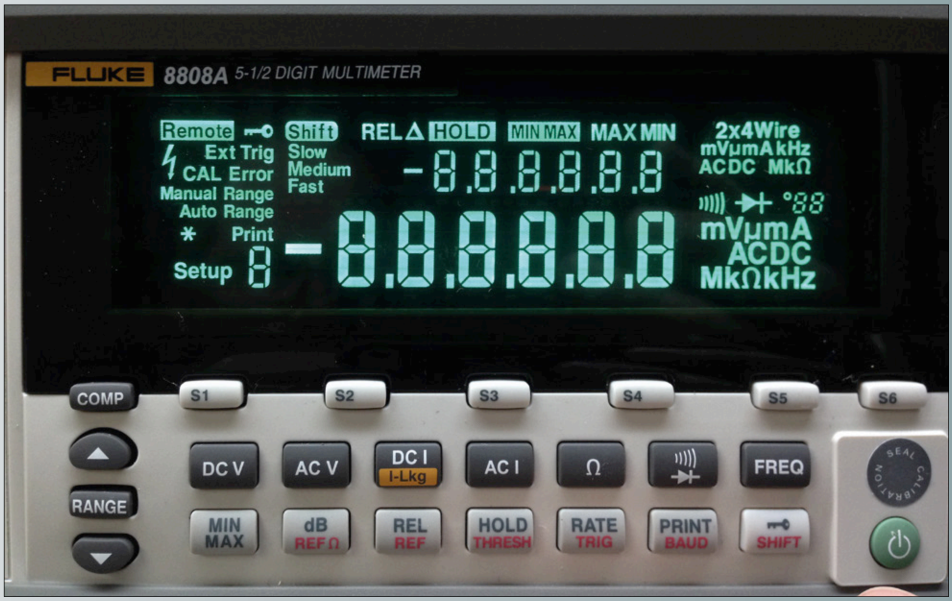


Figure 1. Fluke 8808A VFD.

The VFD grid and anode connections are driven with DC voltages although they usually need to be quite a bit higher than CMOS logic levels which can be a challenge. Some small VFDs use only one grid connection and can wire out all of the anodes. They are called static displays because the whole display is active all of the time which simplifies the control circuitry. It also reduces the anode and grid voltages required to properly illuminate the display because the grid is always on. But most VFDs are more complex devices that require a multiplex or dynamic drive circuit. These VFDs have anodes that are shared amongst several grids and the grids are used to select the group of anodes that are currently active which typically needs a more elaborate control circuit. The grid and anode voltage also usually need to be higher than those used for a static display to achieve the same intensity since each grid is only on for a portion of the time.

Fortunately it's possible to buy CMOS friendly VFD modules and there are even some dedicated VFD driver ICs available. So don't be scared to try them out in your next project! ⏮

(150484)

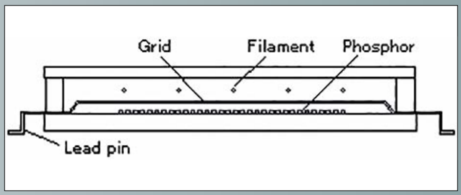
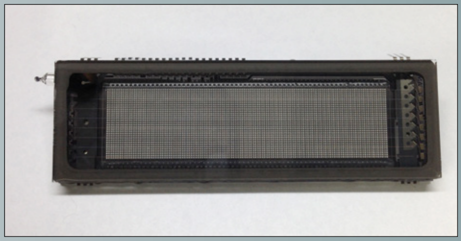


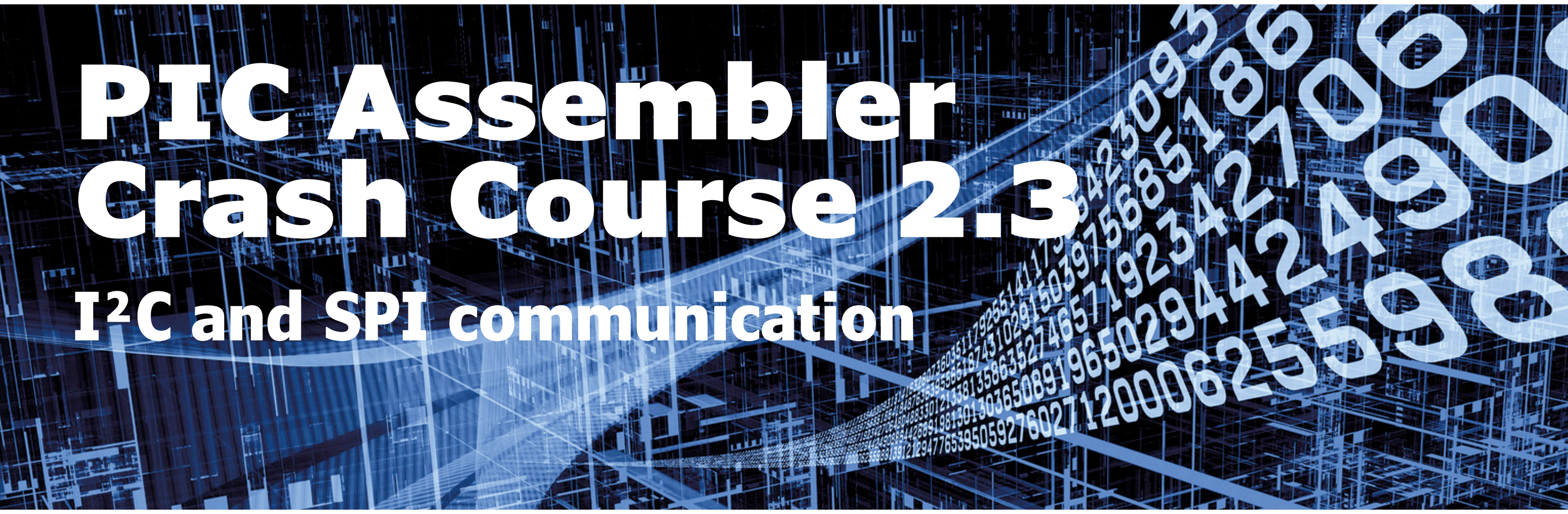
Figure 2. VFD basic construction. Source: [1]



Web Link

[1] www.noritake-elec.com/display/vfd_operation.html

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



PIC Assembler Crash Course 2.3

I²C and SPI communication

In the first two installments of this second crash course in PIC assembler we looked at how to drive a multiplexed LED display and at indirect addressing. In this third and final installment we turn to communications using the I²C and SPI interfaces.

By **Miroslav Cina** (Germany) miroslav.cina@t-online.de

Microcontrollers since the year dot have had built-in serial ports for communicating with other devices. In the past these were almost invariably UARTs (universal asynchronous receiver/transmitters), but these days interfaces for buses like I²C (inter-integrated circuit) and SPI (serial peripheral interface) are often also provided. Both these systems are very widely used, do not require too much effort on either the software or the hardware side, and are compatible with the voltage levels typically used by microcontrollers. There is a large range of peripheral chips available using one or the other of these interfaces, including many different types of sensor. Here we will look at how these interfaces are used from assembler code.

The MSSP

Every member of the family of PIC microcontrollers we are using, even the very smallest, includes at least one MSSP (master synchronous serial port) unit; many have two. The function of the MSSP should be obvious from its name. An MSSP comes in handy in the following situations:

- when many I/O ports are required, more than the microcontroller itself offers;
- when special-purpose peripheral chips such as ADCs, DACs, RTCs and sensors are to be driven;
- when a simple approach is needed to communicating with a large number of peripheral chips;
- when communication must be carried out over relatively long distances (say a few meters).

The MSSP can be operated in a range of different modes. The most important of these are its configuration as an I²C or an SPI port. In both of these configurations the PIC can be arranged to act as a slave or as a master.

Many microcontrollers offer an SSP module, but the additional letter 'M', for 'master', is particularly important: this feature saves a lot of time and program memory when the serial interface is used in master mode.

I²C

The I²C bus and protocol was developed in 1982 for use in 'smart' televisions by Philips Semiconductors (today NXP). It was mainly aimed at providing simple communications between devices within a single appliance. In practice one microcontroller would use the serial bus to talk to other ICs or other microcontrollers: typically one master and many slaves. Because of its simplicity, Atmel adopted the protocol, renaming it TWI (two-wire interface) to reflect the fact that only two signal lines (plus ground) are used. Today over 50 manufacturers produce devices supporting this bus.

I²C offers synchronous bidirectional communication, although only in one direction at a time ('half duplex'). In order to communicate with an individual chip, each slave on the bus has an address which it listens for: when this address is matched the slave acknowledges the message from the master. The data line is called SDA and the clock line is called SCL: obviously only one message can be in transit on the bus at any one time. The basic data rate is 100 Kbit/s, and high speed modes run up to 5 Mbit/s. Rates of 100 Kbit/s, 400 Kbit/s and 1000 Kbit/s are commonly supported.

SPI

The SPI bus and protocol was designed by Motorola alongside its introduction of the illustrious 68000 CPU series. SPI is again a synchronous serial interface, but in this case it is 'full duplex'. This makes the hardware of the bus slightly more complicated: along with the SCLK clock signal, bidirectional communication requires both a MOSI (master out, slave in) or SDO (serial data out) data line and a MISO (master in slave out) or SDI (serial data in) line. Furthermore, the protocol requires at least one SS (slave select) signal and, of course, ground. More precisely, each slave requires its own SS signal (also called 'CS' for 'chip select'), and these signals can conveniently be numbered consecutively.

Table 1.

MSSP #	Signal	Port	Pin #
1	SDA1	RB4	13
	SCL1	RB6	11
2	SDA2	RB5	12
	SCL2	RB7	10

Table 2.

MSSP #	Signal	Port	Pin #
1	SDO1	RC7	9
	SDI1	RB4	13
	SCK1	RB6	11
2	SDO2	RA5/RC1	2/15
	SDI2	RB5	12
	SCK2	RB7	10

The extra complexity and the use of push-pull output drivers (in contrast to the open-collector architecture of the I^{2}C bus) allows for much higher data rates (up to 80 Mbit/s) and lower operating power. A disadvantage is the relatively limited length of the signal lines: a similar problem affects UART-type interfaces. Because of the high data rates possible, SPI is ideal for connecting to chips that have a high throughput such as fast ADCs or external flash memories.

Some microcontrollers can support the two protocols simultaneously for extra flexibility: for example MSSP1 can act as an I^{2}C bus master while MSSP2 acts as an SPI bus master.

Design change

When a microcontroller is used in MSSP master mode there are certain pins that take on dedicated communications functions. The allocated pins for I^{2}C and SPI communication on the PIC16F1829 microcontroller are shown in **Table 1** and **Table 2**. For SPI the chip select pins can be freely chosen, as they are controlled by the firmware rather than by the MSSP hardware. So, for example, we can see from Table 2 that when MSSP2 is configured for SPI the SDO signal can be on port pin RA5 or on RC1.

We should note at this point an infelicity in the design we presented in the first installment of this series, where the port pins RA5, RC1 and RC7 were used in driving the display. If we want to use this design in conjunction with the I^{2}C interface there is no problem, as this can be configured to use only port pins RB4 and RB7. But, if we want to experiment with the SPI port, we will need to free up at least one of the three port pins RA5, RA1 or RC7.

However, we can create an opportunity from this setback: we can see how easy it is to modify the software to cope with this kind of hardware change. So, if we wish to experiment with MSSP2, we can free up pin RA5 by moving the connection to the cathode of LD3 from RA5 to RA2. After making the change to the hardware, a small modification is required to the firmware as follows.

1. In the routine init_main we need to configure RA2 as an output, by setting bit 2 of the TRISA register to 1 instead of 0. So:

```
movlw B'11001100'
movwf TRISA
```

becomes

```
movlw B'11001000'
movwf TRISA
```

2. Where previously we referred to `PORTA,D'005'`, we must instead refer to `PORTA,D'002'`. That occurs at the following points in the code.

(a) At the end of the routine init_main where all the cathode signals are deactivated.

(b) In the subroutine ishow_digit3 where the last cathode signal is activated.

(c) In the subroutine disp_off where all the cathode signals are deactivated.

The change at each of these points is essentially the same. Instead of

```
bsf PORTA,D'005'
```

we need to write

```
bsf PORTA,D'002'
```

and likewise instead of

```
bcf PORTA,D'005'
```

we need to write

```
bcf PORTA,D'002'
```

I^{2}C communication

Now that we have seen the big picture of how the MSSP can be used, we can start to look at I^{2}C communication in more detail.

PIC16F1829 as I^{2}C bus master

As we mentioned above there is a wide choice of interesting chips that can be controlled by a microcontroller operating as a bus master. But first, before using the MSSP functions, we need a few lines of code to set up values in three registers.

The first of these registers is **SSPxCON1**. Here ‘x’ can be ‘1’ or ‘2’, depending on whether MSSP1 or MSSP2 is being configured. The ‘larger’ (from an MSSP point of view) microcontrollers have both an SSP1CON1 and an SSP2CON1 register,

Table 3.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/C/HS	R/W	R/W	R/W	R/W	R/W	R/W	R/W
WCOL	SSPOV	SSPEN	CK	SSPM<3:0>			

Listing 1.

```
;---------------------------------------------------
;set up MSSP2 for I2C BUS communication (PIC is the
master)
;---------------------------------------------------
;
;step 1
i2c_init nop
   banksel  TRISB
   bsf   TRISB,D'007'   ;SCL
   bsf   TRISB,D'005'   ;SDA
   movlw H'04'
   movwf BSR   ;select BANK 4
;
;step 2
   movlw B'00101000' ;see Datasheet Page 277
         ;bit 3-0 SSPM<3:0> = 1000
         ;1000 = I2C Master, clock = FOSC/
(4*(SSPADD+1))
   movwf SSP2CON1
;
;step 3, set I2C speed
   movlw B'00111111'
   movwf SSP2ADD
   clrf  SSP2STAT
   clrf  BSR   ;select BANK 0
   return
```

while the smaller ones just have SSP1CON1. The layout of the register is shown in **Table 3**. Initially we are interested in just the four least significant bits SSPM<3:0> and the SSPEN bit. SSPM<3:0> is responsible for the main configuration of the MSSP module. We will not go into too much detail here, but suffice it to say that setting these bits to 1000b enables I^{2}C master functions.

The data rate can be computed from the main clock frequency using the formula

```
      I2C Clock Speed = Fosc / (4 * (SSPxADD+1))
```

where **SSPxADD** is our second register of interest. In I^{2}C master mode this register stores the clock division ratio, but note that in I^{2}C slave mode it contains the I^{2}C bus address of the slave. The SSPEN bit enables (when set to 1) or disables (when set to 0) the MSSP unit.

The final register of interest is **SSPxSTAT**. For our experiments it suffices to ensure that all its bits are set to 0.

A complete I^{2}C initialization routine written in assembler therefore looks like the code in **Listing 1**. First we configure the two port pins as inputs (step 1), which is a prerequisite for using them in I^{2}C mode. Next, in step 2, we set up SSP2CON1. The mode control bits are set to 1000b to configure the unit in I^{2}C master mode. Simultaneously we enable the MSSP unit by setting bit 5 (SSPEN) to 1. Finally we configure the data transfer rate and initialize the status register (step 3). In our code example we set the data rate to 62.5 kHz (F_{OSC} = 16 MHz, SSP2ADD = 63). The microcontroller is now ready for use as an I^{2}C bus master.

I^{2}C communication routines

We will assume that you are familiar with the basics of the I^{2}C protocol before proceeding: if not, a little Googling will set you on the right track.

Every communication activity takes time. It is possible to monitor the SSP2STAT register to determine exactly when the current activity has finished, but for our simple experiments and in many real applications it is sufficient to add a brief delay after each step and then just assume that everything is ready. This makes the firmware very straightforward. The approach described below will be perfectly adequate as long as speed is not critical.

Each I^{2}C message begins with what is called a 'start condition', which is initiated by the master. When using the MSSP the start condition is easy to generate in assembler as follows.

```
i2c_start  bsf   BSR,D'002'  ;select register BANK 4
   bsf   SSP2CON2, SEN
   bcf   BSR,D'002'  ;select register BANK 0
   call  d55
   return
```

As can be seen, all we have to do it set the SEN bit of register SSP2CON2 to 1. The subroutine d55, which is very simple, generates a delay after this communication step.

```
d55   movlw D'255'
   movwf TIMER2
d55_loop decfsz   TIMER2,F
   goto  d55_loop
   return
```

The subroutine executes its empty loop 255 times. With a clock frequency of 16 MHz this gives enough delay for any step of the I^{2}C communications protocol at our chosen data rate.

Each I^{2}C message finishes with a 'stop condition'. Like the start condition above, implementing this in assembler is child's play:

```
i2c_stop bsf   BSR,D'002'
   bsf   SSP2CON2, PEN
   bcf   BSR,D'002'
   call  d55
   return
```

As can be seen, the hard work is done by the MSSP hardware, and all we have to do in firmware is set the PEN bit of the SSP2CON2 register to 1.

For the transfer itself we will need to know how to receive and transmit data. The following routine illustrates how it is possible for a master to transmit one byte to a slave device.

```
;transmit the byte value from register W
i2c_send bsf   BSR,D'002'
   movwf SSP2BUF
   bcf   BSR,D'002'
   call  d55   ;delay
   return
```

As can be seen, sending a byte is very straightforward: all we have to do is write the byte to be transmitted into the register SSP2BUF. The MSSP hardware then once more takes care of the actual work, and the byte passed to the routine in the W register is sent out on the I²C bus.

The receive routine, which reads one byte from a slave, is a little more complicated.

```
;store received byte into register W
;step 1
i2c_receive bsf   BSR,D'002'
   bsf   SSP2CON2,RCEN
   bcf   BSR,D'002'
   call  d55
;step 2
   bsf   BSR,D'002'
   movf  SSP2BUF,0
   bcf   BSR,D'002'
   return
```

First, in step 1, we set the RCEN bit in the SSP2CON2 register to 1. This tells the MSSP hardware that a byte is to be received. We then wait for a short while, after which the received byte should appear in the SSP2BUF register. Finally, in step 2, the received data byte is read from this register and returned in the W register.

Although the next three code snippets are strictly speaking not required for the examples we will be looking at, they nevertheless represent important elements of I²C communication and so are worth exploring.
After the master has received a byte from a slave, it will normally send an acknowledge (ACK) or negative acknowledge (NACK) in return, to tell the slave whether receipt was successful. The routine to generate the ACK signal looks like the following.

```
i2c_ack  bsf   BSR,D'002'
   bcf   SSP2CON2, ACKDT   ;ACK
   bsf   SSP2CON2, ACKEN
   bcf   BSR,D'002'
   call  d55
   return
```

When the ACKEN bit is set to 1 the MSSP hardware will initiate an ACK sequence, which will place the bit stored in ACKDT on the bus. If this latter bit is a zero, an ACK is sent; if the bit is a one, a NACK is sent. The negative acknowledge subroutine therefore looks almost identical: the exception is just that the ACKDT bit of the SSP2CON2 register is set to 1.

```
i2c_not_ack bsf   BSR,D'002'
   bsf   SSP2CON2, ACKDT   ;NACK
   bsf   SSP2CON2, ACKEN
```

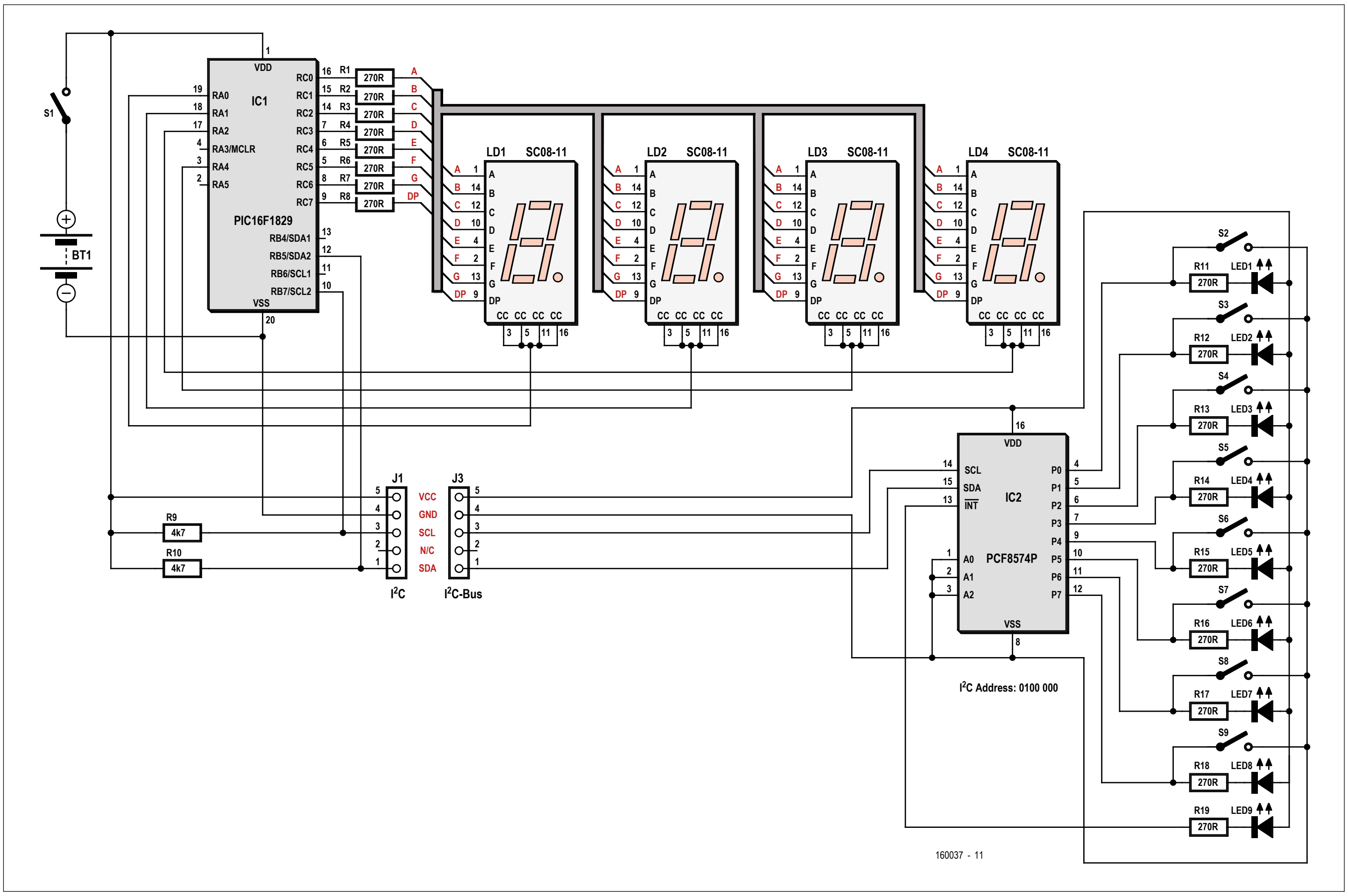


Figure 1. The LED display unit extended using a port expander and eight discrete LEDs.

```
    bcf   BSR,D'002'
    call  d55
    return
```

We complete the set of routines with code to perform an I²C restart. An I²C restart is required when the direction of communication with a slave needs to be changed. The following routine implements this function.

```
i2c_restart bsf   BSR,D'002'
    bsf   SSP2CON2, RSEN
    bcf   BSR,D'002'
    call  d55
    return
```

As you may already have guessed, all we need to do is set a bit in the SSP2CON2 register, here the RSEN bit, to 1.
One further remark: all these subroutines assume that when they are called the bank select register BSR is pointing to bank 0 (BSR = 00h). If this is not the case, then the instruction `clrf BSR` must be inserted before each `bsf BSR,D'002'` instruction.

Port expansion using a PCF8574

Now that we have the firmware components required to implement I²C communication in place, we can look at how to use them in practice. For our first experiments it is convenient to use a peripheral chip that lets us see directly what is going on by driving some LEDs. The PCF8574 from Texas Instruments or from NXP fits this bill: it is an eight-bit I/O port expander with an I²C interface, and it is available in a 16-pin DIL package.
The eight I/O pins of the device can be configured as inputs or as outputs. If any significant load is to be connected to an output, it should be connected between V_{DD} and the output pin; also, the maximum load is 25 mA per output, and there is an overall current limit for the whole device of 100 mA.
First we will see how to drive the eight LEDs, and then we will also read back the input logic level on the eight I/O pins. **Figure 1** shows the circuit for this first experiment with the I²C bus, and **Figure 2** shows how the display circuit from the first installment of this series, assembled on a piece of prototyping board, might look when extended to include I²C and SPI connections. The port expansion circuit built around IC2, including its eight LEDs, appears towards the bottom right of Figure 1. **Figure 3** shows how this part of the circuit might look, built on its own board as a self-contained module.
The output drivers of the PCF8574 are in an open-drain configuration. This allows an output to drive an LED and at the same time have a pushbutton connected to ground, without the two functions interfering with one another. When a button is pressed, the corresponding LED will light independent of the state of the output of the device. For the prototype in Figure 3 an eight-way DIP switch was used instead of a row of pushbuttons in the interests of simplicity.
The circuit of Figure 1 also differs from the original in the addition of the two pull-up resistors R9 and R10. These are required because the I²C bus uses open-collector or open-drain drivers rather than push-pull drivers; without the resistors communication would not be possible.

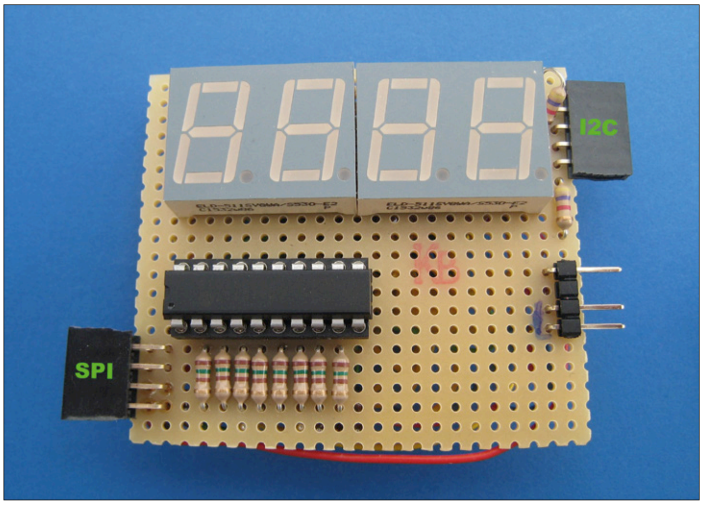


Figure 2. The LED display unit expanded to include I²C and SPI connections.

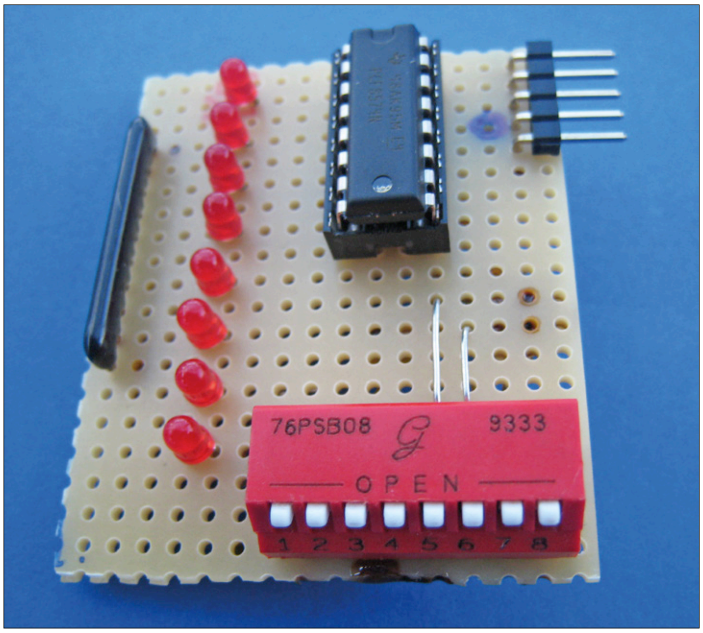


Figure 3. The port expander with eight LEDs assembled on prototyping board as a standalone module.

Experiment 1: I²C write

Our first experiment involves writing data to the PCF8574 port expansion chip. To do this we simply extend the final demonstration application from the second installment of this series. As well as showing data read from the flash memory on the display in hexadecimal form as before, we will also display them in binary using the PCF8574 and its eight LEDs. The way that the eight LEDs are connected, they will light when the device outputs a binary zero and will be dark when the device outputs a binary one; if you wish you can compensate for this in the code.

The only change to the main loop in the demonstration application is that before sending the data to the LED display we call the new subroutine PCF8574_send. As its name indicates, this sends one byte of data over the I²C bus to the PCF8574.

```
;step 1
PCF8574_send   call  i2c_start
   movlw B'01000000'
   call  i2c_send
;step 2
   movf  v_value,0
   call  i2c_send
;step 3
   call  i2c_stop
   return
```

Thanks to the components we have already written, this subroutine is very simple. First, in step 1, we generate the start condition; then we set up the address of the PCF8574 slave, which in this case is 0100000b. The address itself is only seven bits long: the final zero signifies that we are performing a write operation. Then, in step 2, we transmit the data byte to the PCF8574. The LEDs should light in the correct pattern immediately this byte is received. Finally, in step 3, we finish the communication by generating the stop condition.

Experiment 2: I²C read

In this example we show how to request and receive data from the PCF8574 slave. To do this we modify the main loop so that instead of showing data bytes from the flash memory on the LED display, we show the byte received from the PCF8574.

```
;step 1
PCF8574_read   call  i2c_start
   movlw B'01000001'
   call  i2c_send
;step 2
   call  i2c_receive
   movwf v_value
;step 3
   call  i2c_not_ack
   call  i2c_stop
   return
```

At the beginning of the routine, in step 1, we generate the start condition. We then set up the address of the PCF8574 as before, but this time in read mode: the last bit of the address is therefore one. This tells the slave that it is to send a byte,

Listing 2.

```
;----------------------------------------------------
;set up MSSP2 for SPI BUS communication (PIC is the
master)
;----------------------------------------------------

;SPI initilization on MSSP2
   ;RB7 = SCK2
   ;RB5 = SDI2
   ;RA5 = SDO2
;
;step 1
spi_init nop
   banksel TRISB
   bsf   TRISB,D'005'   ;SDI2 (RB5 -- input)
   bcf   TRISA,D'005'   ;SDO2 (RA5 -- output)
   bcf   TRISB,D'007'   ;SCK2 (RB7 -- output)
   bcf   TRISB,D'006'   ;CS   (RB6 -- output)
;
;step 2
   banksel  APFCON1
   bsfAPFCON1,SDO2SEL ;SDO2 is on RA5 (not on RC1)
   banksel  SSP2CON1
   movlw B'00000010' ;see Datasheet Page 260
      ;bit 7    = 0    -> N/A
      ;bit 6    = 0    -> N/A
      ;bit 5    = 0    -> disable MSSP2
      ;bit 4    = 0    -> Idle state for clock
                          is a low level
      ;bit 3-0 = 0010 -> SPI Master mode,
                          clock = FOSC/64
   movwf SSP2CON1
   clrf  SSP2STAT
;
;step 3
;bit 7    = 0 -> Input data sampled at middle
                 of data output time
;bit 6    = 0 -> Transmit on transition from
                 idle to active clock state
   bsf   SSP2CON1,D'005'   ;enable SPI on MSSP2
   clrf  BSR
   bcf   PORTB,D'006'   ;CS = 0 --> inactive
   return
```

Listing 3.

```
;-------------------------
;send and receive SPI data
;-------------------------
;
;step 1
spi_send nop
   banksel  SSP2CON1
   bcf   SSP2CON1,SSPOV ;for safety...
   bcf   SSP2STAT,BF ;for safety...
;
;step 2
   movwf SSP2BUF
spil2 btfss SSP2STAT,BF
   goto  spil2
;
;step 3
   movf  SSP2BUF,0
   clrf  BSR
   movwf v_spi_buffi
   return
```

in this case the data read from the I/O port pins. Then, in step 2, the subroutine i2c_receive is called to read in the byte and place it in the W register. The received byte is saved in the variable v_value for later use.

Finally, in step 3, we finish the communication. We send a NACK to the slave, which indicates the end of the read operation, and then generate the stop condition.

SPI communication

In principle communicating over SPI is rather easier than over I²C. At any rate, configuring the PIC as an SPI master is very similar to the I²C case: as you can see from **Listing 2**, the same MSSP registers are used.

SPI uses two data connections, one clock connection, and at least one chip select connection. This is rather more than is required for I²C, and so the initialization procedure (step 1) is somewhat more involved than before. Here we use RB5 as SDI (serial data input), and so we have to configure this pin as an input. The other connections (SDO, CLK and CS) are all outputs.

We now come to the settings needed in the SSP2CON1 register (step 2). Bits 0 to 3 configure the MSSP2 unit as an SPI master. SDO can appear on either RA5 or RC1, and so next we specify which: setting the SDO2SEL bit of the APFCON1 register configures RA5 for use as SDO.

We also need to configure the 'SPI mode' to be compatible with that used by the connected slave. The various SPI modes differ in the following ways, any combination of which is possible.

- Idle clock level: bit 4 of SSP2CON1 specifies whether the level on the SCK line should be logic 0 or logic 1 when inactive.
- SPI clock edge select: bit 6 of the SSP2STAT register specifies the point in time relative to the SCK signal when the level on the SDO line changes.

Normally the inactive state of the CS signal is at logic 1, and a low on CS enables the peripheral device. However, in the case of the DS1306 RTC (real-time clock) we will be using here, it is the other way around.

We will configure the SPI mode so that SCK is low when inactive, and so that the data line changes state on the rising edge (from 0 to 1) of the clock. If you are using the code with a different SPI slave device, you should check its requirements in its datasheet.

To complete the initialization (step 3) we enable the MSSP2 module and set the chip select low, which leaves the interface initially in a disabled state.

We will need a single firmware component to implement communication over the SPI bus. The relevant code can be found in **Listing 3**. The same code is used for transmitting and receiving data, as the SPI bus allows communications in both directions at once, although this feature is not frequently used. If the master wishes only to receive data, it must send a dummy byte to the slave in order to clock its data out.

The subroutine spi_send expects the byte to be transmitted to be supplied in the W register. At the beginning (step 1) the overflow flags are cleared: this is just a precaution in case something had gone wrong with the previous data transfer. Then, in step 2, the byte to be sent is written to the SSP2BUF register. The MSSP hardware now springs into life and sends out the contents of this register over the bus on the SDO pin. The firmware now waits in a loop until the transmission has completed: this is detected using the BF bit in the SSP2STAT register, which is set to 1 at the end of a communication operation.

Because the interface is bidirectional, the SDI pin is sampled at the same time. The information received is written to the SSP2BUF register at the end of the operation.

Finally, in step 3, the SSP2BUF register is read and a copy of its contents is stored in memory at v_spi_buffi for later use.

Connecting a DS1306 RTC

Connecting to a DS1306 real-time clock device over SPI is not particularly complicated. **Figure 4** shows the circuit. **Figure 5** shows how the additional circuitry at the bottom right might look if built as a separate module on a piece of prototyping board.

For the SPI connection from the main board to the RTC module to function correctly it is essential to make sure that the SDI and SDO signals are crossed over: what is an output on the master is an input on the slave, and vice versa, and the same goes for any other SPI peripheral device. We use RB6 for the chip select signal. A real-time clock requires a watch crystal, typically running at 32.768 kHz. The IC we have chosen requires a crystal specified for a load capacitance of 6 pF.

Experiment 3: Clock with LED display

In order to convert the LED display into a clock we need to write some special-purpose code, but it will turn out rather simpler than might be expected. The main requirement is a subroutine to read the time information from the DS1306, and to implement this we need to look deeper into the characteristics of the device.

The clock IC has 128 addressable locations. The first seven of these are reserved for the clock functions, with address 00h holding the seconds count, address 01h the minutes count, and 02h the hours count. These three will be sufficient for our purposes. The values themselves are stored in BCD format. For example, if the time is 14:32:56 (in 24-hour clock) the following values will be read from the device.

- Address 00h (seconds) = 56h
- Address 01h (minutes) = 32h
- Address 02h (hours) = 14h

For each communication operation the DS1306 first expects to receive the address, and then outputs the required data. As long as the CS signal is enabled, the address counter will be automatically incremented after each read. It is therefore very simple to read several bytes consecutively from the device.

If the address is increased by 80h then instead of reading data from the device, data can be written to it. So, if address 81h is used instead of 01h, then the DS1306 will expect to receive a value to be written to the minutes counter.

Listing 4 shows a routine that reads the hours, minutes and seconds from the DS1306. First, in step 1, the CS signal is activated (that is, set to 1). From this point the slave starts listening to the bus, and expects to receive the address of the register to be read. The subroutine spi_send is used to transmit the value 00h on the bus (step 2): this tells the DS1306 that it is the value of the seconds counter that is wanted.

In order to read the seconds value we call the spi_send routine again (step 3). The byte transmitted here does not matter, and so we do not need to pass in any particular value to the routine. The slave ignores the received byte and sends back the value at address 00h over the other data line. After the call the seconds counter value is in the W register, and we store a copy in memory at v_sec. We then call spi_send again. Because of the automatic increment function of the DS1306, the value we receive is the contents of the minutes counter. A final call to spi_send retrieves the contents of the hours counter.

The other registers in the DS1306 are not of interest in this experiment. We therefore finish the communication operation by deactivating the CS signal (step 4).

Now we have the current time stored in the variables v_hrs, v_min and v_sec for later use. Reading these from the DS1306 took just four calls to spi_send: one to send a byte to the slave and three to read bytes back. All that is left to do is to display the results as a time.

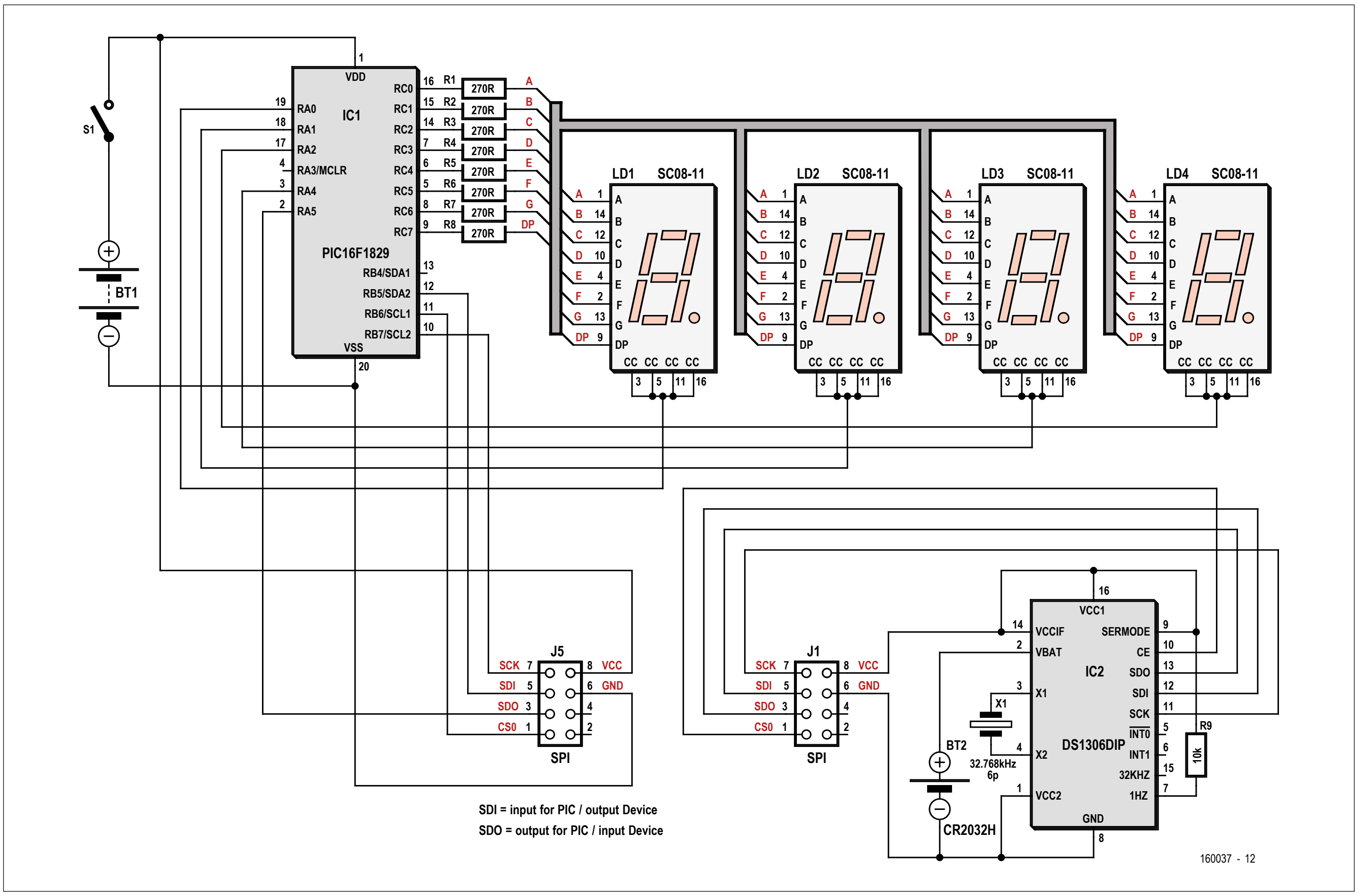
The time data bytes are in BCD format, and we already have a routine to display hexadecimal values. Displaying the time is thus very easy. **Listing 5** shows the main part of the code, the routine Main_Appl. In step 1 this code shows '-- --' on the display and then, in step 2, initializes the SPI port.

The main loop at clock_loop (step 3 onwards) starts with a brief delay and then uses the subroutine spi_read_time to read the time information from the DS1306. Then, as described in the previous installment of this series, each BCD byte is divided into two nibbles and, in step 4, the result is copied to the 'video buffer variables' v_pos0 to v_pos3. And that is all there is to it.

The seconds value is not used, as the display has only four digits. If you wish, you could make the clock display minutes and seconds by using the variable v_min in step 4a and the variable v_sec in step 4b.

A further small extension to the display routine is possible to make the middle decimal point flash. For reasons of space we shall not describe the details here, but interested readers will find the grisly details in the source code available for download from [1].

We have also not included code to allow you to set the time, but given the knowledge acquired during this course writing it should not preset too great a problem. Note that when new data values are written to the DS1306, the old values are lost. But even without the code to set the time it is possible to use the unit as an accurate clock: simply wait patiently until exactly midnight before powering up



SDI = input for PIC / output Device
SDO = output for PIC / input Device

Figure 4. The circuit for the LED display with real-time clock device attached using the SPI port.

the circuit. No further configuration is required. By default the device uses the 24-hour clock.

Further experiments

As we mentioned above, having an I²C bus or an SPI bus makes it easy to interface to a large number of peripheral chips: a quick Internet search for ‘I2C IC’ or ‘SPI IC’ will show that almost every conceivable function is covered. For example, it would be very easy to combine the LED display unit with an INA219 to make a voltage and current meter. All that is needed in addition to the chip (and of course a shunt for current measurements) is a small amount of code. It is easy to make the microcontroller control the INA219, to select its mode of operation and to retrieve and display the measured values.
It is equally easy to repurpose the LED display unit as a thermometer. For this a temperature sensor chip such as the Microchip TCN75, the Texas Instruments TMP100 or the Analog Devices ADT7410 can be used. These devices are economical, easy to use, require practically nothing in the way of external components, and deliver temperature values digitally over an I²C bus.

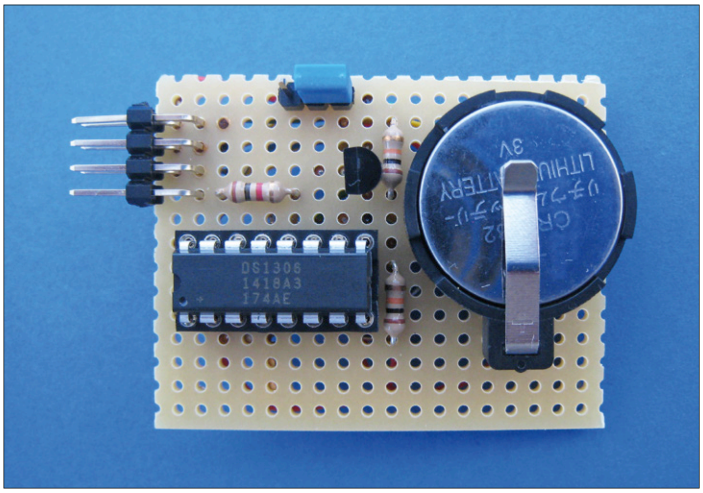


Figure 5. The SPI real-time clock assembled on prototyping board as a standalone module.

Conclusion

This marks the end of our second assembler crash course. We hope that it has been helpful, instructive and inspiring. If you have any questions on the topics covered, please do not hesitate to contact the author through his email address: miroslav.cina@t-online.de.

(160037)

Web Link

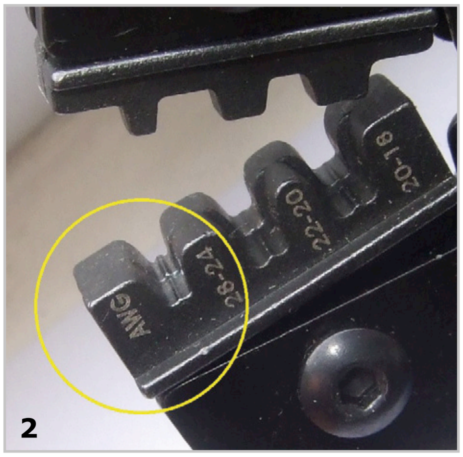
[1] www.elektormagazine.com/160037

Listing 4.

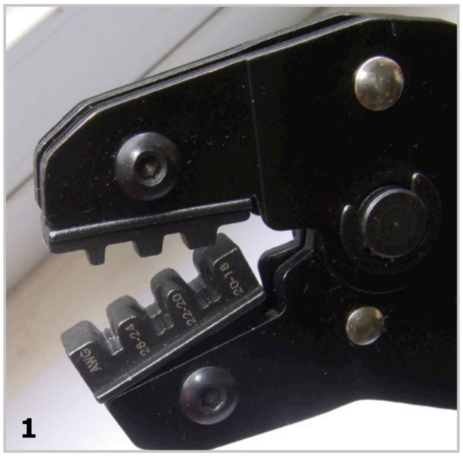
```
;------------------------------
;read time from DS1306 via SPI
;------------------------------
;
;step 1
spi_read_time  bsf   PORTB,D'006'   ;activate CS
;
;step 2
   movlw H'00'
   call  spi_send ;send address
;
;step 3
   call  spi_send ;read seconds
   movf  v_spi_buffi,0
   movwf v_sec
   call  spi_send ;read minutes
   movf  v_spi_buffi,0
   movwf v_min
   call  spi_send ;read hours
   movf  v_spi_buffi,0
   movwf v_hrs
;
;step 4
   bcf   PORTB,D'006'   ;deactivate CS
   return
```

Listing 5.

```
;-------------------------------
;Main Application - DS1306 Clock
;-------------------------------
;
;step 1
Main_Appl   movlw H'10'
   movwf v_pos0
   movwf v_pos1
   movwf v_pos2
   movwf v_pos3
;
;step 2
   call  spi_init
;
;Step 3
clock_loop  call  dr2
   call  spi_read_time
;
;step 4a
   movf  v_hrs,0
   movwf v_value
   call  calc_hex_disp
   movf  v_msb,0
   movwf v_pos0
   movf  v_lsb,0
   movwf v_pos1
;
;step 4b
   movf  v_min,0
   movwf v_value
   call  calc_hex_disp
   movf  v_msb,0
   movwf v_pos2
   movf  v_lsb,0
   movwf v_pos3
   goto  clock_loop
```

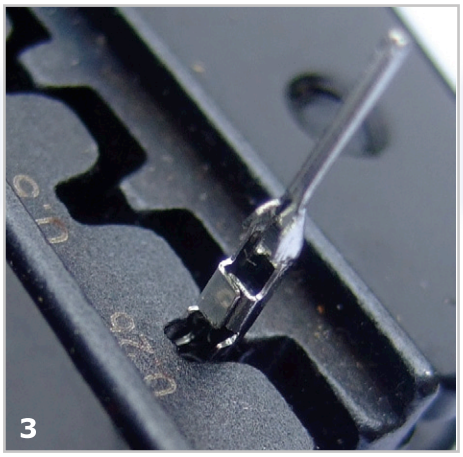


2

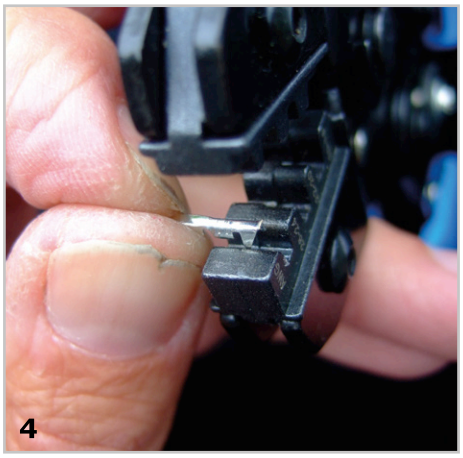


1

Tips and Tricks



3



4

Crimping pinheaders

Contributed by Stephen Bernhoeft

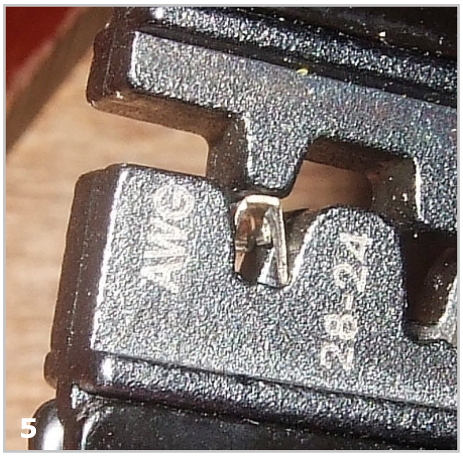
*Crimping those so-called 'pinheaders' can be an extremely frustrating experience to the uninitiated. The first hurdle is to identify a suitable tool. One that I've used successfully is the SN-28B (**Figure 1**). Notice the fine detail on the jaws "AWG28-24" — the two sizes of "heart shaped" die in particular (**2**).*

*The wire side of the crimp goes into the "AWG" (wider) end. This is tricky, because you must only just place the wire side into this part. You have to ensure that the working side of the crimp insert (the part you use as a 'plug' or 'socket' when the assembly is finished) does not get squashed. Pictured in **3** is the working side of a male pinheader. The section showing must not be in, or too close to the crimp jaws, otherwise the pin will be damaged. The only way you can ensure that is to pay very close attention to the crimp placement (**4**).*

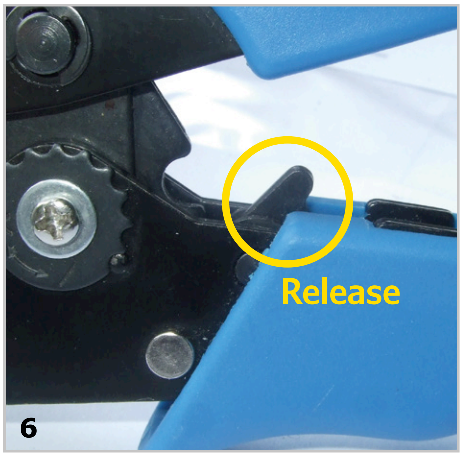
*Once this is correct, carefully ratchet the jaws closed (click, click…) to **gently hold** the crimp (**5**) — you can release them if required using the lever (**6**).*

*To prepare the wire, lay it in place along an unused crimp, and mark where the insulation will be removed (~ 3 mm) and where the crimp ends (**7**).*

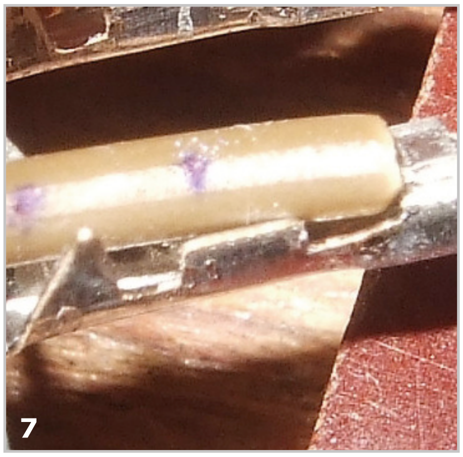
*Now strip the marked area, gently twist the strands, and re-check. It is important that the bare wire does not extend into the working end of the crimp (**8**).*



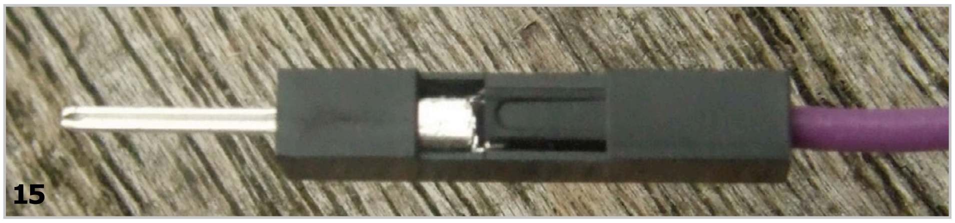
5

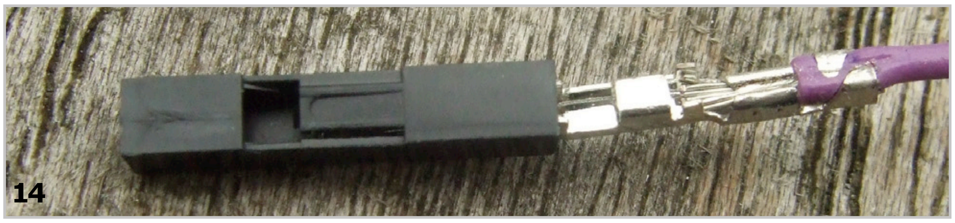


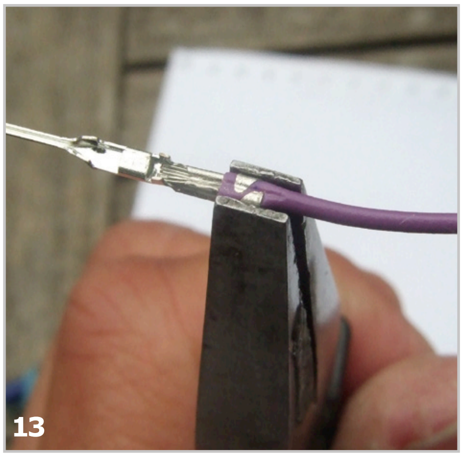
6



7


15


14


13

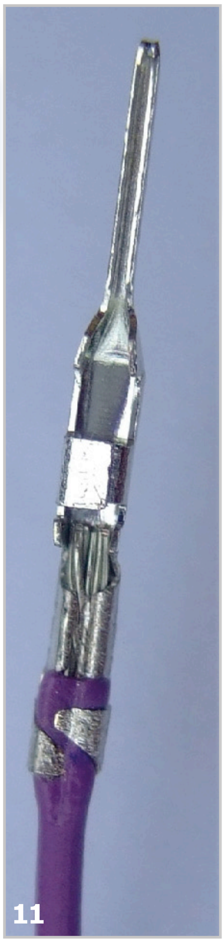
From and for readers

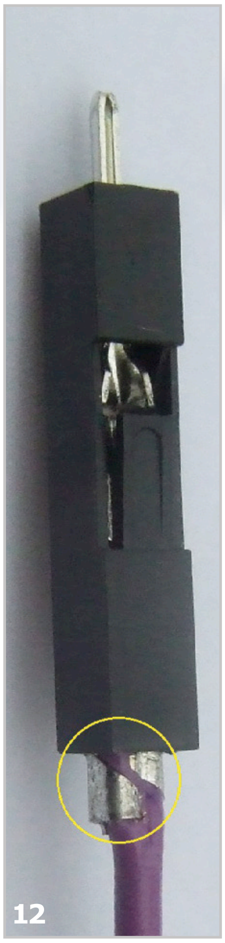
Next, carefully insert the prepared wire into the crimp. For larger-diameter wires it may be necessary to slightly squeeze the insulation to make the wire fit into the crimp. You might just be able to feel when the bare twisted wires touch the working part of the crimp.

*Placement here is absolutely crucial. Too far in, and the wire will interfere with the working of the completed crimp (**9**). Too far out, and the wire or insulation can miss being held by the crimp tabs (**10**). Expect to waste easily 50% of crimps initially.*

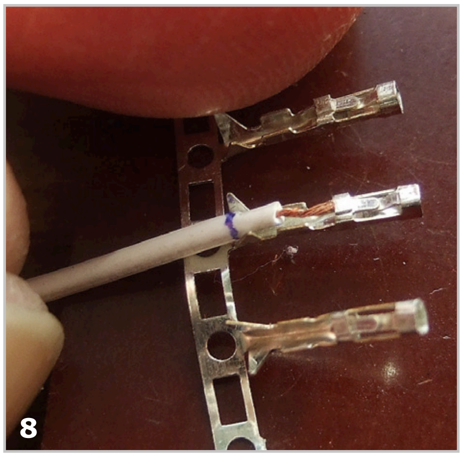
*Fully close the tool handles — the tool will then automatically release. With luck and care, you should be rewarded with a result like **11**. Now, slide the crimped assembly into its housing. It can happen that the wire side of the crimp will not easily push into the housing (**12**). Therefore, it might be that you need to very carefully squeeze the part of the crimp where the wire enters (**13**). And try again: this time it should fit (**14, 15**).*

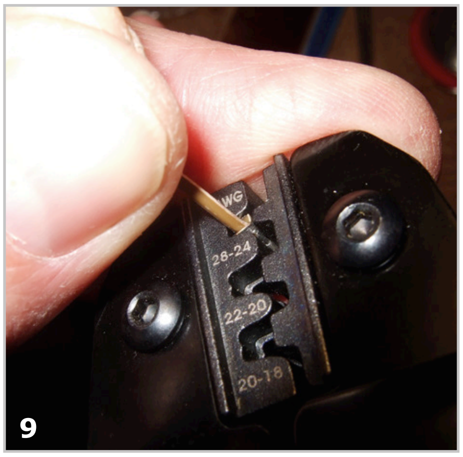
(160192)


11

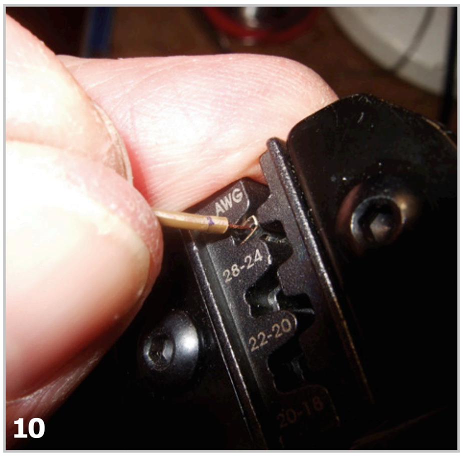

12

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


8



9



10

Q&A

(almost) everything you ever wanted to know about ... CAD

By **David Ardouin** (France)

In the last 30 years, the design techniques used in electronics have changed completely. The omnipresence of computer technology has revolutionized our passion. Memories and evolution…

Q *What does CAD mean?*

A CAD is an acronym of **C**omputer **A**ided **D**esign. The youngest among us have always known and used computer technology, but barely a few decades ago everything was done by hand. This applies especially to electronics, but also to mechanics, acoustics and programming. This computer assistance has resulted in exponential growth in the most advanced fields, from the smallest environmental sensor to the biggest airliners.

Q *Daddy, what was electronics like when you were young?*

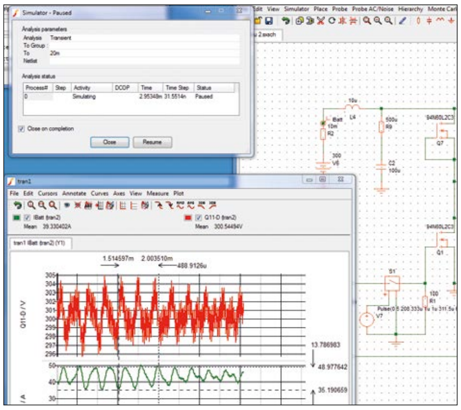
A Those readers with the longest beards still remember… the ‘80s had no Internet, not many screens and paper was king. Hobbyists awaited their favorite magazine so they could photocopy the printed circuit board layouts and make them at home with the aid of foul-smelling mixtures. Personal projects were done with a pencil and eraser. We applied the tracks with a felt-tip pen or with decals directly on the copper. Component specifications were listed in ‘bibles’ of a few hundred pages, and we’d go and stock up at the dealer on the corner, praying that the components we wanted were in stock.

Q *How does Information Technology help us today?*

A Thanks to the internet, we have instant access to a plethora of information. Catalogs and characteristics of components are available with a couple of clicks. Electronic simulators like LTSpice or SIMetrix show on your screen a detailed analysis of your circuit’s functions (**Figure 1**). Plotting the frequency response of a filter, as complex as it is, has never been so simple.

Other applications go even further, such as Proteus which can simulate not only the analog part of your circuit, but also the software embedded into microcontrollers. Some trust it so much that they no longer feel the need to heat up the soldering iron and make a prototype to test their circuits. Drawing schematics can today use databases of symbols provided by the manufacturers themselves.

Software suites are no longer simple drawing tools; they can tell us when there are component interconnection errors and continuously check the rules when laying out PCB tracks. So printed circuit boards (single or multilayer) can attain unequalled levels of detail (**Figure 2**) and the computer can do the component placement and track layout automatically.



Simulator - Paused

Analysis parameters
Analysis Transient
To Group
To 20m
Netlist

Analysis status

Process#	Step	Activity	DCOP	Time	Time Step	Status
0		Simulating		2.95348m	31.5514n	Paused

☑ Close on completion

Figure 1. Overview of the interface of the SIMetrix simulator.

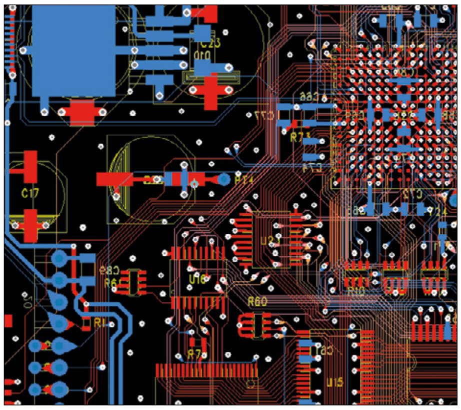


Figure 2. Part of a ‘small’ modern PCB routing.

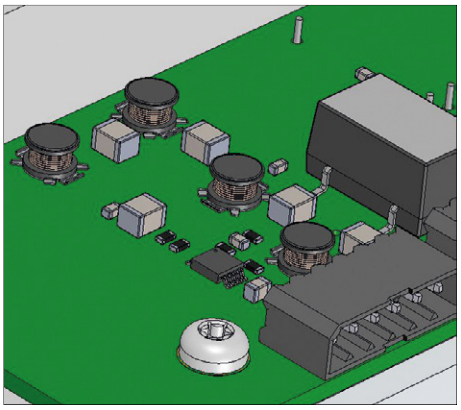


Figure 3. 3D mechanical integration.

Q *What are the most recent developments?*

A In the past few years, electronics and mechanics have benefitted from three dimensional modeling which allows us to study the actual layout on the card (**Figure 3**). That not only simplifies the technical drawings, but it also allows resolution of thermal problems. Finally, it has never been easier, quicker and cheaper to have printed circuit boards made up professionally, no matter where in the world you are.

Q *And what is happening in programming?*

A Strange as it may seem, programs used to be written without the aid of a computer. Instructions (in assembly language) and operands were represented by hole patterns on punched cards. The arrival of high-level languages, much more easily understood by humans, was made possible by compilers, which are actually Man/Machine interfaces. One modern instruction (an operation on an SQL database for example) replaces an unimaginable amount of cards! Nearer our era, we can look at Matlab/Simulink (or the free equivalent Scilab/Xcos) which cleverly mix programming, matrix calculations and a graphical interface (**Figure 4**) to simulate and resolve problems in mathematics or automation.

Q *How are these tools likely to evolve?*

A In our ultra-connected world, we're moving towards collaborative projects and sharing. The profusion of extension cards for open environments like Arduino is daily proof of this. Software designers gain huge amounts of time by reusing base code already tested by others and freely available (saving them the reading hundreds of pages of microcontroller documentation to find the right register). Programming is done more and more in graphical environments (the same way as LabView) to the detriment of the good old line of code. On the hardware side, integrated circuit manufacturers make available online tools (like the WEBENCH suite from Texas Instruments) which can design parts of the schematic, calculate the associated component values, create the component numbering, and create and order the Bill Of Materials (**Figure 5**).

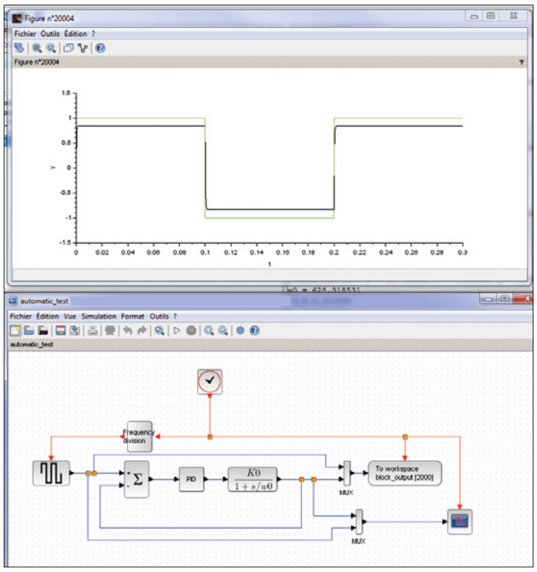


Figure 4. Detail of the SciLab environment.

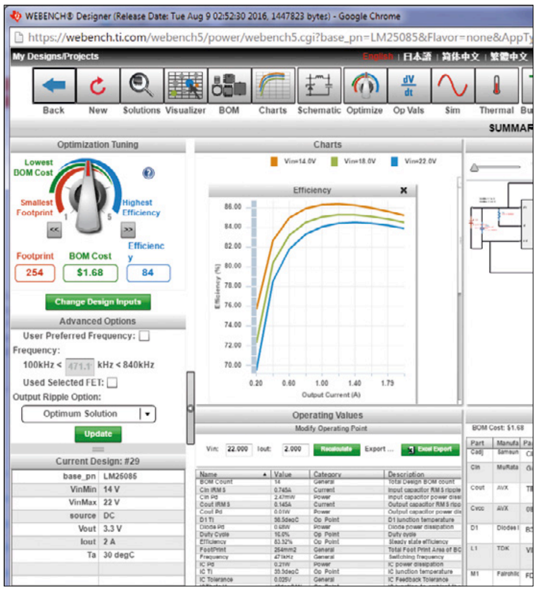


Figure 5. Detail of the online design suite Webench.

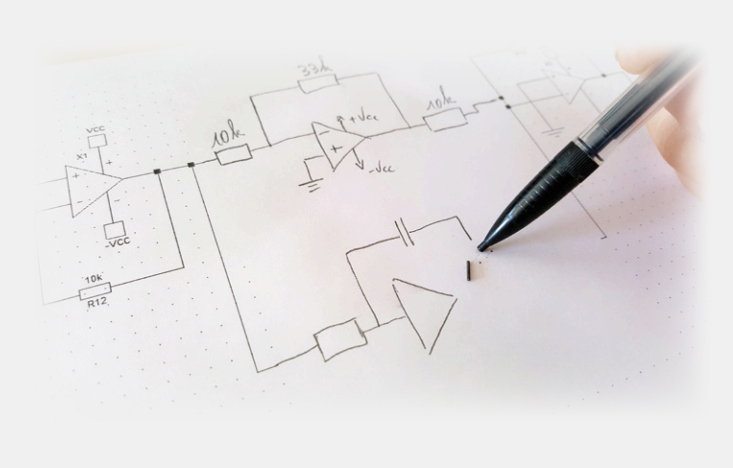
These new ways of designing free the human designers from the boring tasks and let them concentrate on the high level architecture of their systems. Collaborative work and the re-use of existing blocks are encouraged even more by the emergence of development suites that are entirely online. However, and happily for us humans, it will be a long time before the machines, which for a long time have assisted us, end up by replacing us, to the point where it becomes **D**esign **B**y **C**omputer.

(160029)

The Author

David Ardouin is passionate about technology. He is an Electronics Engineer, graduate of ESEO at Angers, France. Among his personal projects are the Ultiprop Clock published in Elektor December 2013 and the touch lock Pattern Lock published in April 2013. He is a father of two girls aged 5 and 6 and has taken the trouble to introduce them to electronics — they are happy to pick up a soldering iron.

His professional activities are mainly hardware design of embedded systems used in industry, renewable energy and aeronautics. He works principally with the PADS (Mentor Graphics), Altium Designer, Atmel Studio, SIMetrix and SolidWorks software suites.



470 m2 ±10% of Elektor @ electronica 2k016 Munich

At *electronica 2016* in Munich, on November 8-11, Elektor International Media will have its largest exhibition floor space in the history of the publishing company. Those 500-odd square meters are due to Elektor having teamed up with Messe München, the *electronica* show organizers, to jointly roll out the prestigious **Fast Forward Award 2016**. FFA comprises video interviews, presentations, lectures, and culminates in a grand prize award ceremony on Friday.

But it's not all FFA on the brand new stand: the Elektor team present at the show is ready and eager to meet up with everyone with an interest in the projects and articles published in the magazine and on our labs website. Subscriber or not, come chat with Elektor editors, designers and book authors, or discuss (and touch!) recently published projects like **Würth Wireless Power Kit**, **500 ppm LCR meter**, **SDR Reloaded**, **Elektor Uno R4**, **Nixie Clock**, **SmartScope**. Or preview stuff under wraps in the labs like **New Red Pitaya**, **Sandclock**, or **BBC Micro:bit Dock**, and surely some more stuff as we so much space available! At the show you can also order Elektor products and have them sent to you free of charge if not available at the sales desk. We're also happy to sign you up as a new member, or help you with issues.

Come see us and talk **electronics** at *electronica*. Like that curious number **470** and tolerance in the title, does it ring any E12 bells?

Booth EOE 200, East Entry, Munich Messe, November 8-11.

Rebellion race car wins LMP1 Class in Mexico Endurance

The Mouser-sponsored No. 13 (!) Rebellion R-One AER car secured fifth place overall and the win in the LMP1 Privateer Class on September 3 in the first-ever 6 Hours of Mexico, the fifth round the 2016 FIA World Endurance Championship.

Rebellion Racing, sponsored this year by Mouser Electronics and LEMO, significantly strengthened its position in their bid for the championship. Next, the team ventures to Austin, Texas, on September 17 for the 6 Hours of the Circuit of the Americas, a 3.4-mile course that features 20 tough turns.

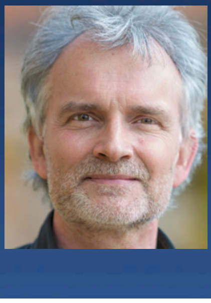
Mouser distributes LEMO's products, including a wide range of push-pull circular connectors and module connector solutions. LEMO connectors are ideal for harsh environment, medical equipment, test and measurement instrumentation, telecommunications, audio/video, and other applications.

The race in Austin will be followed by races at Fuji, Japan, October 16; Shanghai, China, November 6; and the season finale in Sakhir, Bahrain, November 19. The Rebellion Racing team kicked off the 2016 FIA WEC racing season on April 17, finishing third and fourth at Britain's 6 Hours of Silverstone and taking the top two positions in the LMP1 Privateer class. It was an identical finish at Spa-Francorchamps, where Rebellion's No. 13 car came in a strong third, closely followed by their sister car, No. 12. Both had strong outings at Germany's renowned Nürburgring and held onto their 1-2 standing in the Privateer LMP1 class through the always-challenging Le Mans endurance race.

www.mouser.com (160186-2)

The ElektorBusiness section in Elektor Magazine accommodates articles, news items and other contributions from companies and institutions active in electronics. Publication is at the discretion of the Editor. Contributions to: newsdesk@elektor.com

Jan Buiting,
your ElektorBusiness Editor



New Microchip Explorer is 16/32-bit

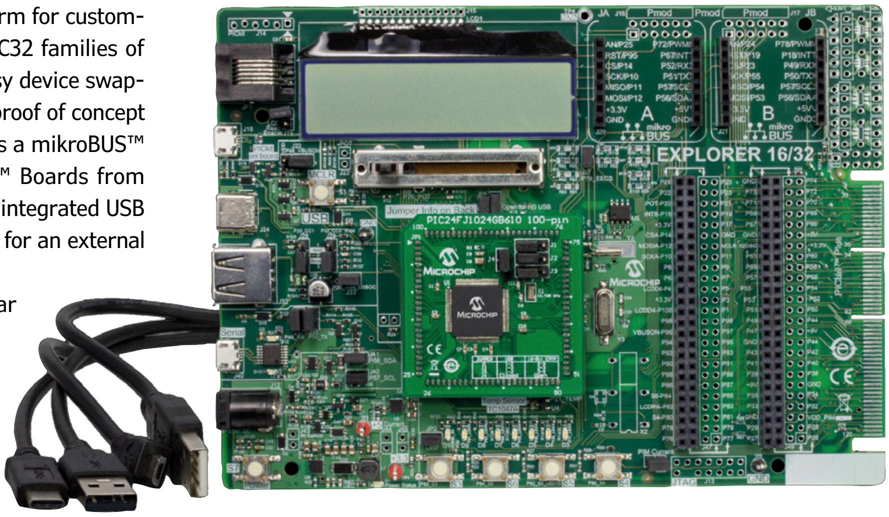
Microchip' new Explorer 16/32 Development Board or PIC® devices is offered at a lower cost than the popular Explorer 16 and comes with an integrated programmer/debugger and several new features that address the latest embedded systems design needs. The board provides a flexible, convenient and easy-to-start tool while being backwards compatible to the classic Explorer 16 Board.

The Explorer 16/32 Development Board serves as a platform for customers to evaluate the 16-bit PIC24, dsPIC33 and 32-bit PIC32 families of devices through Processor Plug-In-Modules (PIMs) for easy device swapping. It facilitates prototyping of the end application for proof of concept before migrating to the actual design. The board features a mikroBUS™ interface to easily add new functionalities using Click™ Boards from MikroElectronika. In addition, the board also supports an integrated USB for both power and communication, eliminating the need for an external power adapter and communication accessory.

This new board is backwards compatible with the popular Explorer 16 Board which allows users to continue using existing code, libraries, prototypes, PIMs, and PICtail™ Plus daughter cards interfaced via a side PICtail Plus connector. The PICtail Plus daughter cards interfaced via a vertical connector can be re-used using the additional PICtail Plus Expansion Board.

The Explorer 16/32 Development Kit (DM240001-3) includes the main Explorer 16/32 Development Board as well as a PIC24FJ1024GB610 Plug-In-Module and two USB cables. The Plug-In-Module features a 16-bit PIC24FJ1024GB610 microcontroller with 1 MB Flash, a superset of memory and functions for the lower power PIC24F family. This kit with PIM and cables is priced at $109.99.

www.microchip.com/Explorer1632 (160186-5)



QRV on 2.4 GHz with educational radar demo kits

Pasternack's PEM11000-KIT and PEM11002-KIT radar demonstration kits allow classroom instruction and R&D for the study of fundamental radar principles. The kits cover the popular 2.4-GHz industrial, scientific, and medical band with no special licensing required.

Experimental radar operating modes can be selected for CW, FMSW, or Doppler. Output waveforms can be set for Single Tone, Frequency Ramp, or Sawtooth.

This unique kit contains all digital and RF hardware that features a radar board that includes an integrated speaker and signal indicator to support both audio and visual feedback of received signal strength. A tunable signal filter is also included to allow for design and implementation of customized passive or active filtering.

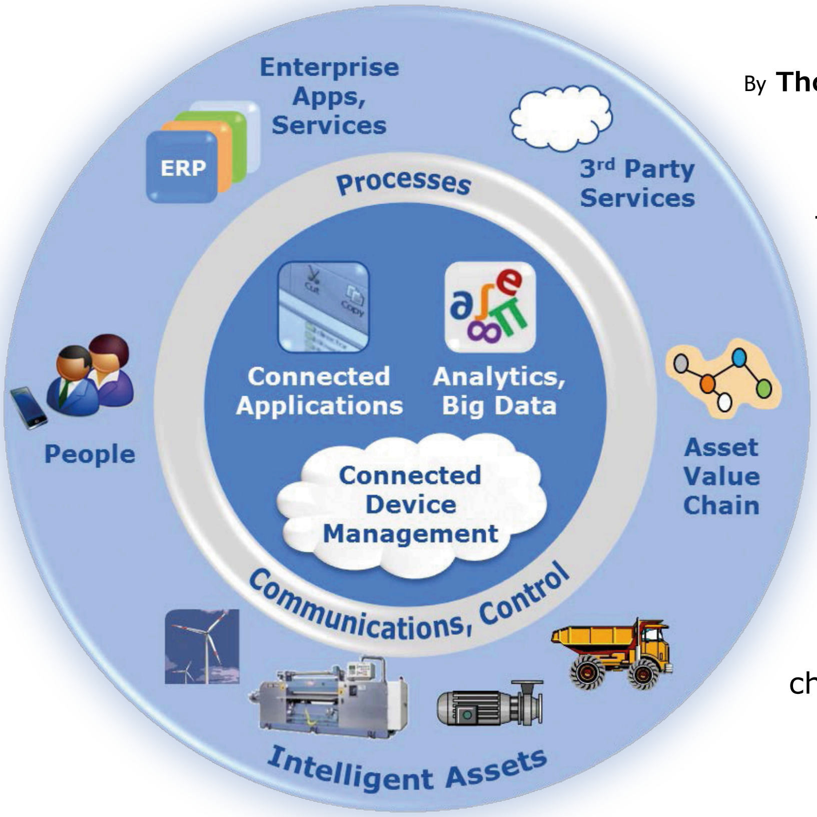
Programmers! Your radar demonstration unit is command-controlled via a downloadable graphical user interface for Windows-based computers using the VISA (hey?) protocol. Newbies! The connection is made using a simple USB or Bluetooth connection.

The PEM11002-KIT model comes complete with a radar board, antennas, cables, and accessories that include a mounting plate, tripod, and USB battery pack. The kit includes a detailed user guide and programming manual as well as a lesson guide.

www.pasternack.com (160186-6

Industry 4.0

from the Perspective of a Semiconductor Manufacturer



By **Thomas Brand**, Applications Engineer, Analog Devices

The industrial world is in the midst of a paradigm shift commonly designated as "Industry 4.0" in Germany. In international usage, it is also called the Internet of Things, the Industrial Internet of Things, Smart Factory, Cyber Physical Systems, or Cyber Physical Production Systems. What this involves is the global digitalization and networking of the industrial value creation chain and its products.

Particularly in Germany, Industry 4.0 is not only held in high esteem in the commercial sector; it is also strongly promoted in the political sector where the economic initiatives primarily focus on improving the international competitiveness of Germany as an industrial nation. In this regard, political leaders are pursuing a dual strategy which combines the perspective of the supplier industry with the perspective of user enterprises, predominantly in automation as well as mechanical and plant engineering. The aim here is twofold: on the one hand to deploy the latest technologies for in-house production, and on the other hand to commercialize these technologies and products.

Dual strategy

From the perspective of leading vendors, who view the potential benefits of Industry 4.0 from the supplier industry standpoint, the main objective to is expand the production landscape and its technologies by merging information or communications technologies with conventional high-tech approaches. This also creates the conditions necessary for managing increasingly dynamic markets and rising complexity. From the perspective of production enterprises, which means the leading markets, the objective is to fashion smart technologies and products for new markets and then serve these markets.

Enterprises which are able to engage with both perspectives appear to be especially well positioned to pursue this sort of dual strategy. Companies in the semiconductor industry are particularly good examples of such enterprises.

For example, semiconductor manufacturers such as Analog Devices are working on the complex project of converting their production facilities into fully automated factories (dubbed "smart factories") and at the same time supplying innovative technologies and products to companies in the production industry and supporting them in the implementation of their own smart factories. Small and medium-size enterprises (SMEs) have a particularly high significance in this regard because they form the vast majority (98%) of industrial companies in Germany and offer correspondingly high potential. Exactly what form this support might take, the challenges confronting semiconductor manufacturers in this regard, and the accompanying opportunities or perspectives, are described in more detail below.

Smart factory: yes or no?

The first thing an enterprise needs to consider is whether converting to their own

smart factory would be profitable or only represent useless expense.
Countless studies addressing this key question have already been carried out by a wide variety of market research companies. They all share the conclusion that the advantages of Industry 4.0 vastly outweigh the disadvantages, for which reason the opportunities it presents should be grasped to ensure the continuity of the enterprise. For semiconductor manufacturers in particular, the conclusions can be further divided into three categories: new technologies, new product portfolios, and new business models. They cover the entire value creation chain of production and products, from sensors through the cloud to downstream services.

For the realization of smart factories, there is an urgent need for smart and energy-efficient products, or even completely autonomous systems, which can be integrated into the existing production matrix as easily as possible in full keeping with the plug & play principle. These systems consist of a wide variety of semiconductor components and sensors, suitably combined with analog and digital signal processing circuitry. The value creation chain starts with these systems because they first acquire the data from the physical layer and then handle the conversion, conditioning and processing of the data in the digital layer (see **Figure 1**).
Considering this value creation chain, market requirements will change in the future, for which reason semiconductor manufacturers must also adapt their business and their products to the circumstances of smart factories. Here the technological trend is increasingly heading in the direction of smart and extremely energy-efficient products, as much as possible with integrated safety and security functions or energy harvesting capabilities. **Figure 2** shows what this sort of component or entire system might look like, using the MEMS-based ADXL356 acceleration sensor from Analog Devices as an example. This device is a low-cost, extremely low-noise three-axis acceleration sensor with a measuring range of ±40 *g* (FSR), which is amongst the best in its class thanks in particular to its very low offset drift and low power consumption. Its hermetic package makes it suitable for applications such as precise inclination angle measurements in harsh environmental conditions, high-resolution vibration measurements, and high-performance measurements (including long-term measurements) in low-power or battery-operated wireless sensor networks, such as are used in structural health monitoring, aircraft attitude heading reference systems, and a variety of other applications in which large forces can suddenly occur.

The ADXL356 thus forms the basis of the system, which is augmented with additional functions including integrated signal processing by A/D converters and microcontrollers or various analog and digital interfaces, some of which are isolated or fully wireless, which support communication standards such as Ethernet, 6LpoWPAN or ADRadioNet. The last of these is a standard developed in-house by Analog Devices, which features a wireless, self-healing multi-hop communication protocol, high scalability and low memory demand. In addition, integrated encryption algorithms such as AES-128, AES-256, etc. help to mitigate fears of potential cyber-attacks. Functional safety, which is also becoming increasingly important in the smart factory environment, additionally plays a key role in the development of these systems.

Not just sensors

For semiconductor manufacturers, simply expanding their portfolios of sensors and other semiconductor devices will not be sufficient in the future. To continue to be regarded as global players in the market and to be able to fully exploit the potential of Industry 4.0, they must expand their existing product portfolio to include communications technologies and low-power microprocessor technologies. Analog Devices already has some very promising solutions in its product lineup. In the communications area, for example, they offer HART solutions that allow conventional, overloaded communication paths to be upgraded relatively easily with high-performance, energy-efficient technologies while utilizing the existing infrastructure.

Another concrete conclusion of the studies is that companies want comprehensive support from their suppliers for faster

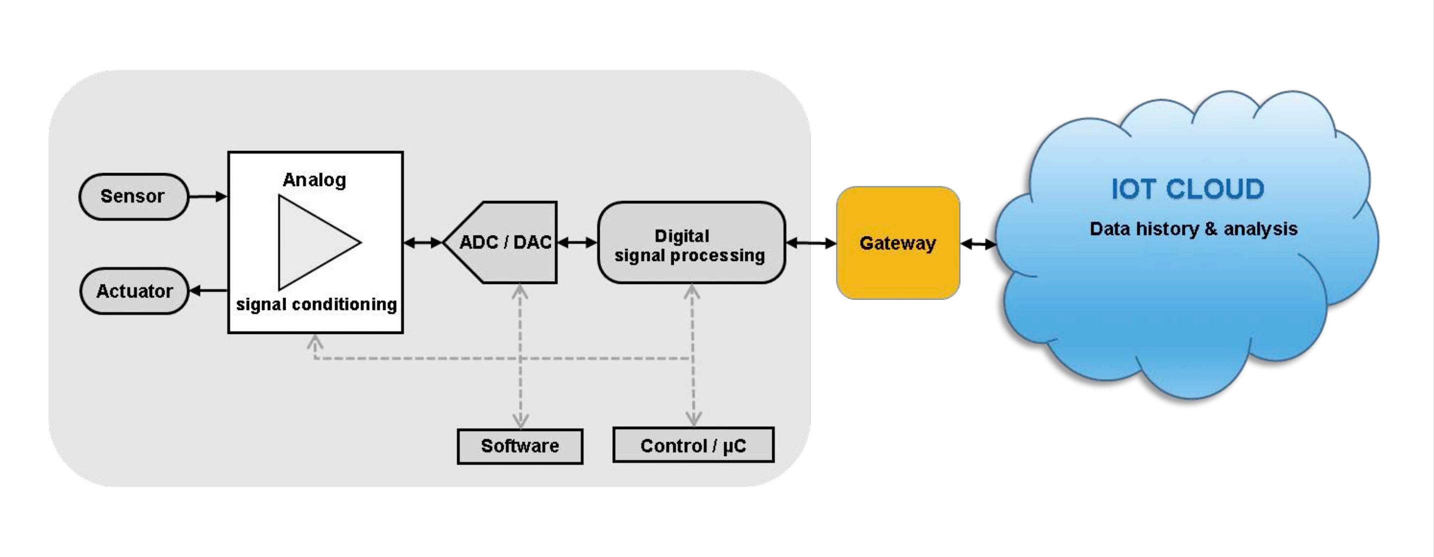


Figure 1. The value creation chain of the smart factory.

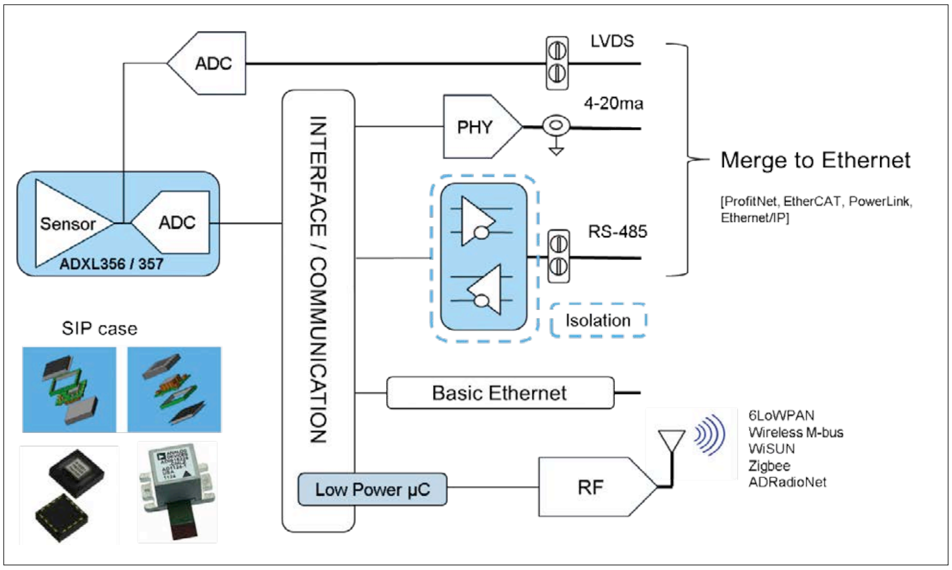


Figure 2. A MEMS-based smart sensor solution.

and more effective integration of future products. The aim is to offload additional effort from the company so that it can continue to concentrate on its core tasks. Partnerships or collaborations are considered as potential forms of support.

Financial matters & more

Considering the many advantages of Industry 4.0, we have to ask ourselves why many companies are hesitating to invest in their own smart factories.

One of the significant hurdles is that the advantages of smart factories and the associated investments are not yet widely recognized. For instance, it is presently very difficult to calculate the return on investment because there is a major lack of monetary evaluation criteria. It is therefore incumbent on semiconductor manufacturers to carry out the necessary educational work and demonstrate to their customers the advantages of smart factories and the added value of the required investments.

However, these investments are not limited to SMEs which wish to implement their own smart factories. In some cases, providers of smart solutions must also invest heavily in their own product portfolios, as well as any necessary reorientation of their various business areas. Investments are necessary in new technologies and business models, and in setting up or expanding marketing activities, qualification measures and distribution management.

With further transformation of the production landscape, topics such as data protection and IT security will repeatedly play a key role. In addition to being important prerequisites for the successful launch of Industry 4.0, they must be consistently and comprehensively anchored in the digital systems as essential acceptance and success factors. For companies it will also be important to define the strategic roadmap for their Industry 4.0 initiatives at an early stage, and to acquire experience with corresponding technologies. However, this process requires patience because many of the technologies relevant to Industry 4.0 are only expected to be able to exploit their full efficiency potential after 5 to 10 years. Consequently, the implementation of Industry 4.0 will be a prolonged process. In addition, due to the lack of a clear definition of Industry 4.0 there is no specific point in time when Industry 4.0 will be fully implemented. For this reason, it is often called an industrial evolution instead of an industrial revolution.

With the transformation of the production landscape, entirely new players and competitors are also appearing on the market. This leads to alliances that will shape the appearance of future production. The future of an industry or a company will be determined less and less by what happens in its production facilities. Instead, factors such as software and services will play a key role because comprehensive productivity potential can only be achieved by linking and merging the real world with the virtual world.

Be part of it

It appears that companies now realize that current developments are moving very quickly and catching up is very difficult. Accordingly, anyone who still dismisses Industry 4.0 as a passing fancy has already lost the race. Current market events clearly show that a distinct and irreversible trend has developed. The big question mark with regard to the lack of standards, the lack of certainty, insufficient knowledge of relevant topics, etc. is becoming noticeably smaller, and the confusion about the future of the production landscape is gradually disappearing.

In summary, we can say that Industry 4.0, which means the shift to smart factories and the technologies necessary for this, offer enormous potential for semiconductor manufacturers and the success of their enterprises.

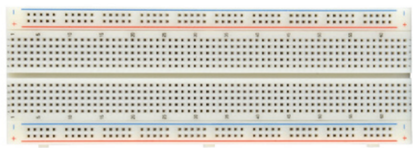
(160170-I)

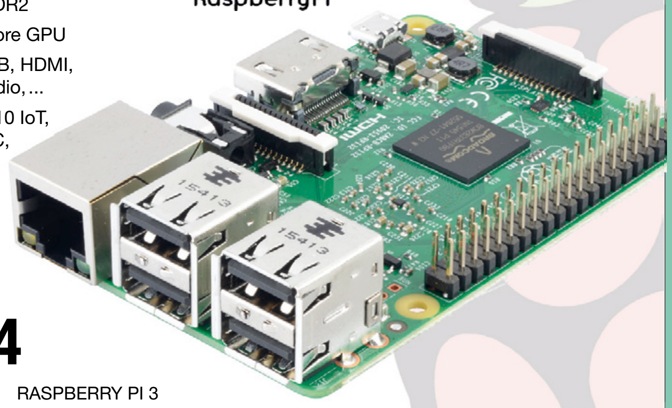
September and October 2016 | € 4,95
elektorbusiness
HOMELAB
STARTUP
TRADE
IoT and Industry 4.0
Technology · Research · Development
4.0

Learn more about Industry 4.0

The IoT and Industry 4.0 issue of Elektor Business Magazine is available for free downloading, also for non-members:
www.elektormagazine.com/files/attachment/3101

Guest (Offline)
pi-top
11:39
Tue
03 Nov
pi-top





The Big Red passion

The Big Red is about passion. **Passion** to dare to be a prototyper, startup or homelab in electronics. **Passion** to share your eagerness with the outside world. Passion about your product or service. **Passion** about that technical breakthrough that was absolutely necessary to become unique. **Passion** so limitless you even want to put your financing in order before contemplating any PR. **Passion** to tell us, the editorial staff of Elektor, what your company is all about.

Who wouldn't want to be so passionate?

Outlook

Trends without the crystal ball

Start-ups 2016

Spirit, sparks and more

Financing & Funding

It's your Foundation

Technologies

Your all-you-need-to-know guide

Company Pages

Presenting from your own newsroom

Elektor Talents

Seeking new colleagues differently

Back to the Future

Starting up? Making history!

RT UP
THE BIG RED 2017
THE BIG RED
2017
Inspiratory Book for Prototypers, Homelabs & Startups from the makers of elektor business
THE BIG RED 2018
THE BIG RED 2019
THE BIG RED 2020
NEW

Welcome to the **DESIGN** section

By **Clemens Valens,** Elektor Labs

Michael Faraday (1771 – 1867)

Considered by many as one of the greatest scientists that have ever lived, Michael Faraday has two units named after him: the faraday and the farad. The first is not a SI unit, and somewhat obsolete, the second is the unit of interest for this piece as it is the unit *par excellence* to express the value of a capacitor. Originally the farad was suggested as a unit of electrical charge, but eventually became the unit of electrical capacitance in 1873 when the newly formed Committee for the Selection and Nomenclature of Dynamical and Electrical Units – then including Thomson (the future Lord Kelvin), Maxwell and Siemens – published its first report. It was Latimer Clark, together with Charles Bright, who, in 1861, had proposed the farad as a unit for electrical charge together with the ohma for potential, the volt for resistance and the galvat for current. If they had it their way, today we would say that a current of one galvat through a resistor of one volt produces an ohmage of one ohma: $\Omega = I \times R$.

Now why did Latimer Clark and Charles Bright propose units anyway? Who were they to be taken seriously by the British Association for the Advancement of Science? Delving into history, we quickly discover that Latimer was the brains of the couple whereas Charles had the money. Latimer's career took off when he became chief engineer at the Electric (and International) Telegraph Company. One major concern of telegraph companies at that time was regarding the cables and especially their losses. At the 'electrical' level everything was not yet fully understood, and things were turning even more complicated by a lack of standardization; every engineer measured and categorized cable resistance in his own way. Having acquired a lot of expertise on cables, Latimer entered into a partnership with Charles Bright, who had completed the first Atlantic cable in 1858. Although this cable didn't last long, Charles continued laying submarine cables and was involved in laying the first truly lasting transatlantic cable. The pair set out to improve submarine cables and found that a unified system of electrical units would come in handy. Since there wasn't one, they invented it.

£20
£20 BANK of ENGLAND
TWENTY
POUNDS
ROYAL INSTITUTION CHRISTMAS LECTURES INITIATED IN 1826
THE MAGNETO ELECTRIC SPARK APPARATUS
© THE GOVERNOR AND COMPANY OF THE BANK OF ENGLAND 1993
Michael Faraday
1791-1867

Latimer was actually quite a brilliant mind. In 1854 he took out a patent on pneumatic tubes and became involved in a Hyperloop-like project for dispatching parcels in London. (If that doesn't sound like 'Back to the future', what does?) His moment of glory came in 1895 when the very stable 1.4328 V battery that he had invented, the Clark cell, was elected international voltage standard and successfully held that position for over a decade.

And Michael Faraday in all this? Ah, he was just being a genius, inventing cool theories about lots of stuff, some of which happened to be related to electrical charge and submarine cables. Latimer knew Michael, having presented some of his experiments to him in 1853 or 54, and, possibly as a sign of admiration, proposed the farad as a unit in return. And it stuck.

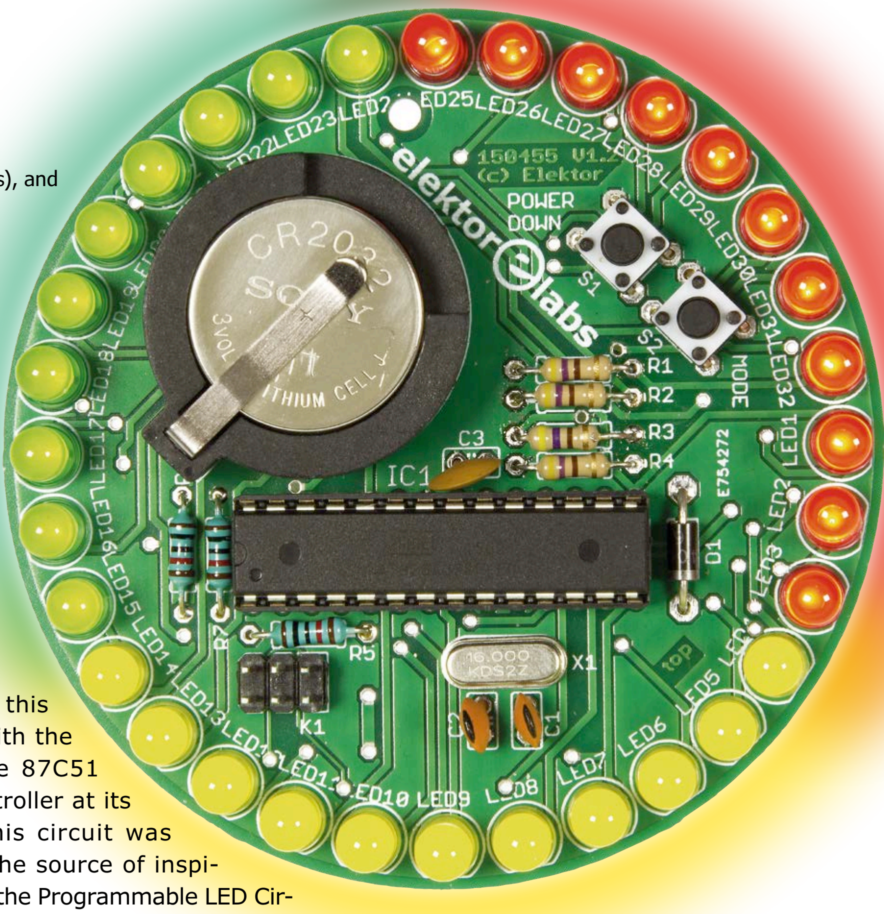
(160177)

Programmable LED Circle

Arduino compatible

By **Niek Laskarzewski, Roy Aarts** (Elektor LABs interns), and **Thijs Beckers** (Elektor Editorial)

During the darker days we're very keen to light things up a bit. (Christmas) lights are sprouting from the ground like mushrooms. This programmable LED circle fits perfectly in this picture, but is, of course, also suitable for other occasions. We already have Mardi Gras in mind...



It's been over 18 years since Elektor published a project that controlled 32 LEDs with a single microcontroller. To be more specific, the article we are hinting at is called *Christmas Running Light* and was published in the December issue of 1997. **Figure 1** shows the simple schematic for this circuit, with the venerable 87C51 microcontroller at its heart. This circuit was used as the source of inspiration for the Programmable LED Circle described there.

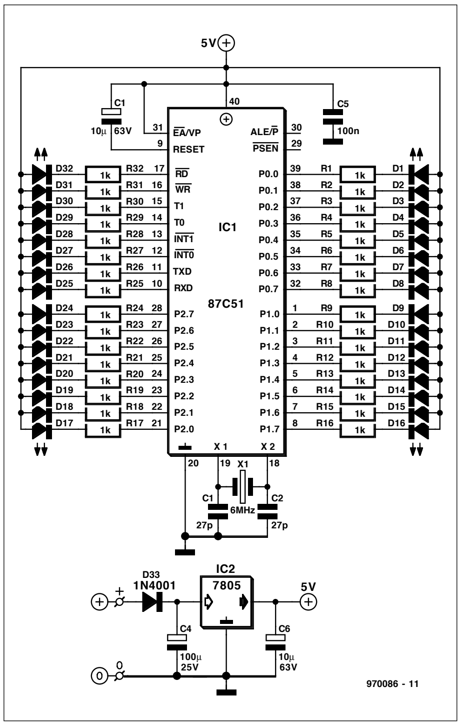


Figure 1. The schematic for the *Christmas Running Lights* from 1997.

Compatibility

In contrast with the microcontroller in the 'proto-circuit', the controller in the Programmable LED Circle can be very easily programmed yourself. It is namely an ATmega328P, the very same as is used in the immensely popular Arduino Uno. And, incidentally, also our own 'T-Board 28'. The advantage of this is that you can program this controller yourself using an Arduino Uno and therefore do not have to purchase a separate programmer.

The design

The 87C51 that was used originally has the advantage that all 32 LEDs could be controlled directly, one LED per pin. The ATmega328P has a total of 'only' 28 pins and therefore does not offer this as an option. So, a clever method is therefore required in order to control this many LEDs.

In this design we have connected the LEDs in a matrix of 4×8 (see **Figure 2**). In this manner we can control 32 LEDs, while using only 12 connections to the microcontroller. Of course, we could also have used a 74HC595 shift register, but that immediately makes the circuit board more complicated. Particularly if we would like to keep the diameter of the LED circle to less than 7 cm.

For the power supply for the circuit, the idea is to use a couple of button cells of the type CR2032, so that there is no need for the circuit to be tethered with a wire to a power supply and the decoration can be hung in any arbitrary location, independent of the location of a wall outlet. This does, however, mean that we have to keep an eye on the current consumption. For this reason, the current limit resistors R1 through R4 have deliberately been given a somewhat higher value.

D1 prevents a dead microcontroller should the button cells inadvertently be connected the wrong way around. C3 serves as power supply decoupling for the microcontroller. And finally, the quartz crystal with a frequency of 16 MHz, together with loading capacitors C1 and C2, determine the clock speed.

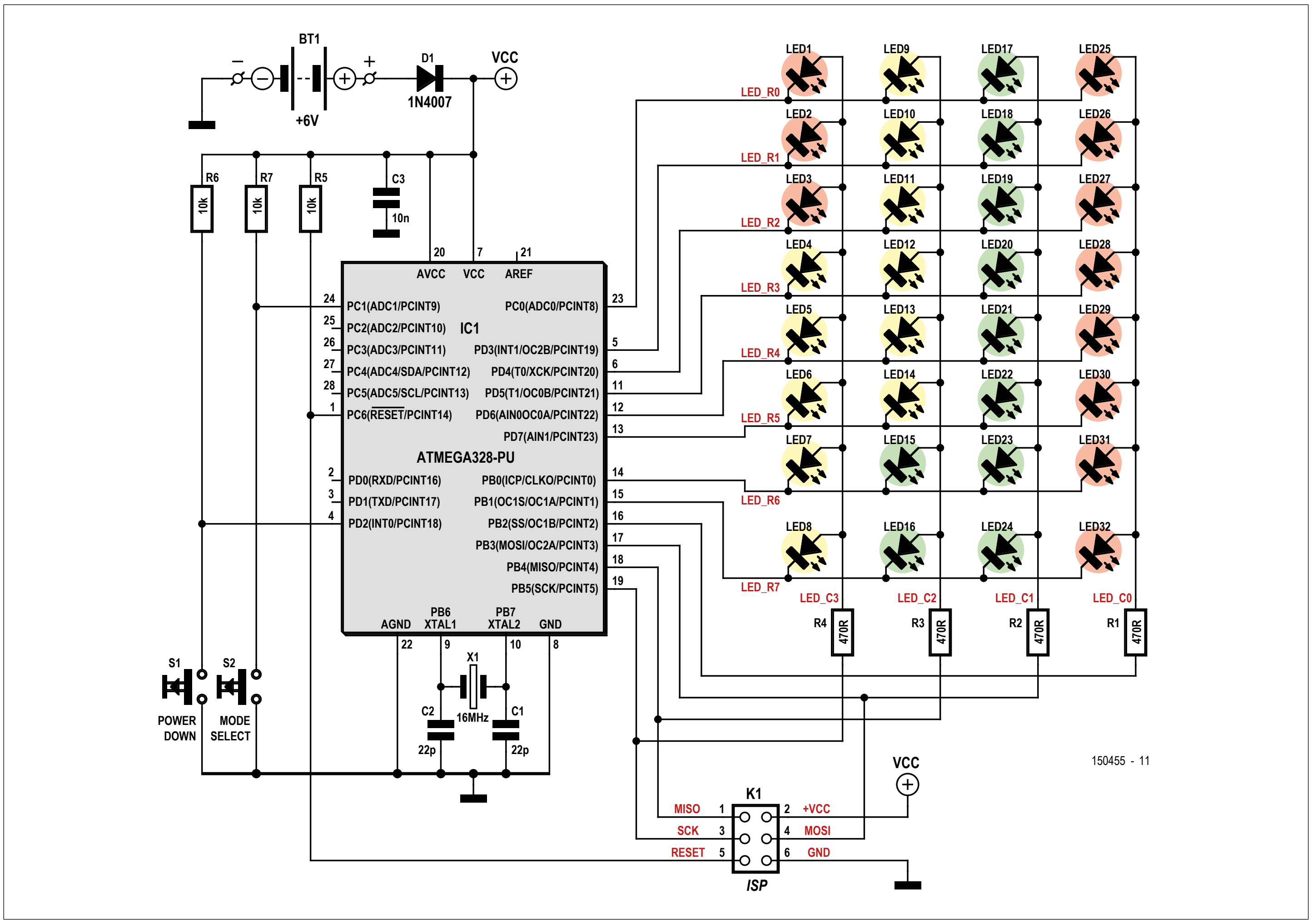


Figure 2. The circuit is built around an ATmega328P. This drives the LEDs in a clever way using a matrix.

When used with the original firmware, we can switch the microcontroller into the power-down mode using switch S1 and also wake it back up. S2 allows different lighting patterns to be selected.
Before starting the printed circuit board layout, it is always a good idea to validate the schematic first. And so our first prototype was soon brought to life (see **Figure 3**).

Software

A simple piece of software was written to test the matrix and the brightness of the LEDs. In order to control this many LEDs a matrix is unavoidable. The disadvantage is, of course, that each LED will receive less current on average and is therefore not as bright. The ATmega328P can handle a maximum current of 40 mA per I/O pin. For that reason, and because the circuit is battery powered and therefore has to be relatively frugal with current, it was decided to light up no more than 4 LEDs simultaneously. From this follows that we have to limit the current to a maximum of 10 mA per LED. Of course, we could have decided to turn all LEDs on at the same time. In that case the 40 mA output current limitation on pins PB2 through PB5 would have limited the current to 5 mA per LED. With the efficiency of LEDs that are currently available, this could also have been an acceptable choice. This can, of course, be easily changed in the software; the schematic has been designed in such a way that this is possible. But as became apparent from the first prototype, there

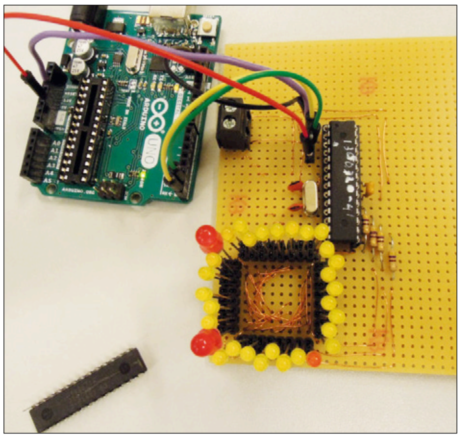


Figure 3. Before the PCB layout was started the circuit was tested on perfboard.

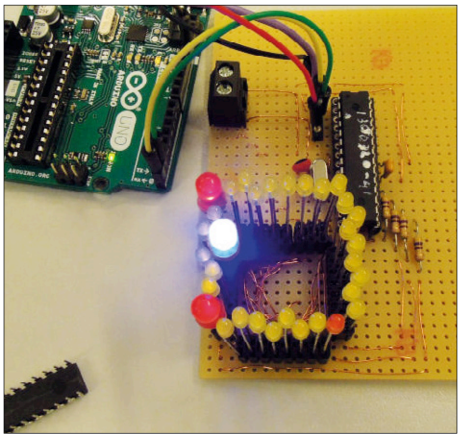


Figure 4. A recent, efficient LED is happy with much less current in order to light up brightly.

Component List

Resistors
R1–R4 = 470Ω
R5,R6,R7 = 10kΩ

Capacitors
C1,C2 = 22pF 50V, 2.5mm pitch
C3 = 10nF 50V, 2.5mm pitch

Semiconductors
D1 = 1N4007
IC1 = ATmega328P, programmed, Elektor Store # 150455-41
LED1–LED32 = LED, 5mm, high efficiency, 2.5mm pitch

Miscellaneous
K1 = 6-pin (2x3) pinheader, vertical, 2.5mm pitch
S1,S2 = pushbutton 24V, 50mA, 6x6mm
X1 = 16MHz quartz crystal
Button cell holder for 2 x CR2032
Socket for IC1, DIP-28, narrow

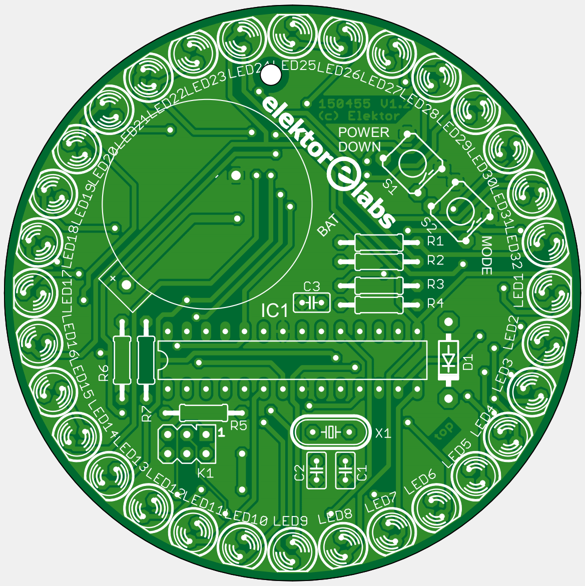


Figure 5. The LEDs fit neatly around the outside edge of the round circuit board. The remaining components all find space inside the circle.

are still plenty of LEDs around that require a little more current. With up to 10 mA per LED available you have a little more wiggle room and the option to increase the brightness somewhat.

However, we have to note that the current limit resistors limit the current for each LED a little more. In the most favorable case of a red LED with a V_f of 1.8 V and a power supply voltage of 6 V there will be a maximum current of ((6–1.8)/470 =) 8.94 mA. At V_f = 3.3 V, as is typically the case with white and blue LEDs, and a 5 V power supply voltage there is only ((5–3.3)/470 =) 3.62 mA left. In **Figure 4** we can see that this is still more than sufficient for the present generation of efficient LEDs.

The circuit board

Once the tests on the prototype were completed successfully, the light was green for the design of the circuit board (see **Figure 5**). In order to fit the LEDs neatly in a circle, a circuit board with a diameter of 76.2 mm was required; just a little bigger than the target size. The (through-hole) components all fit neatly within the circle. There is even room for an ISP connector, which can be used to program the microcontroller using an ISP programmer, avoiding the intermediate step of an Arduino Uno.

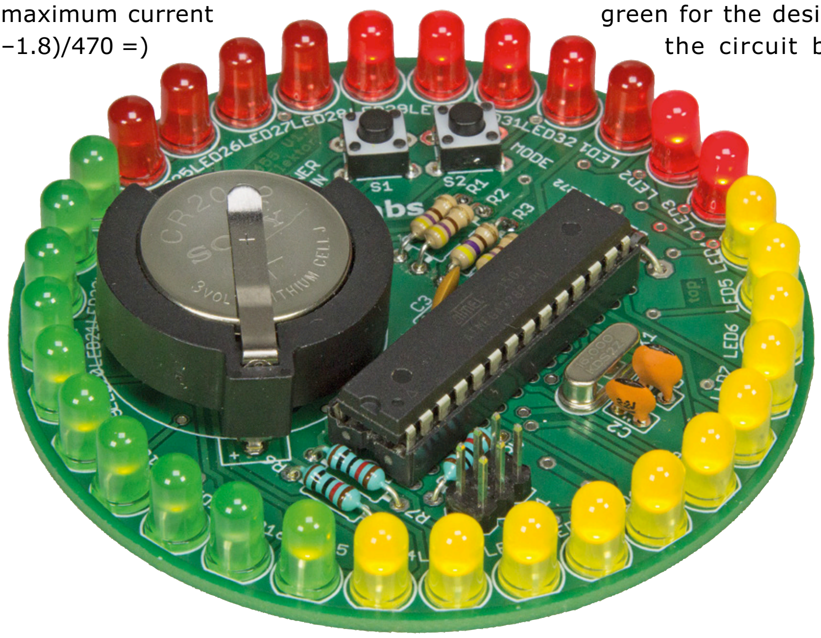


Figure 6. In preparation for Mardi Gras, we selected LEDs in matching colors.

The assembly of the circuit is dead simple because of the use of though-hole components. Start with the components that have the least height (the resistors and the diode) and then solder the remaining components according to their height. In this way it is easy to keep the components in their proper position when the circuit board is turned over for soldering. **Figure 6** shows the final result of the assembled circuit board.

The Arduino sketches can be downloaded from [1]. In order to be able to program these sketches from within the free Arduino development environment [2], the ATmega328P needs to be pre-programmed with the Arduino bootloader. Every ATmega328P in an Arduino Uno is provided with this bootloader. If necessary, you can order 'spare' microcontrollers from our online Store [3]. If you have neither an Arduino Uno nor a programmer then you can order a completely programmed ATmega328P from our Store [4]. And to make it really easy for you, we also offer a complete kit including the programmed controller [4].

Provided all the components are mounted correctly and the microcontroller is programmed with the correct firmware then there is nothing that should prevent the circuit from functioning correctly when the button cells are fitted. When you are tired of the pre-programmed patterns, then you can easily program your own patterns using the Arduino development environment. ◀

(150455)

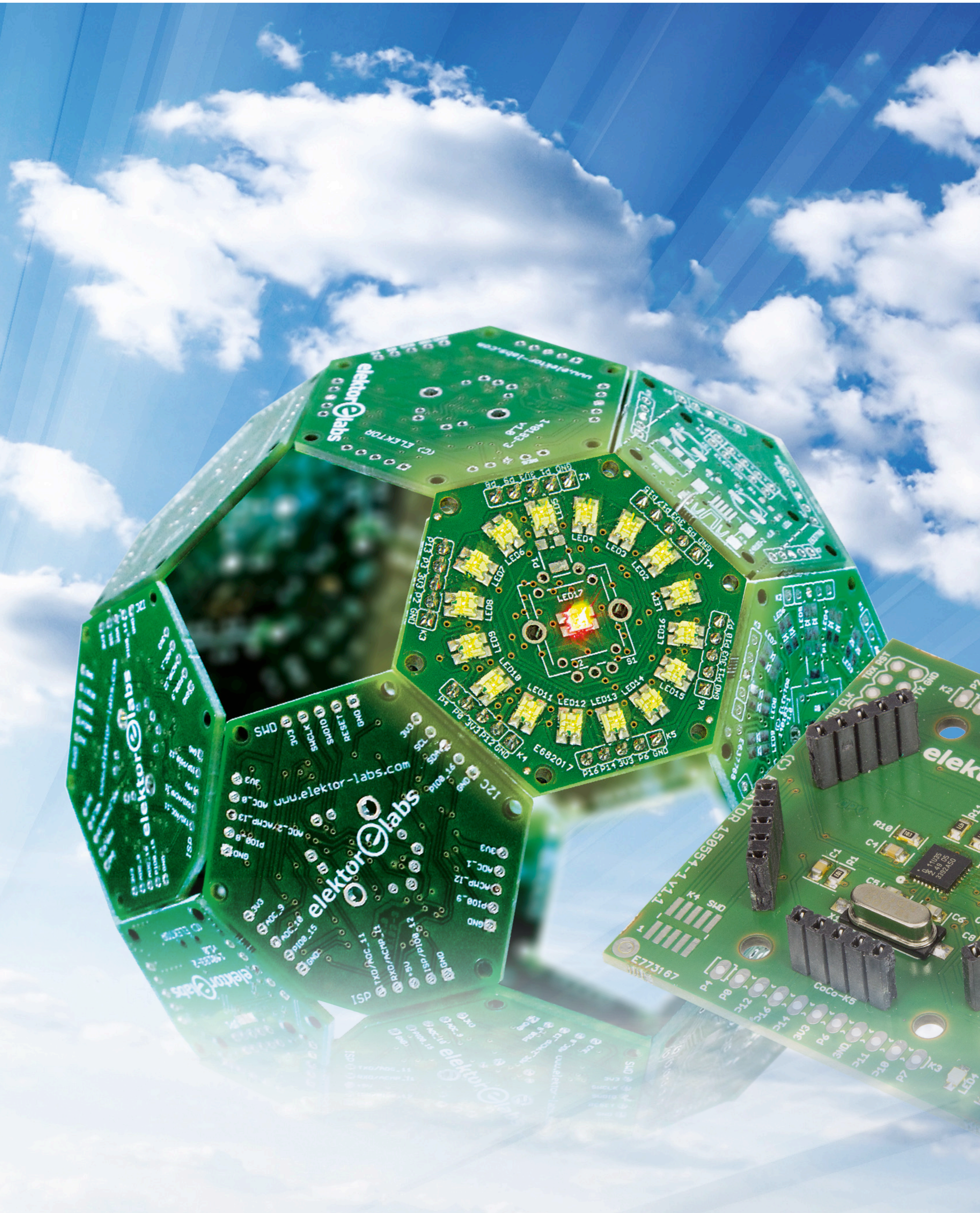
Web Links

[1] www.elektormagazine.com/150455

[2] www.arduino.cc/software

[3] www.elektor.com, search term 'backup microcontroller'

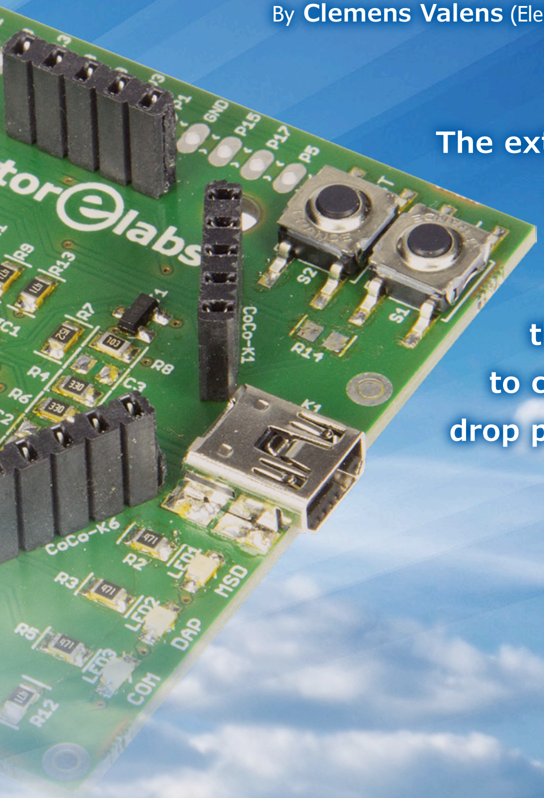
[4] www.elektor.com, search term '150455'


elektor labs
www.elektor-labs.com
E682017
LED17
SWD
I2C
ISP
E773167
CoCo-K5

Elektor mbed interface

CoCo-ri-Co in the cloud

By **Clemens Valens** (Elektor Labs)



The extended open source library together with free online code editor and compiler suite made available by ARM (or shouid we say Softbank?) as the fast prototyping platform *mbed* is an excellent tool for developing powerful applications on third-party hardware. But did you know that it is also possible to create your own mbed enabled hardware and add drag 'n' drop programming to your design?

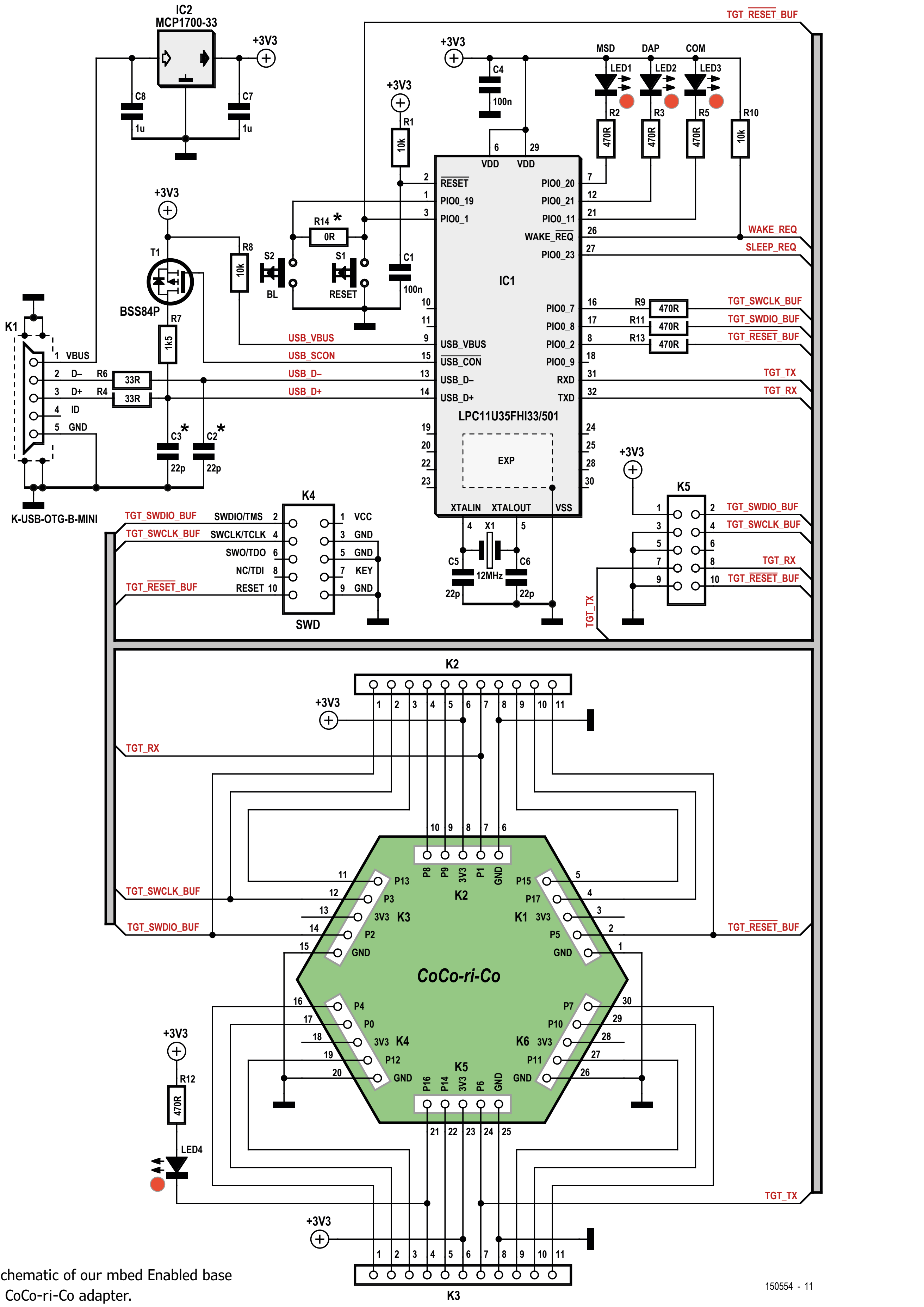


Figure 1. Schematic of our mbed Enabled base board with CoCo-ri-Co adapter.

Since we first published about mbed in September 2010 [1], the platform has evolved considerably. At the time mbed was only partly open source and there existed just one supported microcontroller board, the mbed LPC2368. That one was soon to be replaced by the mbed LPC1768, a board that quickly rose to stardom, thanks to its low price (brpi, ie before Raspberry Pi) and its small size.
Fast Forward 2016. The mbed website [2] now proudly lists over one hundred mbed compatible boards (platforms) produced by more than 25 manufacturers targeting nine different microcontroller families (targets).

With a clear ambition to grow, mbed (see **inset**) provides tools and instructions for those who want to create mbed compatible hard- and software. However, unlike your typical open-source movement, mbed wants to keep tight control over its quality. You are free to create your own mbed compatible board, sell it and do what you want with it, except for calling it mbed enabled and using the mbed logos. If you want to do that, you must go through an acceptance procedure to prove that your product has what it takes to be called mbed enabled. If you do so, and your product passes, it becomes a new platform on the mbed website where other people can select it and develop programs for it. The acceptance procedure is free; all you have to do is provide ten working pieces of your hardware to the mbed team. And, of course, invest time and effort in learning the tools that determine if your board will pass or fail.

Because mbed is getting more and more traction we decided to design an mbed compatible board and submit it to the mbed team in order to obtain the official mbed enabled label. One word of caution is in place though. Before you start doing exactly like we did, please know that the tools and documentation were evolving quickly during our application period and they may still be evolving, so what we did do and what we did not understand may now be obsolete and/or different. Always refer to the mbed website for the latest details on the hardware development kit and the acceptance procedure [3].

After this long introduction I will come to the point, or at least I will try so. Let's start by summarizing the procedure:

- choose target;
- create platform;
- port mbed software to platform;
- run platform tests;
- ship platform to mbed;
- make mbed team accept platform;
- create webpage;
- create example code;
- go public.

Note that the terms "target" and "platform" are used as they are used by mbed, i.e. the target is the microcontroller that will execute your application program; the platform is the board on which the target is mounted.

1. Choose a target

This is the easy part if you are not a microcontroller manufacturer, because you can just pick one from the targets that are already supported by mbed. It is also possible to add a new target to mbed as long as it is an ARM-based controller, but you will have to do a lot of programming, porting and testing in close collaboration with the mbed team. We went the easy route, we chose an existing target, the LPC812. We did not pick this one accidentally, but because we already had an Elektor board for it, the CoCo-ri-Co board [4] to be exact. Without doing anything it is possible to use the mbed tools to create programs for this board, but it lacks the cool drag and drop programming capabilities that mbed enabled boards offer.

2. Create a platform

The CoCo-ri-Co board in itself is an mbed compatible platform, but it is not mbed enabled. To make it mbed enabled it is necessary to add the mbed interface to it. This interface is another ARM-based controller that runs special firmware to allow it to program the target simply by dragging an executable file on a USB drive. Target programming is done through the Serial Wire Debug (SWD) interface that most recent ARM controllers have.
When the mbed interface is connected to a computer it is detected as a pen drive. It is also detected as a serial port (if the driver is installed), which is really handy for debugging. By bringing it up in a special mode it can program itself, making firmware updates very simple.

The Hardware Development Kit (HDK) [5] contains the information you need to get started. There are two options for the mbed interface:

- Use one of the available interfaces
- Create a new mbed interface

As you will understand, option 2 is much more work than option 1 and so we opted for the latter. When we started this project option 1 was even simpler than we could hope for because only one existing mbed interface came with all the files needed to program itself and our LPC812 target and so we settled for the LPC11U35. This controller comes in three versions – FB, FE and FH – of which we tried two (FB and FH, not FE because it is a BGA type) to find that only the FH version (QFN package) worked.

The circuit that we put around the mbed interface is shown in **Figure 1**, which is an almost exact copy of the circuit published by the mbed team for this interface. All we did was adding connectors for our target to it and some more connectors to enhance connectivity of the board. The result is a sandwich consisting of an mbed interface board with the CoCo-ri-Co

mbed, IoT & The Cloud

Originally mbed was a free, easy-to-use online compiler with libraries to develop applications for hardware with ARM-based microcontrollers without installing any tools. However, with the introduction of mbed 3.0 a branch was added to the platform repurposing mbed to the Internet of Things or IoT and cloud-based services.
From the website: *The ARM mbed IoT Device Platform provides the operating system, cloud services, tools and developer ecosystem to make the creation and deployment of commercial, standards-based IoT solutions possible at scale.* The original mbed ("developer") got more or less hidden away. Not so long ago mbed OS 5.0 was launched as *a platform operating system designed for the internet of things*. The "old" mbed is now called mbed OS 2.0. Version 3.0 exists too, but 4.0 was skipped for some reason. www.mbed.com

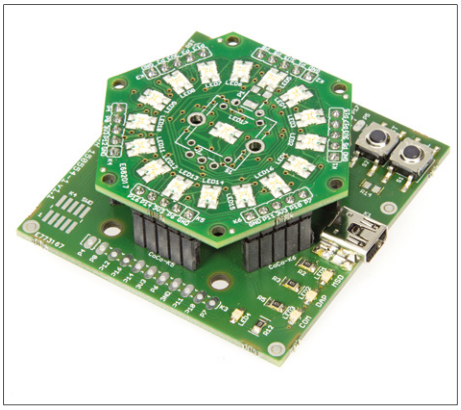


Figure 2. The Elektor mbed Enabled CoCo-ri-Co sandwich assembled and ready for use.

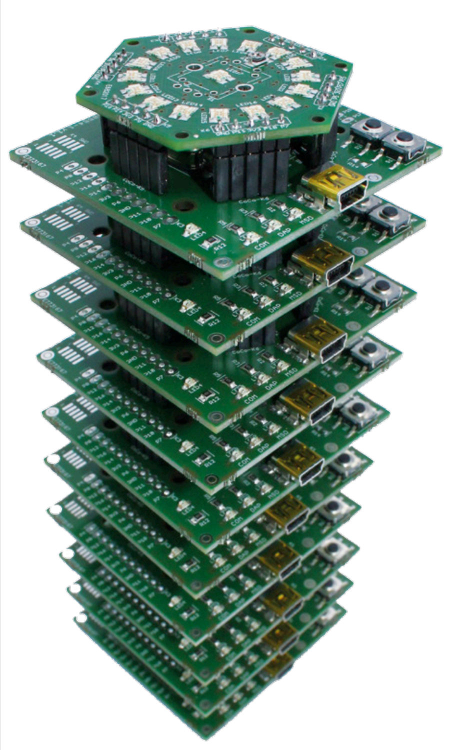


Figure 3. We send this stack of ten systems to ARM for approval.

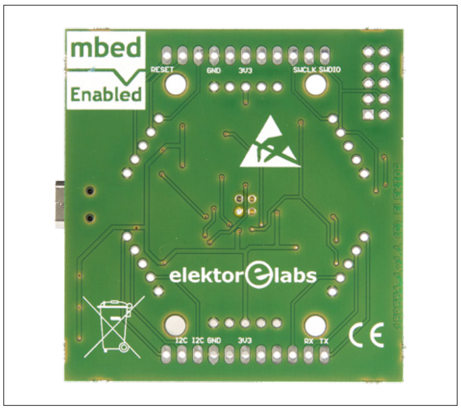


Figure 4. Our board proudly sports the mbed Enabled logo.

board plugged on top of it (**Figure 2**). Connectors K2 and K3 give access to the CoCo-ri-Co board's I/O connectors. K4 is a small-pitched SWD connector that can be used to connect the base board to another target. K5 is a similar connector but with a traditional 0.1" pitch. It also has the serial port signals on it.

If you want to use K4 to program a target that is not an LPC812, you must adapt the mbed interface firmware for it. We did not try this, but it involves obtaining programming routines for your target from the mbed team.

3. Port mbed interface software to platform

Lazy as we are, we skipped this step. We could, because we used precompiled binaries suitable for our LPC812 target to program our mbed interface IC1.

4. Run platform tests

This is a step that requires careful reading of the documentation [6]. Here we discover that since we did this project things have changed: the GitHub repository has changed names [7]. You will need this repository because it contains the test suite, in the tools folder. Follow the documentation carefully to create the scripts for testing your MUT (Mbed Under Test). It is a big-ass test suite written in Python capable of testing several MUTs in one go, but since we only have one board our scripts can be limited to a few lines. Supposing that you have Python installed on your PC, you must first install the tools with the `setup.py` script in the main folder:

```
setup.py install
```

Run a quick test:

```
cd tools
build.py -S >> platforms.txt
```

If this worked you can now find in the file `platforms.txt` the MCU names needed by `muts_all.json` script which in turn is needed by `singletest.py` script (see [4]). Check the file `settings.py` and edit (or create) a file `private_settings.py` to configure the path to the compiler that will compile the test programs that will be run on our CoCo-ri-Co target. Connect the board and run `singletest.py` to create a test report:

```
singletest.py -i test_spec.json -M
   muts_all.json >> test_results.
   txt
```

This can be quite long; it took about 7 minutes to complete on my PC. When ready the report contains a table of all the tests that were done and its results. Results can be OK or TIMEOUT. Some tests will fail due to missing hardware, the Real-Time Clock (RTC) test for instance because the CoCo-ri-Co board does not have an RTC.

Important: Make sur the SWD port and RESET pin are not disabled in LPC812 firmware! They are disabled in the firmware that is programmed by default in our CoCo-ri-Co boards. Use Flash Magic or another LPC812 programming tool to erase the flash completely before testing.

If a test fails when it shouldn't, you can run it again without doing all the other tests. For instance, the command

```
singletest.py -i test_spec.json -M
   muts_all.json -n MBED_10
```

will only run the test named MBED_10. Keep the test results, because the mbed team will want to see them.

5. Ship ten pieces to mbed

If you hadn't done so already, it is now time to contact the mbed team to inform them about your plans [3]. They will direct you to a form to fill in and give you an address to ship ten pieces of your hardware to (**Figure 3**). Make sure they have all passed the platform tests.
When the mbed team has tested and accepted your hardware (this will probably take several weeks) you will receive an email saying so. Don't make the mistake of thinking that you are almost done, because you are not. All you have obtained now, after signing and returning a contract, is that you are now allowed to use the mbed Enabled (without a hyphen and note the capital 'E') logo (**Figure 4**).

6. Make the mbed team accept your platform

This is a tough one and I sincerely hope things have become a bit easier since I went through the procedure. There are two things to accomplish here:

- integration of your platform into the mbed repository;

Mbed quick start

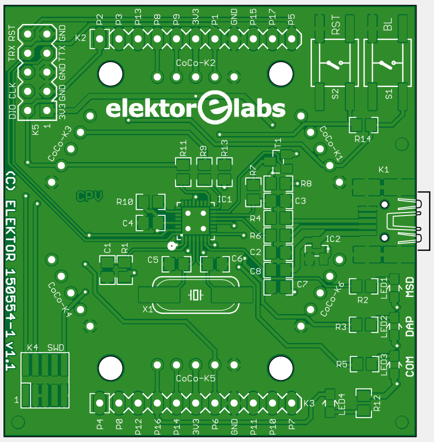
Connect the mbed Enabled board to a free USB port on your computer. The OS will detect an mbed Composite Device and start to install drivers for it. On Windows you may have to install a serial port driver first, it is called mbedWinSerial_xxxxx (where xxxxx is a number, at the time of writing it was 16466) and can be found on the mbed website. When the installation of the board was successful you should now have a mass storage device, probably named DAPLINK, and an mbed Serial Port. Open the mass storage device and double click on the file mbed.htm.

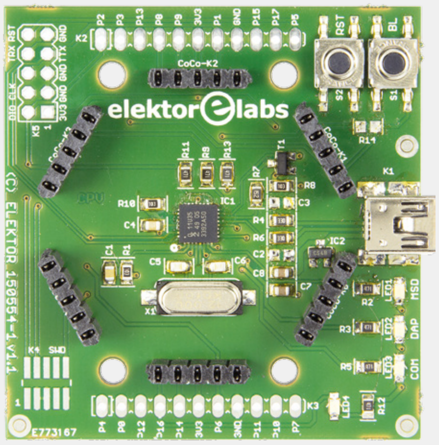
This will launch the web browser which in turn opens the mbed login page where you can login or create an account (free). Once connected you will be redirected to the mbed board's platform page. (If you were already connected the login step is skipped.) In the case of the board described in this article together with CoCo-ri-Co you will end up on [8]. On this page you will find everything you need to know about the platform. After reading this information carefully you can click on the Compiler button at the top of the page to open the workspace manager where you can create programs. This works very much like a traditional offline compiler except that when a program is successfully compiled the executable will be downloaded to your computer. Copy this file on the mbed mass storage device and reset the board to start your program (some boards like the BBC micro:bit automatically reset). That's it, easy-peasy, nothing to setup, no extra hardware needed.

- modifying the mbed interface code so that it points to your platform page on the mbed website.

Step A means adding files to the mbed repository on GitHub [7]. The objective here is to get your target added to the `hal/targets/hal` branch of the repository. You will now be in contact with software engineers that do this on a daily basis and who tend to communicate in riddles and acronyms and they assume that you are a Git expert. I am not and I still do not know what they were on about, but by playing dumb I managed them to do some of the hard work for me. If you look deep enough, you will discover the folder `hal/targets/hal/TARGET_NXP/TAR-`

Component List



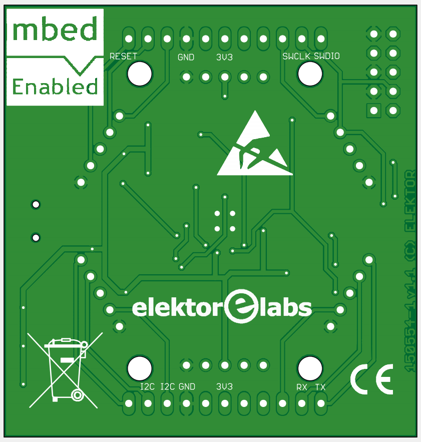


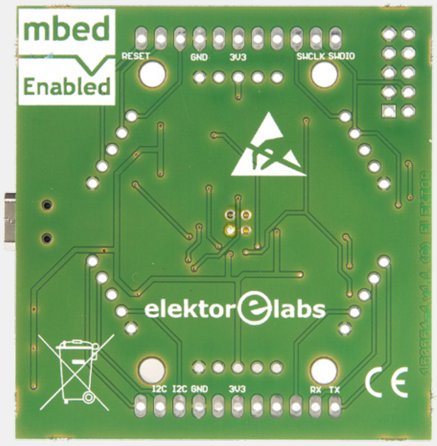
Resistors (5%, 0.1 W, 0805)
R1,R8,R10 = 10kΩ
R2,R3,R5,R9,R11,R12,R13 = 470Ω
R4,R6 = 33Ω
R7 = 1.5kΩ
R14 = 0Ω (*do not mount)

Capacitors (0805)
C2,C3 = 22pF
C5,C6 = 22pF (*mount when observing EMC problems)
C1,C4 = 100nF
C7,C8 = 1µF

Semiconductors
T1 = BSS84P
IC1 = LPC11U35FHI33
IC2 = MCP1700-33
LED1-LED4 = LED, red

Miscellaneous
X1 = 12MHz quartz crystal
S1, S2 = tactile feedbacjk switch, 6x6 mm
K4 = 10-pin pinheader (2x5), 0.05'' pitch, SMD
K2, K3 = 11-pin pinheader, 0.1'' pitch
K1 = USB-B mini connector
CoCo-K1, CoCo-K2, CoCo-K3, CoCo-K4, CoCo-K5, CoCo-K6 = pinheader socket, 1x5 contacts, 0.1'' pitch
PCB # 150554-1 v1.1





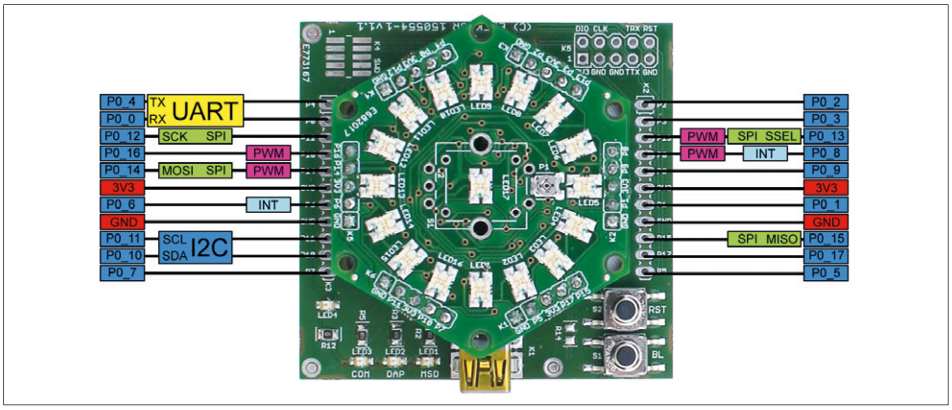


Figure 5. The pinout of our platform, mbed style.

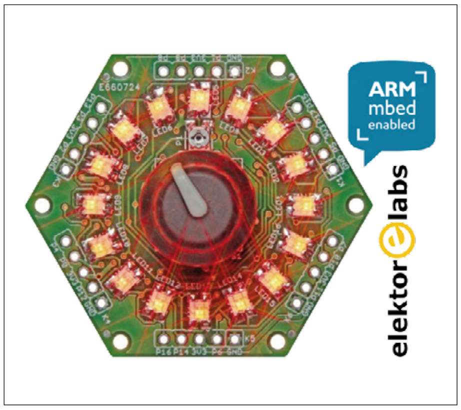
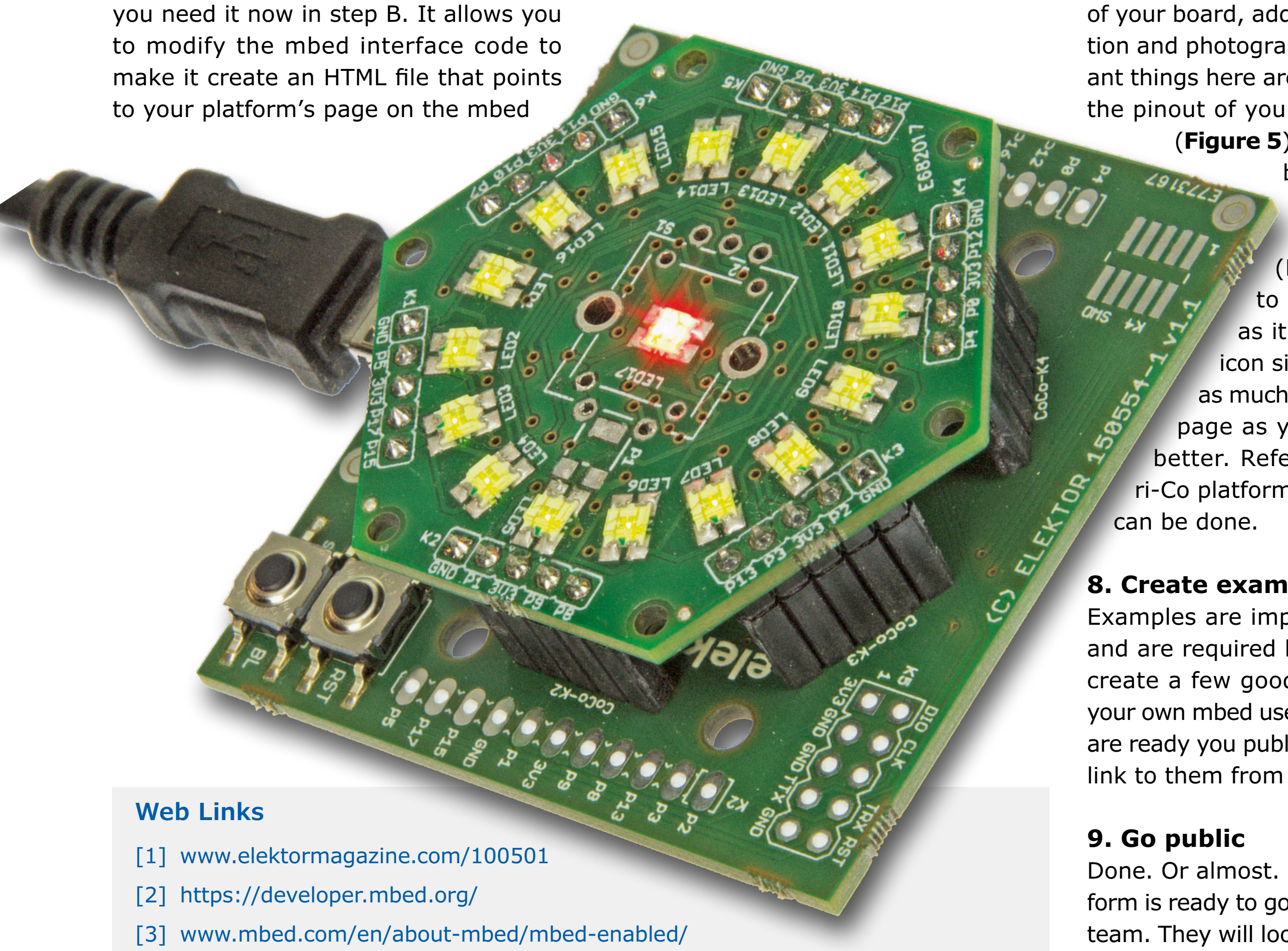


Figure 6. The photo used in the online compiler to reference our platform.

`GET_LPC81X/TARGET_ELEKTOR_COCORICO`, proof that we made it.

At some point you will have received a unique mbed ID for your platform and you need it now in step B. It allows you to modify the mbed interface code to make it create an HTML file that points to your platform's page on the mbed website. The user sees this file when the mbed Enabled board gets connected to the computer. A template for this page will be created for you by the mbed team.



7. Create webpage

If you get this far you are almost there. If you did everything right, this is a quick exercise. You must write the specifications of your board, add other useful information and photographs. The most important things here are the drawing showing the pinout of your board in mbed style (**Figure 5**) and the photo that will be used in the online compiler to represent your platform (**Figure 6**). Make sure to provide a clear photo as it will be shrunk to tiny icon size on most pages. Put as much documentation on this page as you can, the more the better. Refer to the mbed CoCo-ri-Co platform page [8] to see what can be done.

8. Create example code

Examples are important for new users, and are required by the mbed team, so create a few good ones. You do this in your own mbed user space and when they are ready you publish them. You can then link to them from your platform page.

9. Go public

Done. Or almost. If you think your platform is ready to go live, contact the mbed team. They will look your work over once again and if they are happy with it, they will make your platform page public. Congratulations! You have successfully created an mbed Enabled platform. ◀

(150554)

Web Links

[1] www.elektormagazine.com/100501

[2] https://developer.mbed.org/

[3] www.mbed.com/en/about-mbed/mbed-enabled/

[4] www.elektormagazine.com/140183

[5] https://developer.mbed.org/handbook/mbed-HDK

[6] https://developer.mbed.org/teams/SDK-Development/wiki/Mbed-SDK-automated-test-suite

[7] https://github.com/ARMmbed/mbed-os

[8] https://developer.mbed.org/platforms/CoCo-ri-Co/

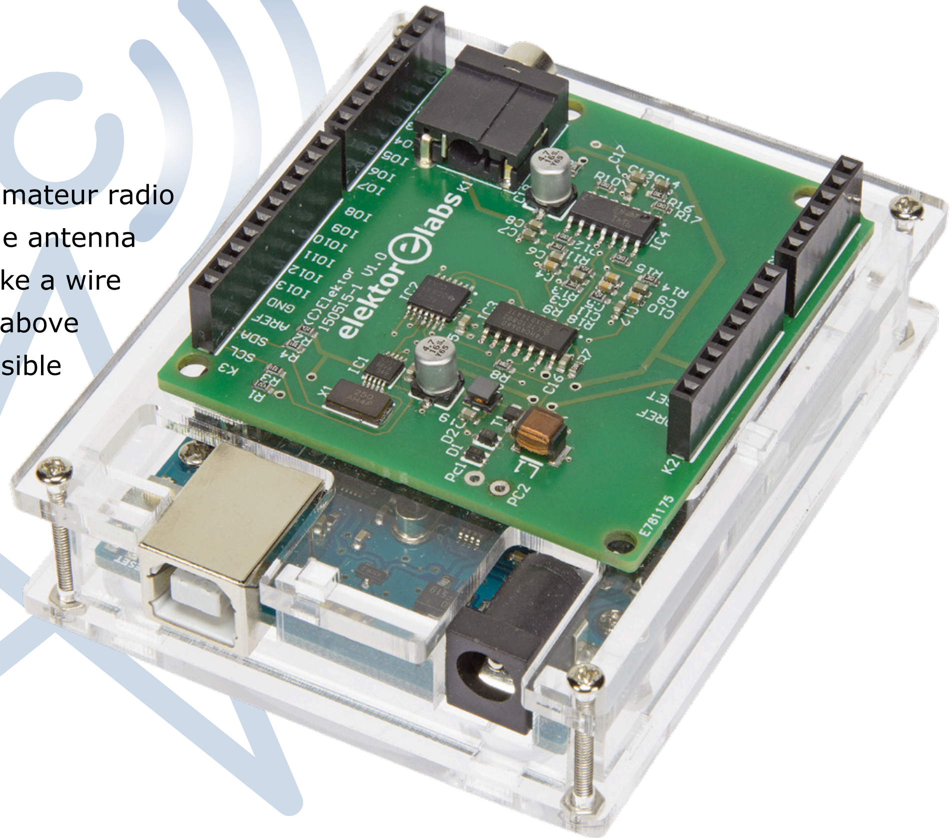
[9] www.elektormagazine.com/150554/

Elektor SDR Reloaded (3)

Antennas for the SDR Shield

By **Burkhard Kainka** (Germany)

Radio reception on AM, shortwave and amateur radio bands stands or falls by the quality of the antenna used. Good results are easy: you just take a wire 30 meters long and string it ten meters above ground between two trees, as far as possible away from your house. In reality this is not always achievable. So we need to find the ideal compromise: an antenna that is effective, unobtrusive — and small.



As experience shows conclusively, this requires pulling in not just maximum signal voltage but (even more so) attaining the best signal-to-noise ratio. Interference is pervasive and its amount grows daily. Living quarters in particular abound with sources of interference. Computers, power supplies, consumer electronics, motors and much more besides generate a dense layer of electronic noise that overwhelms much of what goes on in the amplitude modulation (AM radio) region.

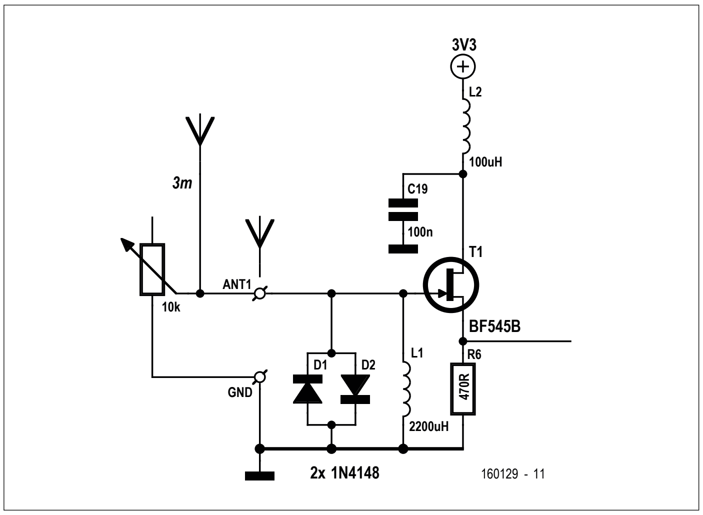


Figure 1. Indoor antenna with attenuator.

Aerial attenuator

On the SDR Shield we already have a FET impedance converter (T1) with an RF choke (L1) and diode limiter (D1/D2). The simplest, yet adequate, antenna for the SDR is a piece of wire about three meters long, which is easy to sling over a cupboard or even on the floor of the room. Using just this you can already hear a dense array of radio stations on the short waves in the evening. Some signals on the 75-meter band may even be strong enough to overload the receiver. A helpful remedy is an attenuator in the form of a 'pot' (potentiometer) that is simply connected in parallel with the antenna input (**Figure 1**).

As soon as you connect the antenna wire, the noise level increases discernibly. The noise floor level varies from band to band and can swamp weak signals with ease. The antenna itself is not only the source; chiefly it arises from the 'polluted' ground or earth wire. Normally your PC is connected to earth via a protective ground wire. But between the actual earth and the power outlet (mains socket) on the wall, it picks up a whole load of 'crap' along the way. Then we have the USB cable carrying all manner of wideband signals, meaning that the noise level at the Arduino's ground connection is even greater. Even a perfectly ideal outdoor antenna cannot solve this problem, because the GND interference and the antenna signal are stacked up at the input. The best a good antenna can do is deliver signals that are stronger than those producing the background noise.

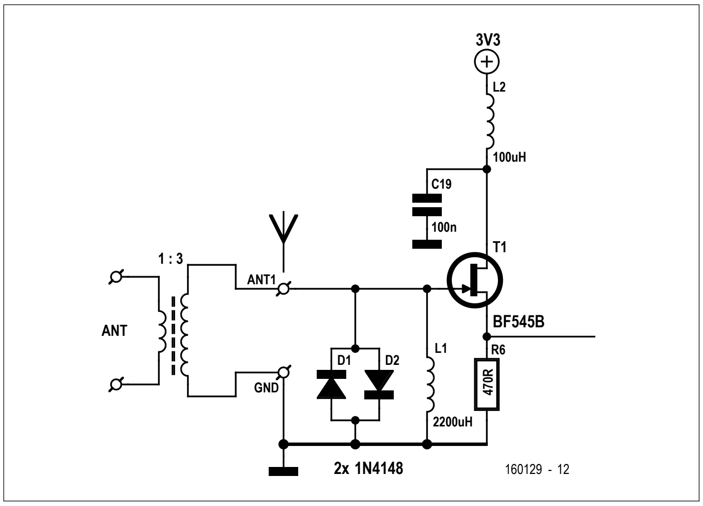


Figure 2. Using an RF transducer.

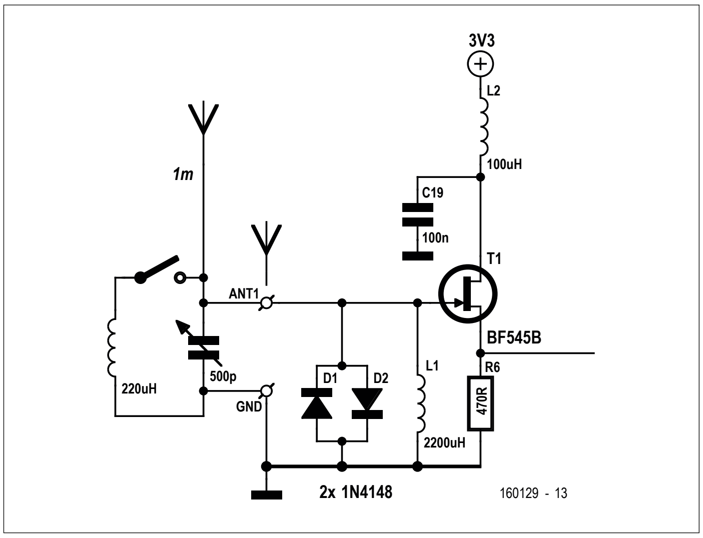


Figure 3: Tuned input circuit.

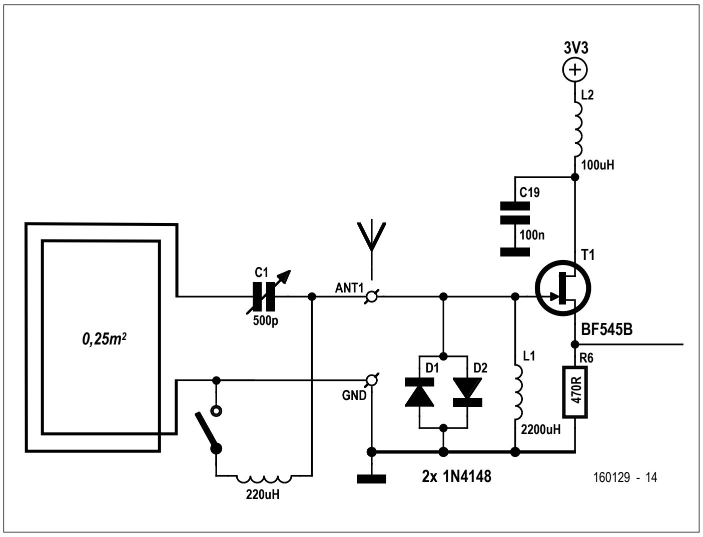


Figure 4: Frame antenna for the long and medium waves.

Isolating transformer

Let's assume you have a perfect outdoor antenna fed with coaxial cable or an aerial setup with an AM antenna on the roof. You always still need to ensure that interference is not being introduced by the ground connection. The best way to do this is with an RF isolating transformer (Figure 2). You can make one of these using a ring core or a double-hole core. Guide values for winding these are two turns on the antenna side and six turns on the SDR side.

Resonant circuits

Signals on the long and medium waves are improved considerably when you employ a resonant circuit (**Figure 3**). In the state of resonance a tuned circuit will simultaneously raise the signal voltage and degrade undesired signals occurring at multiples of the target frequency. With a variable capacitor of 500 pF the internal choke coil (2.2 mH) can tune down to the lower end of the long-wave broadcasting band at 150 kHz. A short antenna just one meter long will suffice now. Admittedly the circuit is relatively narrowband, meaning that you have to retune it each time you change frequency.

For the medium waveband from 500 kHz you simply hook up a coil or fixed inductance of 220 µH in parallel. A medium wave coil can also be wound on a ferrite rod, removing the need for a wire antenna. In the process it becomes plain that the signal-to-noise ratio is improved noticeably. This is because the ferrite rod picks up predominantly the *magnetic* component of radio waves, whereas near-field interference signals occur mainly as alternating *electrical* fields.

In reception tests it is always valuable to make comparisons using other radios. It might turn out for example then that a fairly basic transistor radio (battery operated) might receive weak medium wave signals more clearly than the SDR. That's predictable really: this simple radio uses a ferrite antenna, so it cannot pick up interference related to a ground connection. So we need to do a bit more to make the SDR equally interference-resistant.

A different subject for making comparative tests is a mini stereo system with AM coverage. These tend to use a small frame antenna with multiple turns of wire, connected by a fairly long cable. The instructions tell you to set up this antenna at a suitable spot away from the hi-fi system. The digital modules in the system produce so much interference that using an internal ferrite antenna simply would not work. Tests indicate that adequate distance makes all the difference. You need at least half a meter or so of separation to achieve good reception. On VHF these problems are far less pronounced of course, because the majority of interference occurs on the lower frequencies.

Frame antenna

A good frame antenna could be the solution. And a coil with a physically greater capture area will produce a higher voltage in the antenna. The antenna loop used should have two turns and a frame length of 0.5 m (**Figure 4**). For this we need four meters of wire or better still, eight meters, so that the antenna can be placed two meters away from the computer. Used like this, you have a very usable wideband short wave antenna. But on the long and medium waves the inductance is too low.

You can cure this with a variable capacitor connected in series. The loop antenna now becomes a component of the input circuit and can be brought into resonance on long wave. Using a smaller inductance of 220 µH makes it right for medium wave.

Rod antenna

The familiar car aerial is proof that rod antennas work well. Finding yourself in a parking area away from general habitation one evening and tuning the medium waves on the car radio, you may be amazed how good the reception can be out there. Using this kind of antenna, finding the right location is your only challenge. The best spot would be on the roof or on an aerial mast as far as possible away from the house. A compromise might be an antenna placed at least outside the window or, if necessary, fixed on the inside of the pane.

You will certainly need a fair length of screened cable. And this will work then only with a matching preamplifier or impedance converter (**Figure 5**). In the simplest case a JFET will suffice both as impedance converter and preamplifier, which can be powered remotely over the cable. The Arduino produces the appropriate voltage of 3.3 V, which is filtered in addition. To make the antenna usable in all situations we will need to add a pot as attenuator and a simple low-pass filter.

An ideal installation of this antenna will deliver a fully adequate signal voltage on all bands. Admittedly one problem remains to be solved: wideband interference will be picked up, via the ground conductor, from the computer and other gadgets connected to the AC supply.

Loop antenna

All these tests have proved that (1) magnetic antennas give the best results indoors, (2) they need to be placed as far as possible away from the computer and (3) the best antenna location in the home is next to an external wall or a window. All of these insights point in the direction of a magnetic loop with a preamplifier, powered remotely over a coaxial cable (**Figure 6**).

The actual loop consists of four meters of coax cable, fixed to a window in a square that measures one meter along each edge. The inner conductor of the cable serves as the loop. The braiding of the cable is connected to ground and shields the antenna from electrical fields. At the upper end we need to open and break the screening so that it does not become a closed circuit in itself.

The preamplifier employs a BC547B audio transistor, the amplification of which extends up to higher frequencies. The low input impedance of the transistor represents practically a short circuit to the antenna feeder, which makes the antenna extremely wideband. The preamplifier is connected and powered using a coaxial cable of any length you choose. Extracting the signal, with the attenuator and optional lowpass filter, remains unaltered.

In actual fact there is plenty in this simple project that might be worthy of improvement. The impedance is not an exact match to the cable and improvements could still be made in matters like linearity and overload rejection. Both aspects call for greater collector current. But the amplifier stands out by its simplicity and works very satisfactorily.

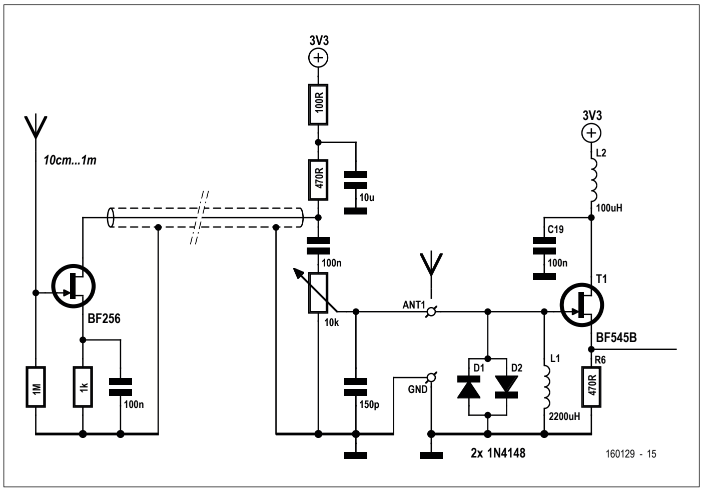


Figure 5: Rod antenna with preamplifier.

> A Magnetic Loop antenna gives very good results.

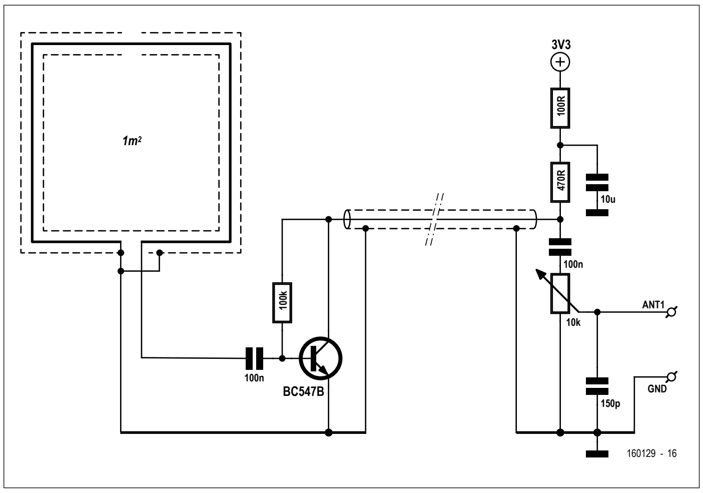


Figure 6: Wideband Loop with preamplifier.

This antenna provides very good results on all frequencies from 150 kHz to 30 MHz. Excellent signal-to-noise ratio is achieved on long wave even during the daytime. Evaluation tests with other antennas have demonstrated best results with the loop almost every time. For comparison an out-of-use wire antenna that the author had set up outdoors was used. At an elevation of about 30 feet this antenna certainly produced great signal voltages, simultaneously, however, with a broad spectrum of interference. For that reason reception using the screened loop gave the better results in most cases. ◄

(160129)

Internet Radio with Fluorescent Display (1)

Modern technology with a traditional look

A fine thing, a kitchen radio, with its CD player, clock and even a timer to make sure you don't burn your dinner, all in one compact unit. So, when the author's unit gave up the ghost a replacement had to be found. That display was a particularly neat feature: not too big, but with everything necessary on it. Perhaps it could be recycled into a new homebrew radio?

By **Michael Busser** (Germany)

A pretty good idea, at least by this engineer's standards. But, before turning a concept into reality, there was of course some (Internet-based) research to be done. Sadly, no sooner had I managed to bring together all the necessary information about the complexities of the display module and the perils of driving it, than an unfortunate accident turned my VFD (vacuum fluorescent display) into a mere FD, the V having been irreversibly dissipated into the atmosphere.

Fortunately similar displays are widely available on the surplus market, and so a 'new old' device can be had cheaply. I managed to obtain a type FV651G VFD (see **Figure 1**): excellent, if not quite as beautiful as the original.

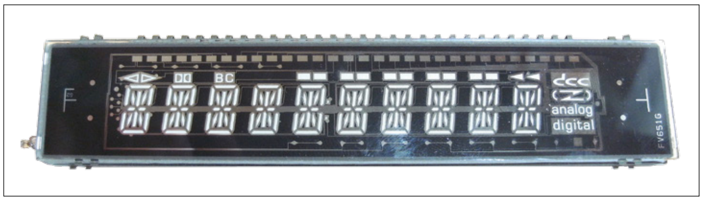


Figure 1. The FV651G VFD used in this project.

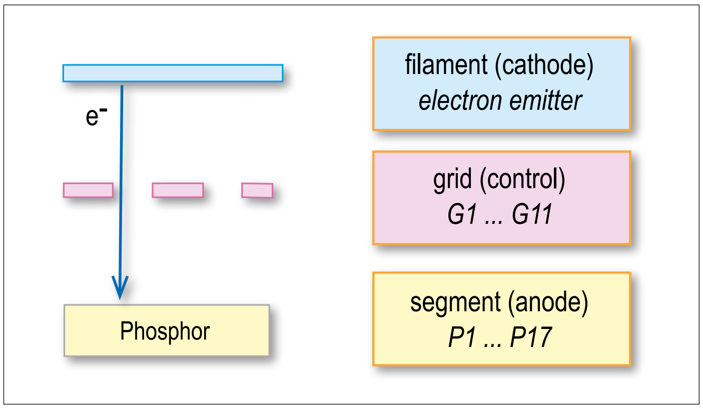


Figure 2. Structure of a VFD. The phosphor-coated anodes light up when they are hit by electrons.

How it works: the VFD

The principle of operation of each segment of a VFD is essentially the same as that of a (triode) tube or valve: see **Figure 2**. Details of its construction can be found at [1] and [2], but in simple terms it works as follows. As with all tubes there is a coated heater filament, which is the cathode of the device and which emits electrons. Each of the individual segments of the display (the anodes) has a phosphor coating: if electrons hit the phosphor it glows. Meanwhile, the voltage on the grid allows the flow of electrons to be controlled. A negative potential prevents the illumination of all the segments (in this case up to 17 of them) in a character position. With a positive potential on the grid the segments of the digits are allowed to be lit, but which segments do in fact light depends on their individual anode potentials. A negative anode potential repels the electrons from the segment, allowing them to be collected on the positive grid: the segment will not light. However, if the anode potential is sufficiently positive, the electrons will hit the phosphor and the segment will light. The grid connections G1 to G11 determine which character position is enabled, while the anode connections P1 to P17 control the individual segments within each character.

According to its datasheet the type FV651G display we use here requires an AC supply of 3.7 V for its heater, while the grid voltage is specified at 32 V. The latter of these requirements makes our driver circuit somewhat more complex. The display has eleven character positions, the one at the far right comprising special annunciators required for its original application. In each character position there are up to seventeen individually-controllable elements; the seventeenth segment, the colon, is only present in one position.

Corresponding segments are all connected together, which means that multiplexed operation is required. The display can conveniently be mapped in software to an array with eleven columns and seventeen rows.

Heater

In the defunct kitchen radio the heaters were powered from the mains transformer using a winding with a center tap. However, the plan is for the new radio design to run from an ordinary

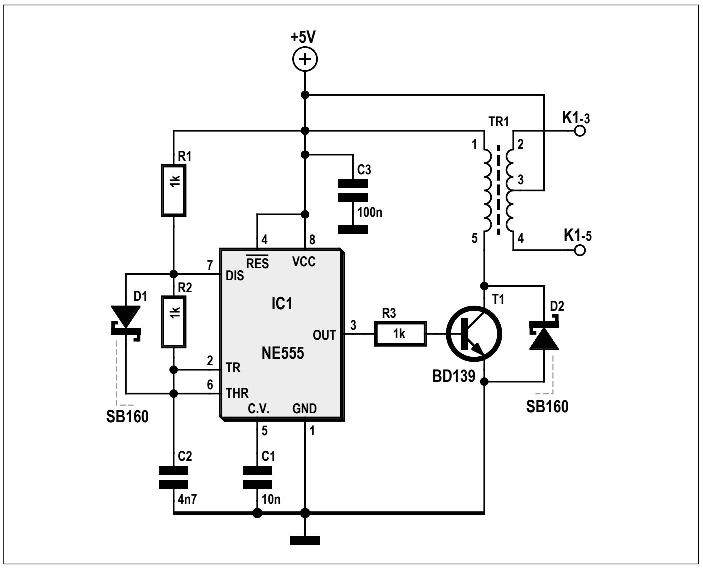


Figure 3. This auxiliary circuit by Kerry D Wong is used to supply the heater.

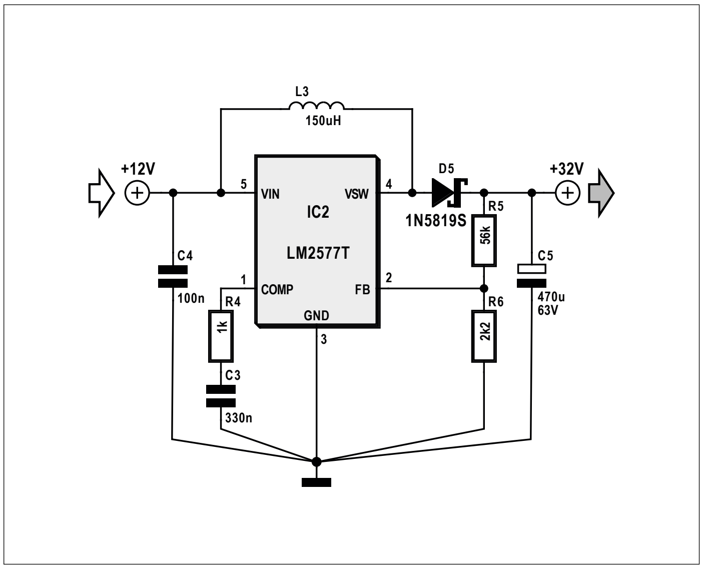


Figure 4. Auxiliary circuit to create the anode supply using a step-up converter.

12 V mains adapter, and so we must look for an alternative approach. Note that it is not essential for the AC to have a frequency of exactly 50 Hz: **Figure 3** shows our solution using a good old NE555 timer and a custom-wound transformer.

A modification from the standard circuit is the addition of D1, which makes the output squarewave almost symmetrical. The transformer is wound on an N27 ferrite core with an external diameter of 16 mm. The primary side consists of 40 turns of enameled copper wire and the secondary side comprises two windings of 18 turns each. The result should be a voltage of just under 2 V_{RMS} across each secondary winding under load. The original circuit was designed by Kerry D Wong [3]. The drive voltage for the heater should not be centered around ground potential, as this can cause 'ghosting' (weak illumination of undriven segments): this is why the center tap of the transformer is tied to +5 V.

Anode voltage

The anode voltage is also generated from the 12 V output of the mains adapter, using a step-up converter: see **Figure 4**. The only change from the basic circuit given in the LM2577T datasheet is the value of the resistors.

According to the manufacturer the output voltage is given by the formula:

$$V_{OUT} = 1.23\ \mathrm{V} \times (1 + R5 / R6)$$

With the values of R5 and R6 shown the anode voltage will be 32.5 V, as required.

Grid drive

In P1 to P17 and G1 to G11 we have a total of 28 display electrodes that have to be driven. The high anode voltage of 32 V means that we will have to use what is known as a 'high side driver' circuit. A suitable device for this application is the UDN2981. This IC contains eight drivers, and so we need a total of four chips. On the input side, these devices just need a regular 5 V drive signal.

The ATmega32 we are using in this project has only a limited number of I/O pins, and in order to avoid sacrificing almost all of them to driving electrodes an eight-bit latch is wired before each driver IC. The four latches are driven in parallel, and one latch is selected using its LE ('latch enable') signal. This reduces the number of outputs required to drive the display to twelve (eight data bits and four LE signals). The microcontroller can also check the status of a couple of user controls. The complete circuit diagram for the display driver is available as the document *Muxer_Schaltplan.pdf* in the zip archive *Muxer.zip* (see the download at [4]). The corresponding Eagle design files are also included in the archive.

ATmega firmware

From the point of view of the microcontroller the display is represented by a 32-bit value divided into four bytes, one for each latch. The main job of the microcontroller is to output these

Table 1. Meaning of the 32 bits

Byte	3								2								1								0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Electrode	G11 to G01											P1 to P17																	unused			
Function	10	9	8	7	6	5	4	3	2	1	0			a	b	f	k	j	h	m	g	n	p	r	c	e	d					
	Character position (grid)											Segments (anode)																				

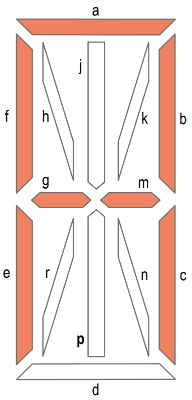


Figure 5. These segments illuminate when the bits shown in red in Table 2 are set to ‘1’.

Table 2. Driving the segments shown in Figure 5

Bit	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Electrode	P1 to P17																
Segments				a	b	f	k	j	h	m	g	n	p	r	c	e	d

four bytes to an eight-bit port at the multiplex frequency, and supply the right control signals to the latches. **Table 1** shows the meanings of the individual bits in the 32-bit word. The four least-significant bits are not used. The definitions can be found in the files *fv651.c* and *fv651.h* in the download.

The data structure *TDispControl* brings together the variables needed to drive the display:

```
typedef struct {
  uint32_t data;
  uint8_tgrid;
  uint16_t anode;
} TDispControl;
```

The variable *data* contains the 32-bit pattern for the four latches; *grid* is a counter indicating the current display character position (from 0 to *MAX_DIGITS*). The variable *anode* contains the bit pattern of the 17 segments. The routine *flush_VFD()* shown in **Listing 1** transfers the contents of *data* to the four latches.

Listing 1. Processing the 32-bit word.

```
void flush_VFD() {
    uint8_t data;

    data = (ctrl.data & 0xFF);           //1st byte
    PORT_VFD_DATA = data;
    LE_VFD_0_ON;                         //impulse with nop =
    asm volatile („nop“);                //760ns
    LE_VFD_0_OFF;                        //measureds with scope

    data = ((ctrl.data >> 8) & 0xFF);    //2nd byte
    PORT_VFD_DATA = data;                //delay between 1st and 2nd impulse =
    LE_VFD_1_ON;                         //2us
    asm volatile („nop“);
    LE_VFD_1_OFF;

    data = ((ctrl.data >> 16) & 0xFF);   //3rd byte
    PORT_VFD_DATA = data;                //delay between 1st and 3rd impulse =
    LE_VFD_2_ON;                         //4.7us
    asm volatile („nop“);
    LE_VFD_2_OFF;

    data = ((ctrl.data >> 24) & 0xFF);   //4th byte
    PORT_VFD_DATA = data;                //delay between 1st and 4th impulse =
    LE_VFD_3_ON;                         //7.5us
    asm volatile („nop“);
    LE_VFD_3_OFF;
}
```

Listing 2. ASCII table and character range tests.

```
uint16_t lookupVFDSegTable(uint8_t index) {
    static const flash uint16_t lookupSeg[SEGMENT_LOOKUP_TABLE_SIZE] = {
        14343,  4100, 12483, 12485,  6340, 10437, 10439, 12292, 14535, 14532,  //0...9
        14534, 14535, 10243, 14343, 10435, 10434, 10375,  6342,   528,   536,  //A...J
         1584,  2051,  7430,  6438, 14343, 14530, 14375, 14562, 10437,  8720,  //K...T
         6151,  4388,  6190,  1320,   784,  8193,                              //U...Z
          103,  2247,   195,  4295,    83,     0,     0,  2130,     0,     0,  //a...j
            0,     0,     0,     0,   103,     0,     0,    66,     0,     0,  //k...t
            7,    36,    46,     0,    37,    73                               //u...z
    };
    if (inRange(LUT_0, LUT_9, index)) {return(lookupSeg[index - LUT_0]);}    //48...57
    if (inRange(LUT_A, LUT_Z, index)) {return(lookupSeg[(index - LUT_A) + LUT_A_OFFSET]);}  //65...90
    if (inRange(LUT_a, LUT_a, index)) {return(lookupSeg[(index - LUT_a) + LUT_a_OFFSET]); }  //97...12
    return (0);
}
```

Listing 3. Display firmware interface.

```
void init_VFD();                               //initialize all data structures
void mux_VFD();                                //displays next digit, timer controlled

void clearVFDDisplay(uint8_t withSymbols);     //clears display
    withSymbols = true                         //deletes all symbols
    withSymbols = false                        //lets symbols untouched

void clearVFDDisplayData(uint8_t index);       //deletes position „index“ in display
                                               //0 = most right digit

void setVFDDisplayData(uint8_t index, uint8_t data);  //shows „data“ on digit „index“

void setVFDDisplayText(uint8_t index, const char *data);  //shows string „*data“,
                                               //starts @ position „index“

void setVFDisplayInt(uint8_t index, uint8_t anzStellen, uint8_t wert);
                                               //shows number „wert“ whith „anzStellen“
                                               //digits starting from rigth

void clearRunText();

void setRunText( const char *data );

void tickRunText();
```

In sequence the four bytes are extracted from the 32-bit word and written to port A, which is connected to the latches' inputs. After each byte is written a pulse is generated on the appropriate latch enable signal so that the byte is stored in the latch. The length of this pulse, which is extended by the 'nop' instruction, is 760 ns. The execution time of the complete function is 9 µs at a clock rate of 16 MHz.

ASCII table

We are left with the question of how the bit pattern corresponding to a character arrives in the variable *data*: this is taken care of in the function *mux_VFD()*. Suppose that we want to display the letter 'A' as shown in **Figure 5**. The segments we want to light are a, b, c, e, f, g and m. What is needed is a look-up table that allows us to convert from a code for a displayable character to a bit pattern. **Table 2** shows the bit pattern for this example, which corresponds to the decimal value 14534. Of course, we need a 16-bit value in the table for each character we might want to display, and generating these values can be done conveniently using a spreadsheet. The table can also be found in the download, and can easily be modified to suit any particular application.

The table of 16-bit values is stored in the microcontroller's memory, and the character code serves as an index into the table. For reasons of economy not all possible codes are included in the table and so there are gaps in the coverage against which the character code must be tested. The ASCII table and the code to test the character value are shown in **Listing 2**.

If this code is called with the ASCII value for 'A' (65 decimal), the second of the three 'if' statements will be triggered and the value to be returned will be read from the table at offset 65 – LUT_A + LUT_A_OFFSET = 10.

Among the lower-case letters there is a number of zero entries in the table. An exercise for the reader: expand the table to that all the lower-case letters can be displayed.

What remains to be done is some bit-shifting so that the data bits read from the table occupy the right field in the 32-bit word. The structure *TDispData* includes a member called *digits* with space for *MAX_CHARS* characters and a bit field called *symbols* that contains a bit for each annunciator that can be displayed outside the main display area.

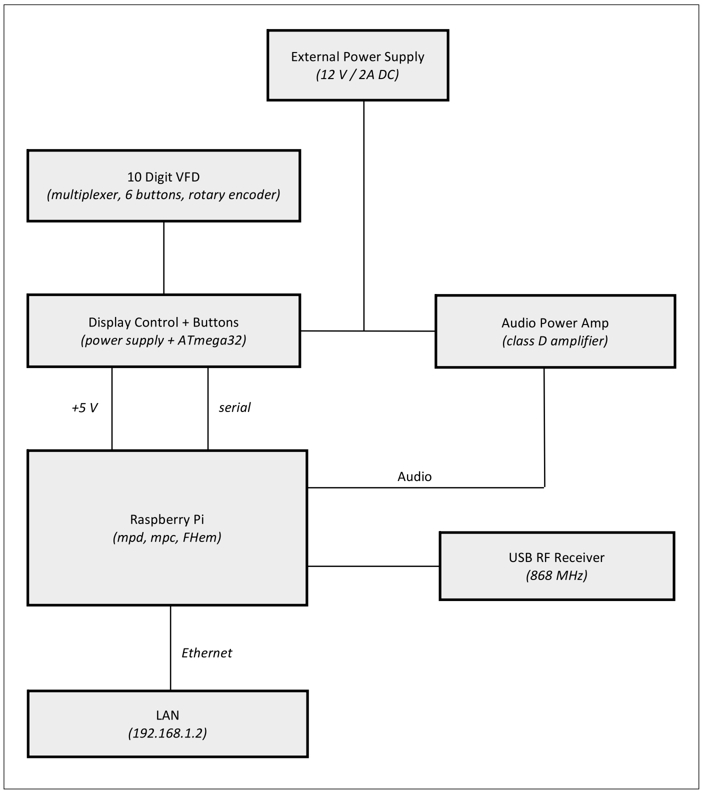


Figure 6. Block diagram of the complete kitchen radio.

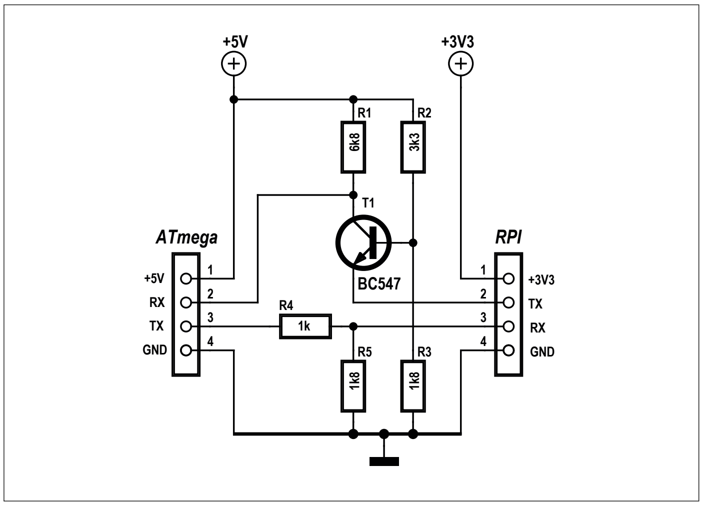


Figure 7. Level shifter for the serial interface, bridging between the 5 V world of the ATmega and the 3.3 V world of the Raspberry Pi.

```
typedef struct {
  uint8_tdigits[MAX_CHARS];  //one for each
                             //character position
  union TSymbolssymbols;     //annunciators on/off
} TDispData;
```

The function *mux_VFD()* is called every 5 ms under control of a timer. It reads the characters from *digits*, creates a bit pattern and writes the character to the display. This routine is rather more involved, and interested readers can find the commented code in the download.
The variable *display*, of type *TDispData*, together with the functions of **Listing 3**, provides the interface to the display module.

Character output

The following example command shows how to write the character 'A' to the fourth position of the display.

```
display.digits[4] = 65
```

The function call *setVFDDisplayData(4, 'A')* does the same thing, but is somewhat easier to read, and includes argument bounds checking. The functions to control the display are in the files *fv651.c* and *fv651.h* in the download. The complete source code is contained in the file *Software_1.zip* in the form of an Atmel Studio 6.2 project.
With this firmware we have complete control over the VFD, and our beautiful display (including some buttons for the user interface) is ready for the kitchen. Now we can turn our attention to the heart of the project: the Internet radio, based on a Raspberry Pi. That will be the main topic of the next part of this series.

Internet radio in outline

The block diagram in **Figure 6** gives an overview of the structure of the kitchen radio. The Raspberry Pi board is connected to the ATmega over a serial interface. Unfortunately, the Raspberry Pi runs at 3.3 V while the ATmega uses 5 V levels, and so we need a level-shifting circuit (**Figure 7**). In the transmit direction from the ATmega a simple voltage divider, formed by R4 and R5, suffices. From Raspberry Pi to ATmega the signal level needs to be increased. The solution here is very simple: the voltage divider formed by R2 and R3 holds the base of T1 at about 1.8 V, and the RX signal drives its emitter. If RX is high T1 turns off and its collector is pulled up to 5 V via resistor R1; if RX is low T1 turns on and a low level appears at its collector. Alternatively, the ATmega could be powered at 3.3 V: the level shifter could then be dispensed with, but the clock frequency of the microcontroller would have to be reduced to 12 MHz.

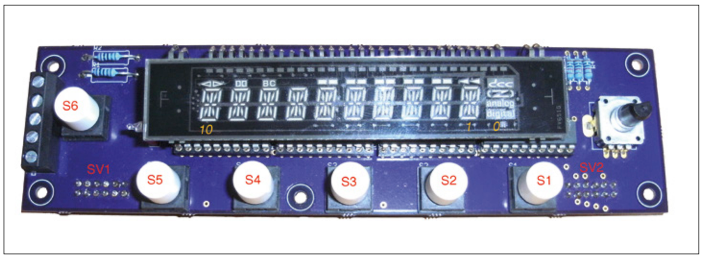


Figure 8. 'Muxer' board with display, drivers, latches, buttons and rotary encoder.

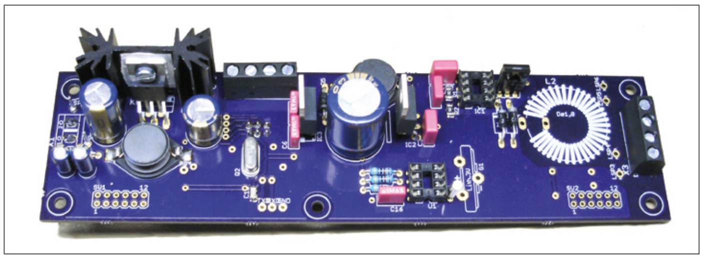


Figure 9. Partially-populated board with power supply and microcontroller.

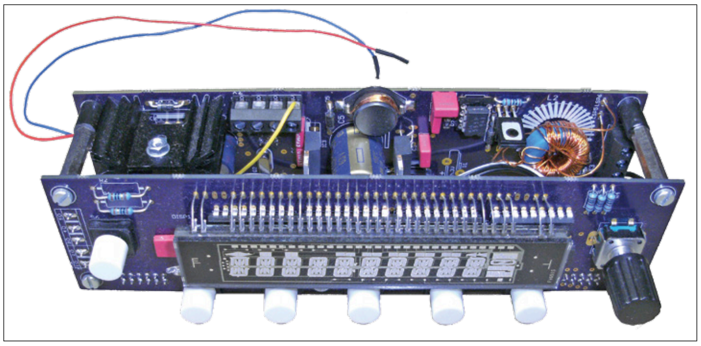


Figure 10. A two-board sandwich.

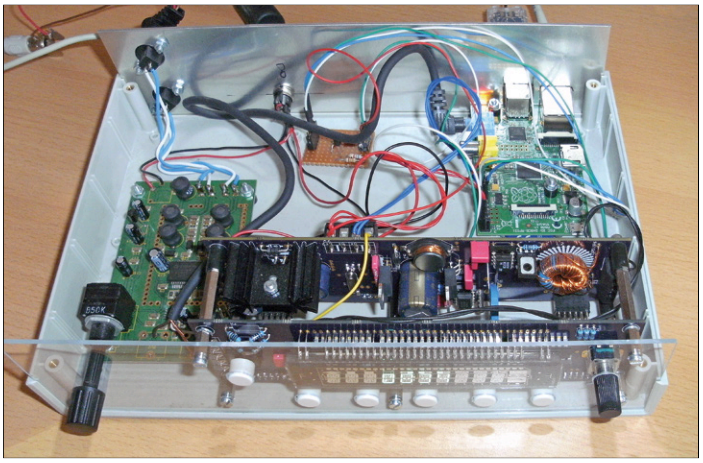


Figure 11. The assembled prototype in a clamshell enclosure, including the Raspberry Pi and the auxiliary circuits such as the level shifter.

Printed circuit boards and construction

The first printed circuit board designed by the author, called ‘Muxer’ (**Figure 8**), carries the display along with its drivers and latches, six buttons, and a rotary encoder with a push switch. The two headers SV1 and SV2 are fitted on the solder side. The drivers and latches are mounted under the display. The circuit diagram [4] also shows the port pins corresponding to the connections on SV1 and SV2.

The second board (**Figure 9**) contains the power supply and the ATmega32 microcontroller, and there is also a real-time clock with battery back-up. The connections for the serial port used to connect the ATmega32 to the Raspberry Pi are brought out. After populating the boards and initial tests the two boards can be bolted together to form a compact module: see **Figure 10**. The module can then be mounted in a box along with the Raspberry Pi. **Figure 11** shows the prototype built into a clamshell enclosure. The board on the left converts the 12 V from the external power supply into a stable 5 V supply, which is also used to power the Raspberry Pi.

Coming up

Although the display module is a versatile unit with many applications, it is the Internet-connected kitchen radio that we will be focusing on in the next and final installment in this series. There we will describe the communication protocol and the software that will be running on the Raspberry Pi. ⏮

(150720)

Internet Links

[1] How vacuum fluorescent displays work: www.explainthatstuff.com/how-vacuum-fluorescent-displays-work.html

[2] Guide to VFD operation: www.noritake-elec.com/display/vfd_operation.html

[3] Heater power supply: www.kerrywong.com/2013/06/05/vfd-filament-driver-using-555/

[4] Article downloads: www.elektormagazine.com/150720

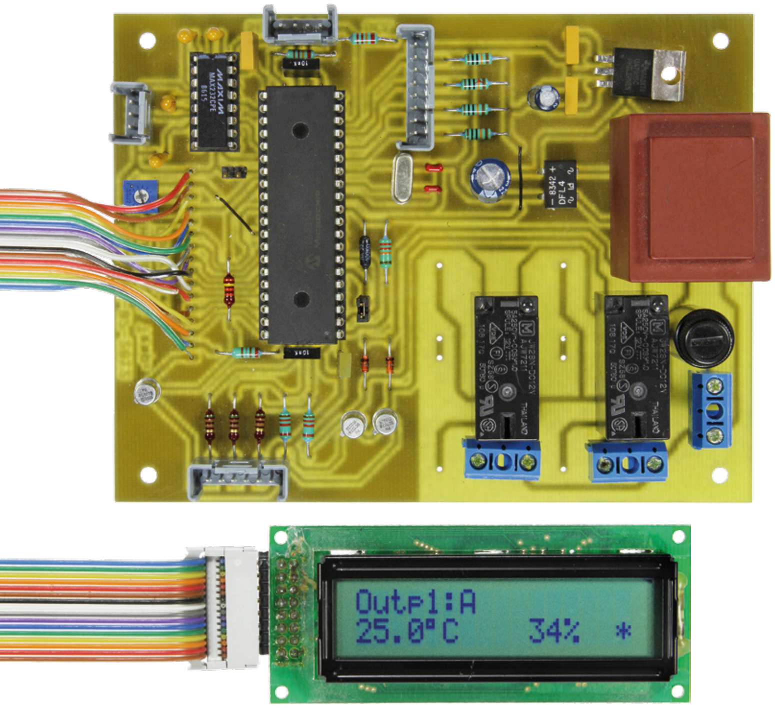
About the Author

After training as a telephone engineer and studying electrical and communications engineering at technical college, Michael Busser spent 20 years at the German state telephone authority; later he worked on a range of projects at T-Systems, a German IT services and consulting company.

He then completed postgraduate study in vocational education at the technical university of Kaiserslautern in Germany, and for the last two years has been teaching electrical engineering and information technology at the BBS TGHS vocational college in Bad Kreuznach, Germany.

Feedback and questions are welcome via michael.busser@t-online.de.

Climate Controller



Measures temperature and/or humidity

By **Goswin Visschers** (Netherlands)

This project provides the ability to control two separate devices, such as an electrical heater and an extraction fan, based on the measured humidity and temperature. Despite the modest number of components, this circuit does offer a large number of settings and is even equipped with a serial interface for communication with your PC.

This project originally began as a fan controller for my bathroom extraction fan. In my renovated bathroom I mounted a humidity sensor near the inlet of the ducting for the extraction fan. When someone is using the shower, the damp air will come into contact with this sensor. This then converts the measured value into a digital number which is subsequently read by the microcontroller and processed into measurement units. The microcontroller then switches a relay on when a preset humidity threshold is exceeded. This relay is, of course, connected to the extraction fan.

The extraction fan is switched off when 30 minutes have passed after the measured humidity has dropped below the preset switching threshold. This prevents the fan from repeatedly switching on and off all the time and we know for sure that the bathroom has been sufficiently ventilated.

Technical characteristics

- Temperature range: –40°C to +125°C, resolution 0.1°C
- Humidity settings range: 0% – 99%, resolution 1%
- Setting range for temperature hysteresis: max. 5°C, resolution 0.1°C
- Setting range humidity hysteresis: max. 10%, resolution 1%
- Outputs can be controlled by combinations of higher or lower temperature than the threshold and/or higher or lower humidity than the threshold
- Adjustable continuation time after the temperature and/or humidity threshold has been passed, in the range from 1 to 99 minutes (separately adjustable for each output)
- Optionally switch on each output separately after an adjustable period from 1 to 99 hours of inactivity; the output then stays on for the above-mentioned continuation time
- Option to switch on one or both outputs for 1 to 99 minutes after the circuit is powered on
- Display of the status of the outputs and the actual temperature and humidity and the option of switching the outputs on and off remotely using the RS232 interface

While I was developing this circuit I realized that there were many more possibilities than just controlling an extraction fan. The software in the microcontroller has now been changed so that the user can choose whether the relay output will switch on or off based on a higher/lower temperature and/or a higher/lower humidity than the preset threshold values. In addition, there was an unused output still available on the microcontroller, so why not connect a second relay that can be operated independently of the first one?

The sidebar lists all the features and characteristics that the current design has to offer.

Hardware

The circuit (**Figure 1**) is a straightforward design. The intelligence is taken care of by a PIC microcontroller type 16F887. Connected to this are the sensor, an LCD, four user-interface buttons, an RS232/TTL converter and a few driver transistors.

The LCD is connected to the microcontroller in the conventional manner. Two control lines and four data lines look after the information on the display. The 4 data lines that are not used are connected to ground. If you do not do this then it is possible that the LCD does not initialize properly and therefore does not operate correctly. The contrast for the LCD is set with the aid of potentiometer R8.

The backlight of the LCD is controlled by the microcontroller via resistor R12 and transistor T3. The backlight is switched off five seconds after the last press on any of the four buttons, to prevent unnecessary energy wastage.

For setting the temperature and humidity thresholds, the 4 pushbuttons are connected to the microcontroller via connector SV3. Four pull-up resistors (R14 through R17) ensure a logic high voltage level at the inputs RC0/1/2/5 when a button is not pressed.

With the aid of jumper J1 the microcontroller can be set to operate with only one or with two relay outputs. When the application (such as my one) requires only one relay then fit jumper J1. The microcontroller will then ignore all the menu options for relay 2. This way it is quicker to change the settings because there is no need to step through unused menu options.

JP2 has not been assigned a function yet. The combined humidity and temperature sensor is connected to connector SV4. The communication with the sensor uses the I²C protocol, the address of the sensor is 0x28 by default. The clock speed of the bus amounts to 100 kHz. This gives trouble free communications with the sensor over a 1-meter long screened cable. The clock speed of the I²C bus is fixed in the microcontroller hardware, it is therefore not possible to reduce this if you wanted to use a longer cable.

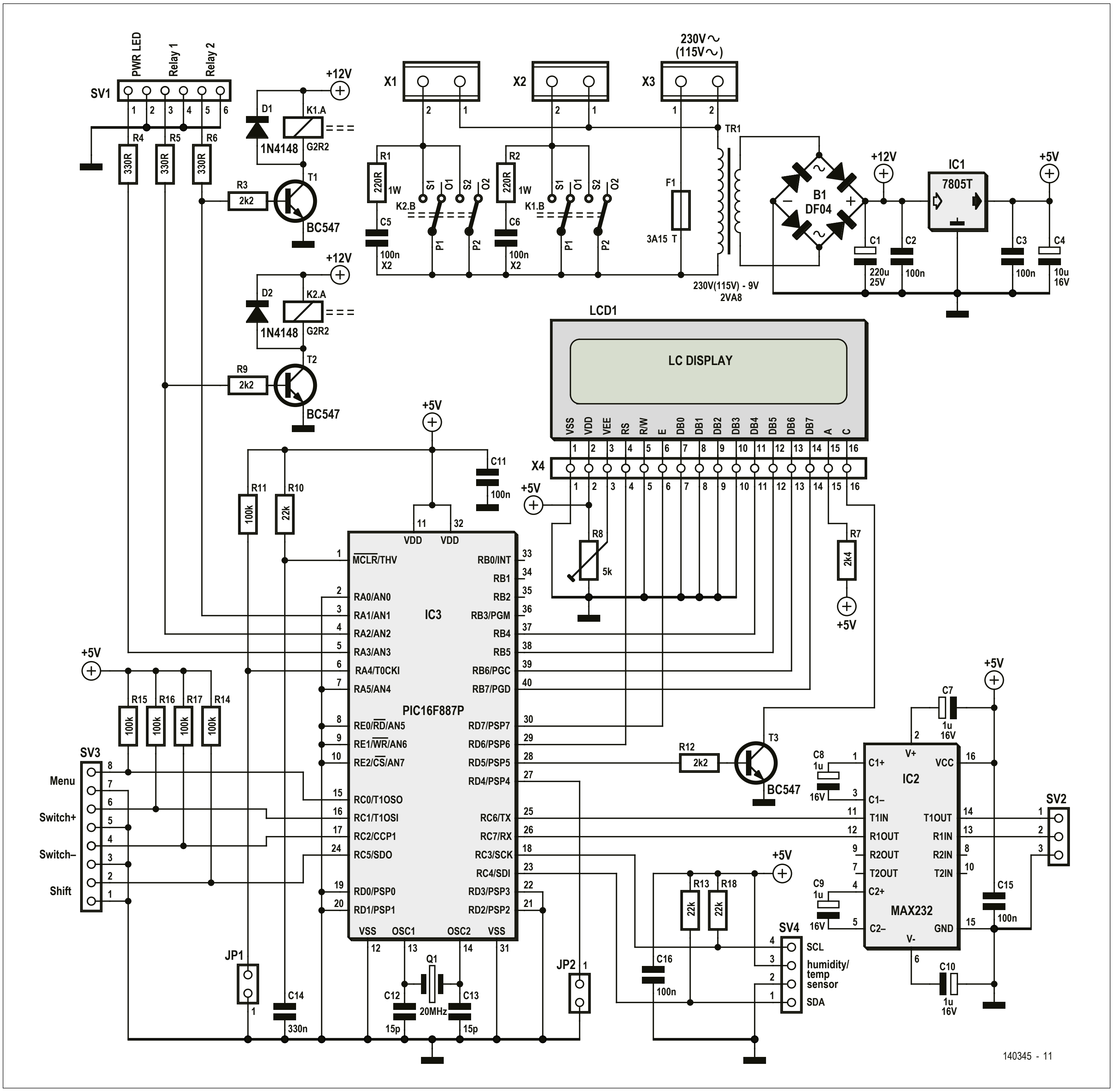


Figure 1. The climate controller is built around a PIC16F887P. The sensor is connected using the I²C bus.

Using the serial interface

status information

Version 2.0s sends a serial string every two seconds (after a new measurement has taken place) which contains the measured values and the present state of the relays via the serial port (SV2). The settings for the serial port are as follows:
9600 baud, 8 bits, 1 stop bit, no parity
These values can be stored on a PC for further processing.
Example of such a serial string:

```
T:23.3°C H:44% O1:0 O2:0[CR][LF]
```

The string has a fixed length. In the event that the temperature has a negative sign or is higher than 99 degrees C then the space after the 'C' symbol is omitted. With the humidity there is an additional space after the '%' when this is lower than 10%.
The value after 'O1:' and 'O2:' indicates with a '1' or '0' whether the output is switched on or off.

Sending commands

When an output in setting '`OutputX ON set:`' is configured to 'RS232', the climate controller can be controlled remotely.
The climate controller accepts the following command:

```
OX=S[CR][LF]
```

X = the number of the output, in our case this can therefore be either 1 or 2.
S = the status of the output, in our case that can be either 0 or 1.
Example: `O1=1[CR][LF]`

This switches output 1 on.
After the command has been sent, the status of the output can be checked by looking at the status information string which is sent by the controller every two seconds.

Menu structure

Overview of the settings options of version 2.0 (additional points relevant to version 2.0s are shown in red)

Selection of the output-mode for the relay 1 ('&' = and, '+' = or)
`Output1 ON set:` [T high | T low | Hum high | Hum low | T high &Hum high | T high & Hum low | T low & Hum high | T low & Hum low | T high +Hum high | T high + Hum low | T low + Hum high | T low + Hum low | ON | OFF | RS232]

Selection of the output-mode for relay 2 ('&' = and, '+' = or)
`Output2 ON set:` [T high | T low | Hum high | Hum low | T high &Hum high | T high & Hum low | T low & Hum high | T low & Hum low | T high +Hum high | T high + Hum low | T low + Hum high | T low + Hum low | ON | OFF | RS232]

Setting of the temperature threshold value at which an output has to be switched on or off
`T thres Outp1:` [–40 – 125]
`T thres Outp2:` [–40 – 125]

Setting of the humidity threshold value at which an output has to be switched on or off
`Hum thres outp1:` [0 – 99]
`Hum thres outp2:` [0 – 99]

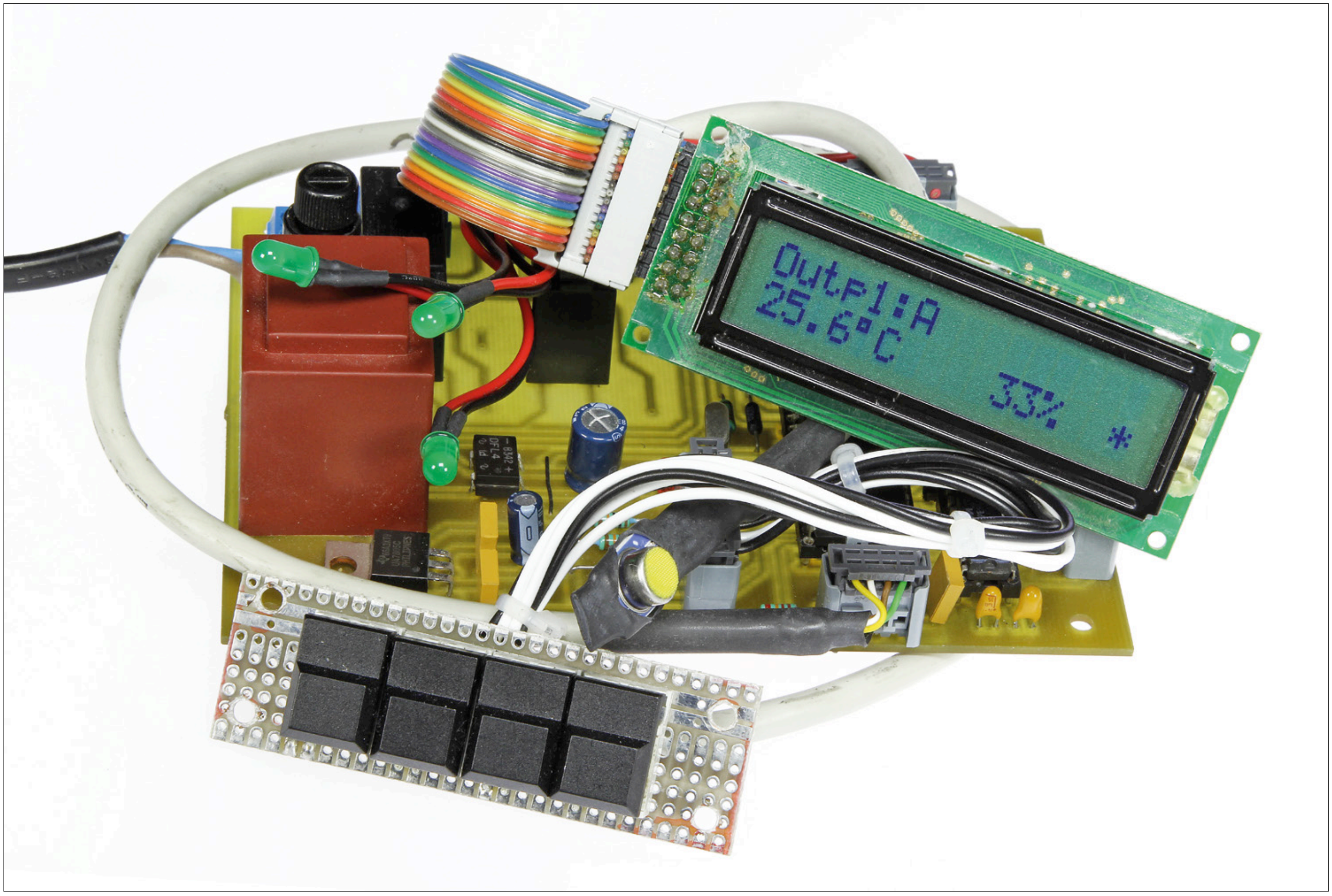
Setting of the temperature hysteresis for both outputs (not present in version 2.0s)
`T hysteresis:` [0 – 5°C]

Setting of the humidity hysteresis for both outputs (not present in version 2.0s)
`Hum hysteresis:` [0 – 10%]

Setting of the maximum inactive time of an output
`Outp1 nonactive` interval: [OFF – 99 hrs]
`Outp2 nonactive interval:` [OFF – 99 hrs]

Setting of the continuation time and the time that an output is active after an inactive period
`Outp1 threshold act time:` [0 – 99 min]
`Outp2 threshold act time:` [0 – 99 min]

Setting of the time that an output has to be active after the circuit is turned on
`Outp1 power on act time:` [OFF – 99 min]
`Outp2 power on act time:` [OFF –99 min]



To protect the connected sensor(s) from spikes on the power supply voltage a capacitor of 100 nF is soldered across the power supply terminals of connector SV4. The two relays are each controlled by a standard NPN transistor (T1 and T2). The relay coils are connected to the unregulated power supply voltage which is about 12 V.

In parallel with the contacts of each relay is a discrete RC-snubber network (R1/C5 and R2/C6 respectively) which suppress voltage spikes during switch-off. When an electric motor or other inductive load is switched off this can generate a brief but nevertheless high voltage that then results in a spark between the relay contacts. These sparks not only shorten the life of the relay but sometimes also disturb the correct operation of the microcontroller.
The disadvantage of this snubber network is that there will always be a small current running through the load, also when the relay contacts are open. It is therefore recommended to fit the snubber network only when switching an inductive load.

It is also possible to read the measured values remotely and control the microcontroller from a distance. The voltage-level shifting for the RS232 communications is provided by a MAX232 (IC2). If necessary, you can connect the circuit via a USB-RS232 adapter because the PC does not have an RS232 interface any more. Three status LEDs can be connected to connector SV1. The power LED flashes at a rate of 1 second to indicate that the circuit is operational. The other 2 LEDs indicate the status of the relay outputs. The power supply is provided by a small PCB transformer (TR1), the secondary voltage of which is rectified (B1) and regulated (IC1). The voltage regulation is handled by the well-known 7805 which is supported by 3 capacitors (C2, C3 and C4) for further stability and the suppression of the tendency to oscillate.
Connected in series with the line voltage connection X3 is a fuse holder with a fuse of 3.15 AT (slow blow).

Software: 2 versions

The program in the microcontroller is written in mikroC Pro version 6.4 with a full license. Unfortunately the available program space of 14 kB in the microcontroller was too limited to implement all the desired features simultaneously. That's why I chose to write two separate variants of the software:

- Version 2.0 does not use RS232 communications. Instead it is possible to set the temperature- and humidity-hysteresis separately.
- Version 2.0s supports the RS232 communications, but in this version the hysteresis settings for temperature and humidity are omitted.

Both versions of the software are in the free download available from the Magazine website [1].

Construction

The author has designed a printed circuit board for this circuit, which contains all the electronics and the mains power supply (see **Figure 2**, download available from [1]). Only normal leaded components have been used, which makes the assembly of the circuit board very straightforward.

The display is connected with a short length of 16-way ribbon cable to connector X4. The sensors are connected in parallel with a 4-way (screened) cable with a length of no more than 1 meter (approx. 3 feet) to SV4. The four push-buttons can be mounted on a small piece of experimenting board and connected with a short cable to SV3. A 9-way sub-D connector is attached to SV2, but only if you actually intend to use the serial interface connection (and use software version 2.0s). The three status indicator LEDs are connected to SV1.

After all this the circuit is ready for use. The mains power cable is connected to X3 and the load(s) to X1 and X2. The entire circuit has to be housed in a suitable, well-insulating enclosure so that there is absolutely no risk of touching any of the high voltage parts. When used in a bathroom there are additional rules that apply to line-powered devices and also a splash-proof enclosure has to be used. Once the power supply voltage is switched on the circuit is immediately ready for use, you only need to set the contrast potentiometer of the LCD and you can then set the desired switching thresholds and times.

Operation

All settings are done using the four push-buttons. The 'Menu' button is used to step through the different settings. The purpose of buttons '+' and '–' is to increase or decrease the selected value. The step size can be increased by simultaneously pressing the 'Shift' butting in combination with the '+' or '–' button.

The changed value is stored as soon as the 'Menu' button is pressed. When the buttons are not operated for more than five seconds, then the changed value is not stored and the original values will reappear on the LCD.

To read the measured values in the dark you can briefly press the 'Shift' button; the backlight will then switch on for a period of five seconds.

You can find an overview of all the settings options in the **inset** 'Menu structure'. ◄

(140345)

Web Link

[1] www.elektormagazine.com/140345

Editor's Note

Additional information on this project and a suggestion for an alternative sensor may be found in Err-lectronics, *Elektor Magazine* July & August 2016, page 120.

Component List

Resistors

R1,R2 = 220Ω 1W
R3,R9,R12 = 2.2kΩ
R4,R5,R6 = 330Ω
R7 = 2.4kΩ
R8 = 5kΩ trimpot, horizontal
R10,R13,R18 = 22kΩ
R11,R14–R17 = 100kΩ

Capacitors

C1 = 220µF 25V, radial
C2,C3,C11 = 100nF
C4 = 10µF 16V, radial
C5,C6 = 100nF 400V, class X2
C7–C10 = 1µF 16V, radial
C12,C13 = 15pF
C14 = 330nF

Semiconductors

D1,D2 = 1N4148
B1 = DF04 bridge rectifier (280V piv, 1A)
LEDs, 3mm, for SV1, 3 pcs
T1,T2,T3 = BC547
IC1 = 7805T
IC2 = PIC16F887P
IC3 = MAX232

Miscellaneous

TR1 = power transformer, PCB mount, 9V / 2.8VA secondary (e.g. Weiss EI 30/23 BV 85/341)
Q1 = 20MHz quartz crystal
X1,X2,X3 = 2-way PCB screw terminal block, 0.2'' lead pitch
K1,K2 = PCB relay with DPDT contacts, 250V / 5A, 12V coil (e.g. Panasonic JW2SN-DC12V)
SV1 = 6-pin pinheader, 0.1'' pitch
SV2 = 3-pin pinheader, 0.1'' pitch
SV3 = 8-pin pinheader, 0.1'' pitch

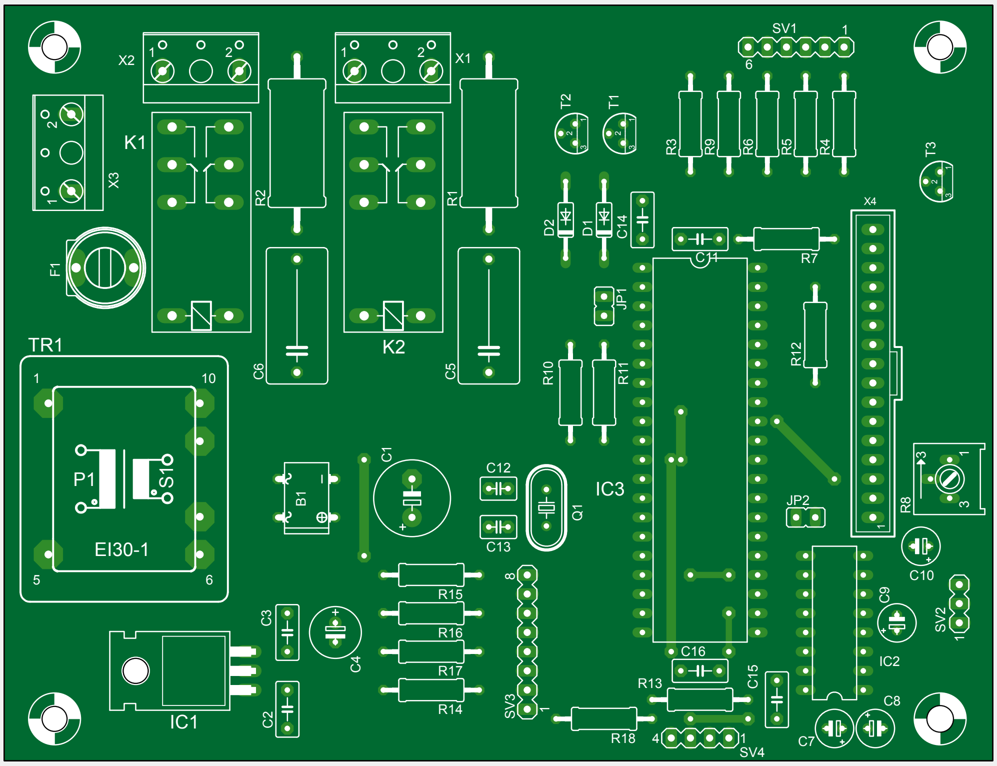


Figure 2. The circuit board is spacious and also contains the power transformer. Because of this take care that the circuit is housed in a safe enclosure.

SV4 = 4-pin pinheader, 0.1'' pitch
X4 = 16-pin pinheader, 0.1'' pitch
LCD, 2x16 characters, e.g. Gleichmann GE-C1602B-YYH-JT/R, Conrad Electronics # 183043 - 89

Humidity and temperature sensor with I2C interface, type **IST** HYT221 (Conrad Electronics #, 505671 - 89)

huge **peak**
strong **bead!**

WÜRTH ELEKTRONIK

#strongBEAD

WE speed up the future

electronica FAST FORWARD AWARD powered by elektor
08. – 11. November 2016 | Messe München
GOLD SPONSOR
WÜRTH ELEKTRONIK

Gold Sponsor **electronica Fast Forward Award,** powered by Elektor

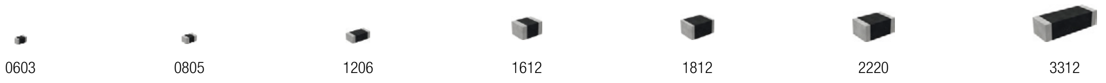
electronica Munich Hall B6 Booth 404

The WE-MPSB series is the world's first ferrite bead that specifies how it performs under high current transients. This unique bead protects and extends the life of your application. It features an ultra-low R_{DC} which delivers the lowest self-heating in its class at high currents. The WE-MPSB is ideal for DC/DC applications requiring high efficiency. For more information, please visit **www.we-online.com/WE-MPSB**

- muscular peak current capability
- high rated currents
- ultra-low R_{DC} to minimize unwanted losses
- effective broadband filtering

Design your filter with our free online software – **REDEXPERT.** You will have the best guidance to ensure your circuit can withstand transient peak currents using our peak-specified WE-MPSB ferrite beads: **www.we-online.com/MPSB-designer**

Products in original size:

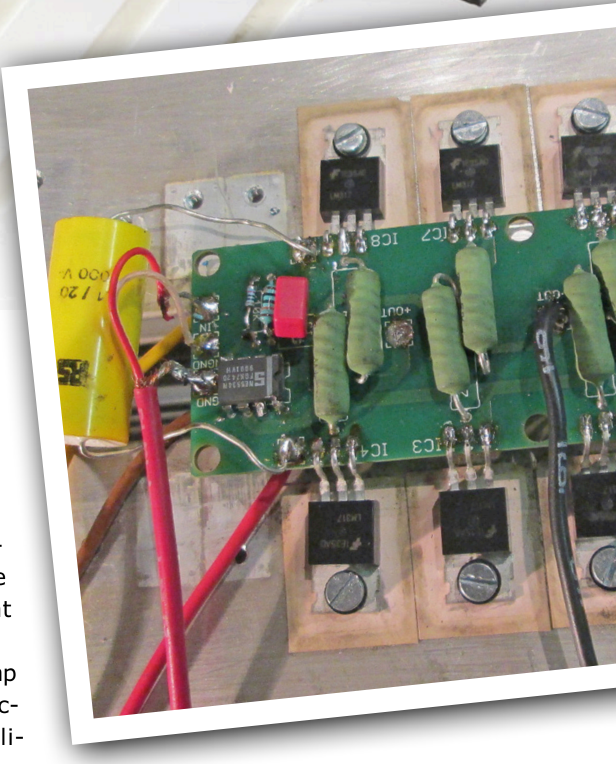


Simple Class-A Amplifier

with an LM317

By **Jurgen Michiels** (Belgium)

Although there are a number of initial considerations when designing an audio amplifier (power, distortion, robustness, etcetera) it is sometimes amusing and interesting to take a completely different approach, using components that haven't been designed for this purpose, but can nevertheless achieve very good results. For example, a simple Class-A audio amplifier based on the … LM317!



The widely known LM317 voltage regulator, reintroduced in 1976, maintains its output voltage always 1.25 V higher than the voltage on its adjust pin, where this pin requires only 100 µA while the output current can be up to 1.5 A. So this is, in fact, an amplifying element. Why then not build an amplifier with it?

Design

Figure 1 shows the basic circuit diagram. IC1 is controlled by opamp IC3. IC2 is configured as a current source and in this way ensures that the current can also return during the negative half of the audio signal. This current is set by R9. Since the output voltage of the LM317 is always 1.25 V higher than the voltage at the adjust pin, this current is equal to 1.25/R9. Finally, R11 and R10 determine the gain of the circuit.

Because of feedback, IC3 will compensate for the 1.25 V offset voltage and maintain the output voltage at 0 V on average.

Brief summary: two LM317s, an opamp and three resistors are all that is necessary for a simple Class-A amplifier (for the audiophiles: there are no capacitors in the signal path!).

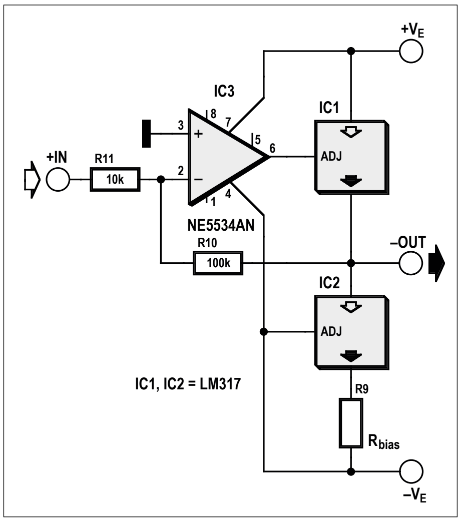


Figure 1. The operating principle of this Class-A amplifier us very simple.

Schematic

Given that an LM317 can handle a maximum current of 1.5 A, this results in an output power capability that is not all that high. Fortunately this can be overcome by connecting multiple LM317s in parallel, see **Figure 2**.

The maximum input voltage for an LM317 is 40 V, so a symmetrical power supply of 20 V is the theoretical maximum power supply voltage that is allowed. The opamp, however, can only tolerate a maximum symmetrical power supply voltage of 18 V. An unregulated power supply voltage of 15 V is therefore a safer choice. Using this we can determine the required bias current. With a load of 8 ohms the theoretical maximum current is equal to 15 V / 8 Ω = 1.875 A.

The theoretical maximum power will then be about 14 watts continuous, although it will be closer to a good 12 W in practice. This is for a pure resistive load, however. Since this is not the case for a loudspeaker, we choose a bias current that is a little higher. This does, however, also determine the power dissipation of the power stage. The current was chosen to be 2.2 A. R_{bias} is then

1.25 / 2.2 = 0.56 Ω.

With this the resistor will dissipate just below 3 W, so it is recommended that a 5-W resistor is used here.

Finally, the total dissipation of the power stage amounts to

30 × 2.2 = 66 W.

Parallel

With four LM317s the maximum output current is allowed to be up to 6 A. A quiescent current of 2.2 A results in a maximum current of 4.4 A through the top LM317s and 2.2 A through the bottom LM317s, so is perfectly achievable. However, 4 ohms requires twice the amount of current, which cannot be achieved with four LM317s, unless the power supply voltage is lowered.

In this amplifier the LM317s are connected four in parallel, with *current sharing* resistors to share the current (more) equally among the four LM317s. Decoupling has also be added for the power supply rails. The input impedance is determined by R11. A 10 k resistor was selected for this. Finally, the gain can be set using R10. This is calculated as follows:

A = –R10 / R11.

Here a gain of 10 was selected, hence R10 needs to be 100 kΩ.

C1 determines the bandwidth and prevents high frequency oscillations. Here a value of 100 pF was chosen, which limits the bandwidth to 100 kHz. But you can experiment with this to your heart's content.

The characteristics are mainly determined by the opamp that is used. With the NE5534 a THD of better than 0.05% is easily obtained. By the way, note that the output is inverted and therefore the 'plus' of the loudspeaker needs to be connected to ground and the 'minus' to the outputs of the LM317s.

The design files (Eagle format) and a circuit board layout are available from the project page for this project [1]. ◀

(150789)

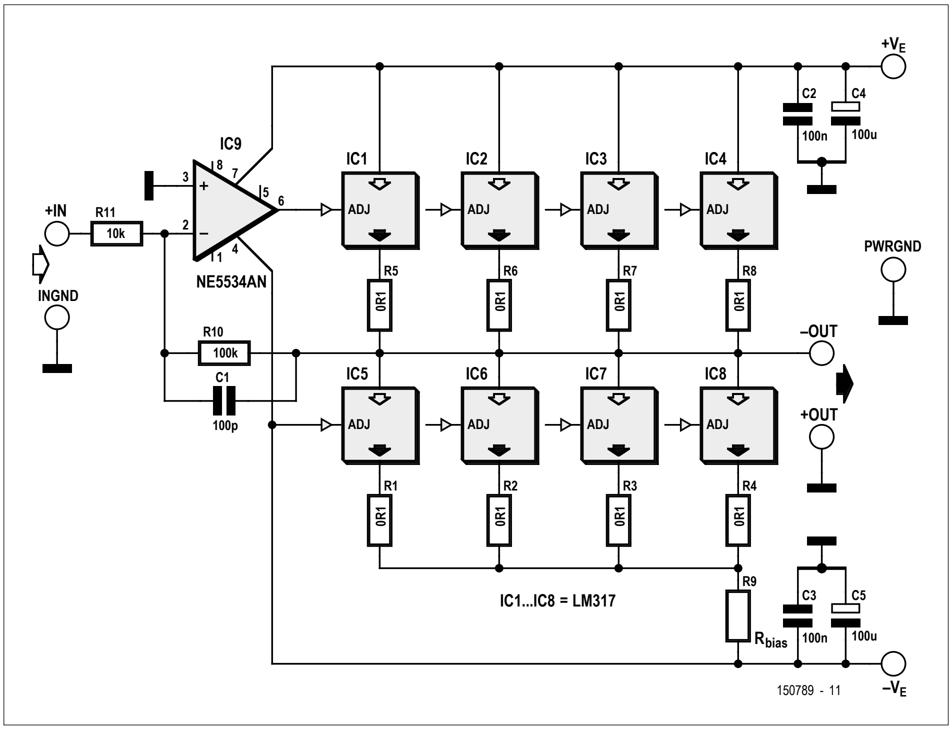
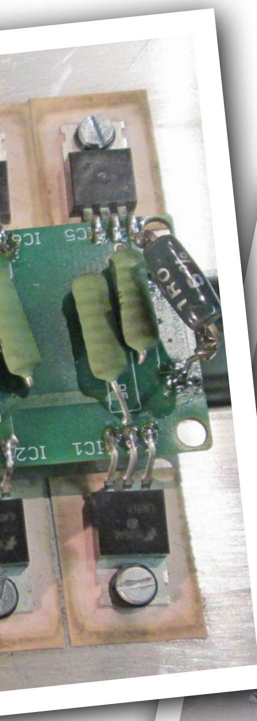
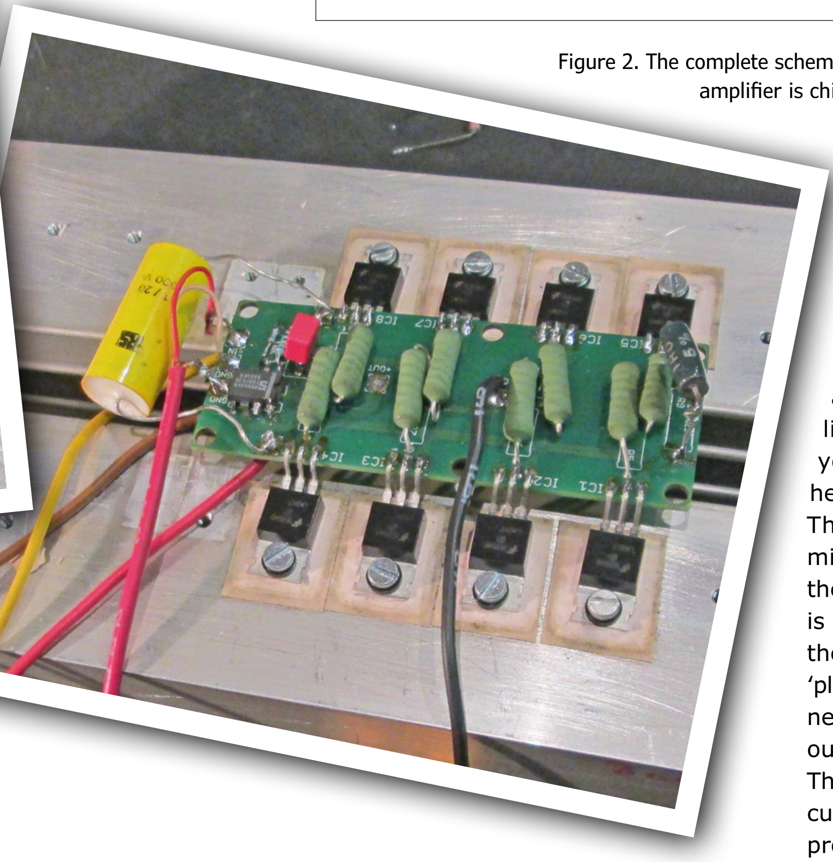


Figure 2. The complete schematic (without the power supply) of the amplifier is child's play, but nevertheless not at all bad.





No capacitors in the signal path

Web Link

[1] www.elektormagazine.com/150789

[2] http://vintageaudioprojects.be

Door Spy using Raspberry Pi

Communicate with a visitor using your smartphone

By **Luc Lemmens** (Elektor Labs)

An unexpected visitor can be a welcome surprise, but at other times you will have regretted answering that doorbell when and finding a less desirable guest on your doorstep. An intercom with a camera at the front door can be a solution, but that's of no use to you when you are not at home. With the project presented here you can, in principle, see from anywhere in the world who rang your doorbell, have a conversation with your visitor and even remotely open the door for them, provided you have your smartphone on you.

Using a Raspberry Pi (B+, 2 of 3, any of these will work), a Raspberry Pi camera and an RPi GSM HAT board designed by us, we will in this article build a doorbell that will take a picture of the visitor the instant they ring the doorbell. The picture is then sent in an MMS-message to your smartphone, so that you can see who is at your door. You can now call your door; the GSM module will answer automatically and after a bell signal you can converse with your visitor, using the built-in microphone and loudspeaker. There is even the possibility of sending an SMS message to the bell, that, with the correct password, can control an electric door opener. As an additional security message it also verifies the mobile phone number that sent the message to make sure that is was sent from your smartphone.

Although the system was devised, and is mainly intended, for those occasions when you are not at home, it can also be handy when — for whatever reason — you are not, or not easily able, to answer the front door. And, of course, you can also use it to decide whether you actually want to be at home for this particular visitor...

The project grew as a spin-off from the GSM breakout board [1], which uses an M95 GSM module from Quectel. This board already contained some features for this doorbell project, but the printed circuit board design was not really suitable to make it into a nice and tidy, compact module. Hence our new HAT board for the RPi, which also contains additional hardware to make the door spy complete. First let's take a closer look at the schematic.

Schematic

The greater part of this schematic has been taken directly from document 'M95 hardware design' from Quectel [2]. At the heart of the circuit (**Figure 1**) there is obviously the M95 GSM-module MOD1 with antenna and SIM-card holder SIM1. The resistors R15 through R17, the diode array D7 and the capacitors C19 through C22 serve as noise suppression and ESD protection for the SIM-card interface. The same is true for capacitors C6 through C17 and D3 through D6 for the microphone input and the audio amplifier output of the M95 respectively.

The M95 is connected to the Raspberry Pi (K5) via its USART, all communications are done using AT commands. Voltage divider R6/R7 adapts the TxD signal to the 3.3 V rated digital input level of the RPi. It would have been the obvious choice to operate the entire GSM board from a 3.3 V power supply voltage to prevent a difference in logic levels, but oddly enough we then run into a different problem. Although the power supply voltage of the M95 is specified to be in the range of 3.3 to 4.6 V, the module continually sends unsolicited warnings via the UART when the power supply voltage approaches these limits within about 100 mV. This interferes with the decoding of those messages that we do want to receive from the M95, and furthermore, the M95 can then suddenly switch itself off. That is why the power supply level, using a MIC29302 (IC1), is set to 4.4 V, not a particularly common logic level. Nowhere in the documentation does Quectel elucidate what the thinking behind this behavior is.

LED2 begins to flash as soon as the M95 is switched on. The label 'netlight' that Quectel uses to name this output would suggest to us that the module is then connected to a mobile network, but that is not the case. The M95 is in a standby mode Immediately after the power supply

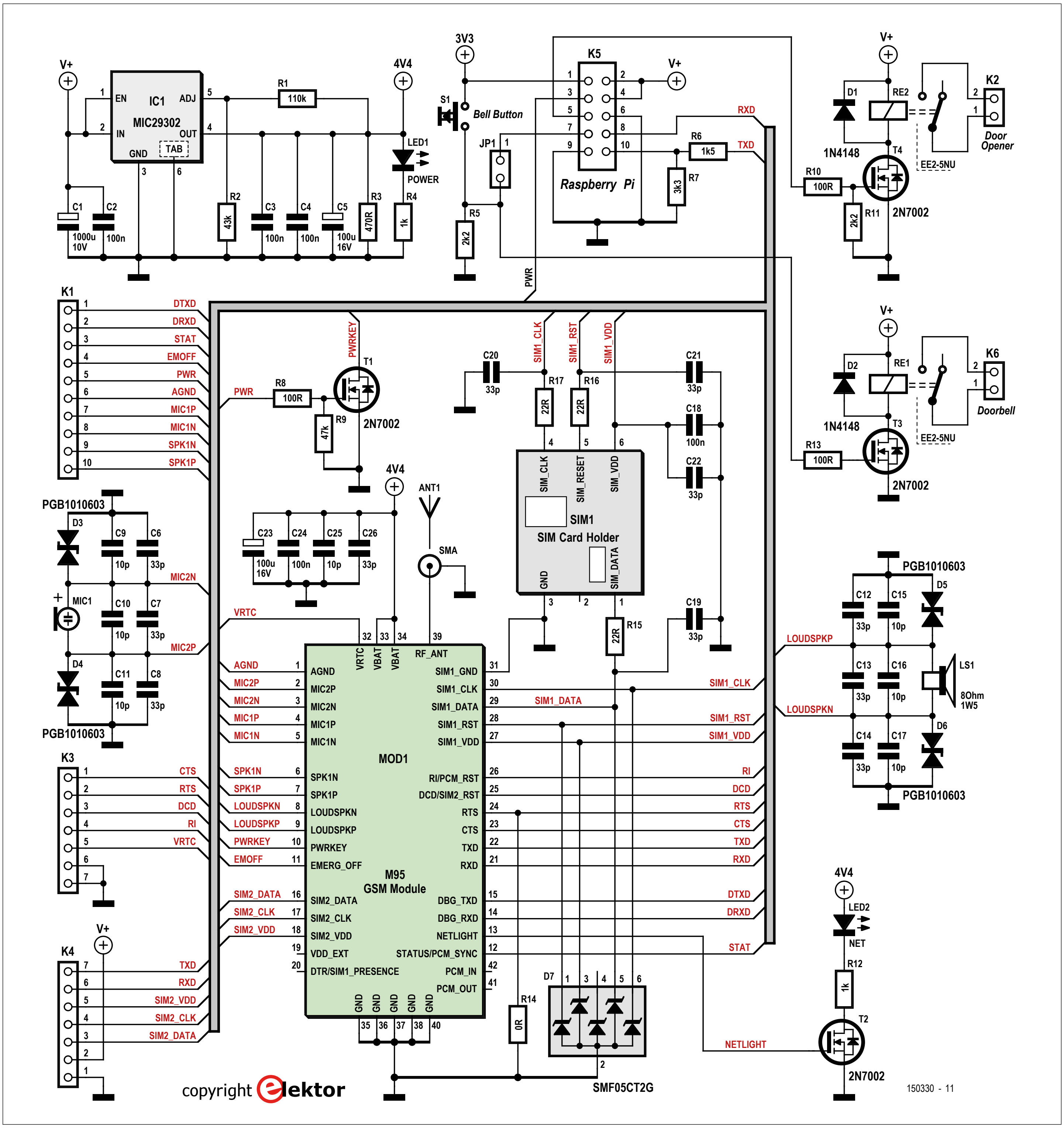


Figure 1. The major parts in this circuit are the M95 GSM module and the SIM card holder.

voltage is applied, a pulse on the ‘pwrkey’ line toggles the module on and off. Pushbutton S1 is the ‘regular’ doorbell pushbutton near the front door. In this case the button operates relay RE1, which can be used to activate the existing (AC) doorbell. JP1 (optionally can be replaced with a switch) determines whether or not the bell signal is sent to the RPi and the transmission of the picture commences. Sending an MMS message costs money, so when you are at home a normal doorbell is usually sufficient.

RE2 is intended for connecting an automatic door opener. This relay is switched from the RPi via T4 when an SMS has been received that contains the correct password and has been sent from the correct mobile phone. To be frank, we ourselves are not all that fond of this feature. It requires knowledge in the area of (hardware) security to prevent someone from operating or bridging this relay and in this way gain access to your house. So using this feature is therefore entirely the responsibility of the user!

Printed circuit board

The circuit board that we have developed for the project can be seen in **Figure 2**.

Component List

Resistors

Default: SMD 0603, 1%, 0.1W)
R1 = 110kΩ
R2 = 43kΩ
R3 = 470Ω
R4,R12 = 1kΩ
R5,R11 = 2.2kΩ
R6 = 1.5kΩ
R7 = 3.3kΩ
R8,R10,R13 = 100Ω
R9 = 47kΩ
R14 = 0Ω
R15,R16,R17 = 22Ω

Capacitors

C1 = 1000µF 10V, radial SMD
C2,C3,C4,C18,C24 = 100nF, X7R, SMD 0805
C5,C23 = 100µF 16V, SMD 2312
C6,C7,C8,C12,C13,C14,C19,C20,C21,C22,C26 = 33pF, X7R, SMD 0805
C9,C10,C11,C15,C16,C17,C25 = 10pF, X7R, SMD 0805

Semiconductors

D1,D2 = 1N4148, SOD-323
D3,D4,D5,D6 = PGB1010603MR, SMD 0603
D7 = SMF05CT2G, SC-88
LED1,LED2 = LED, green, 50mcd, 2.1V @ 20mA, SMD 0805
T1,T2,T3,T4 = 2N7002, SOT-23
IC1 = MIC29302WU TR, TO-263

Miscellaneous
Mod1 = GSM module, Quectel type M95
SIM1 = microSIM socket (Molex 78723-1001)
ANT1 = SMA connector, PCB mount (Molex 73391-0070)
+ SMA antenna, right angled, (RF Solutions ANT-GHEL2R-SMA)
K1 = 10-way pinheader socket, 0.1’’ pitch (optional)
K2,K6 = 2-way pinheader socket, 0.1’’ pitch (optional)
K3,K4 = 7-way pinheader socket, 0.1’’ pitch (optional)
K5 = 10-way (2x5) pinheader socket with extended pins (Samtec ESQ-105-14-G-D)
S1 = 2-pin pinheader for doorbell connection (with make contact)
JP1 = 2-pin pinheader with jumper
LS1 = 2-pin pinheader for mini loudspeaker connection (8Ω, 1.5W)
Mic1 = 2-pin pinheader for electret microphone connection (ABM-715-RC)
Re1,Re2 = relay, SMT, DPDT, 2A, 5V$_{DC}$ (Kemet EE2-5NU)
Raspberry Pi B+, 2 B or 3 B
Raspberry Pi Camera Board (V1.3)
PCB # 150400-1 from Elektor Store
or
PCB with SMDs preassembled: 150400-71 from Elektor Store

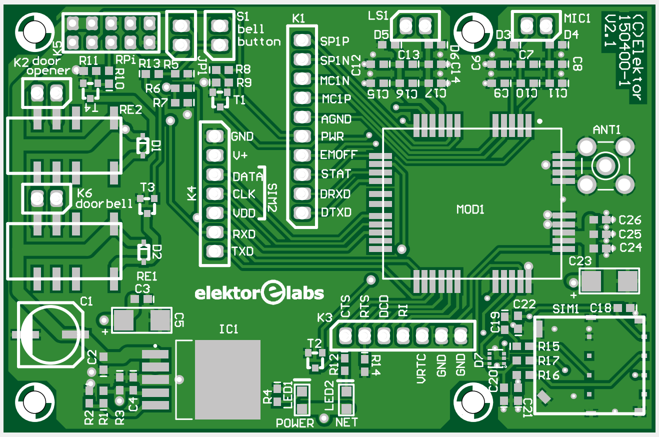


Figure 2. The printed circuit board for the door spy is designed as an RPi-HAT and contains mainly SMDs.

Since the GSM-module is only available in a SMD version, we decided to use SMD parts for all the components, with the exception of the antenna connection and the other connectors. This does, however, make the assembly more difficult for those who are less experienced in SMD soldering. But to make it easy for you, we also offer the PCB with all the SMDs soldered already (no. 150400-71). Real diehards can etch (or have etched) a circuit board themselves (layout can be downloaded from [4]) or a bare circuit board can be ordered from the Elektor Store and start from there. Note that various connectors are not absolutely necessary — this is mentioned in the component list. With K5 you need to insert the socket from the bottom of the circuit board and then solder it from the top side (see **Figure 3**). This connector is later plugged into the GPIO connector of the Raspberry Pi. Before soldering it in place, check whether the height of connector K5 needs to be adjusted so that the solder connections on the bottom stay far enough away from the USB connectors on the RPi.

Preparing the RPi

With the current version of Rasbian Jessie it is reasonably simple to make the Raspberry Pi suitable for our door spy. Download the most recent image from [3], at the time of this writing that is kernel version 4.4, May 2016. Use Win32 DiskImager (or an equivalent tool) to copy the .IMG to the SD card. Boot the RPi with this, open a terminal window and change the settings when starting up with the following command:

```
sudo raspi-config
```

First select option 6. Enable Camera to turn the RPi camera on. After this is done, go to 9. Advanced Options, from here choose A7 Serial and then select ‘No’ to disable access to the login shell via the serial port. After that select Finish and restart the RPi.

Note: in older versions of Raspbian the camera has to be switched on manually and the serial console has to disabled in startup files, for additional help and further explanation refer to the numerous examples that can be found on the Internet.

This does then require a very important change to the Python script for the *door spy: on line 45 the '#'-sign in front of 'port.open' needs to be deleted!* We have had a working prototype for some time, so when, for writing this article, we repeated the installation again 'from scratch', we did that with the most recent version of Raspbian and our script suddenly generated the error message 'Port already open' for this line 45. Apparently the new version makes the line for the 'open' command unnecessary.

Connecting the hardware

Now that we have the Raspberry Pi in order, we have to connect a few things before we can carry out our first test. It is easiest to start by connection the RPi camera; the flex cable can be routed along the bottom of our HAT board.

This board fits exactly on top of the RPi. Verify again that there is sufficient space between the underside of the board and the USB connections of the RPi. In particular look at the antenna connector ANT1 to make sure that it is far enough away from the metal shielding of the USB connectors. You can now assemble the boards neatly together at the correct separation using small bolts and standoffs **(Figure 4)**.

Temporarily connect an arbitrary make pushbutton as 'doorbell' to the terminals of S1 and link jumper J1. The microphone and loudspeaker are attached to MIC1 and LS1 respectively. The antenna is connected to ANT1 and we now need a micro-SD card which will go into SIM1. But before you rush to the shop to purchase one of these first note the following...

SIM card and MMS

The M95 GSM-module in our application will only be able to do its job if there is a SIM card present with a sufficient prepay balance (or subscription, of course). Here in the lab we worked with two different prepaid SIM cards and a well-intentioned warning beforehand: perform careful comparative research before you buy a card. As mentioned earlier, the photo of the visitor is sent using MMS (Multimedia Messaging Service) and apparently this is used relatively little so that various providers deem it unnecessary to clearly specify the costs of this service. For as far as we could determine, most providers have a fixed tariff per MMS-message, provided the attachment – in our case the photo file in JPEG format – is not larger than 300 kB. With the settings for the RPi camera that we use in the script we stay well below that limit, also to reduce as much as possible the amount of time that is necessary to send and receive the JPEG. The latter is of course also very important: When the time between ringing the bell and reacting via the 'intercom' is too long there is a significant chance that the visitor has left already. For us in the lab it took some 20 to 30 seconds before the smartphone indicated that an MMS message was received. Add to that the time it takes for the recipient to act and you may therefore conclude that the system has little utility when the guests are impatient.

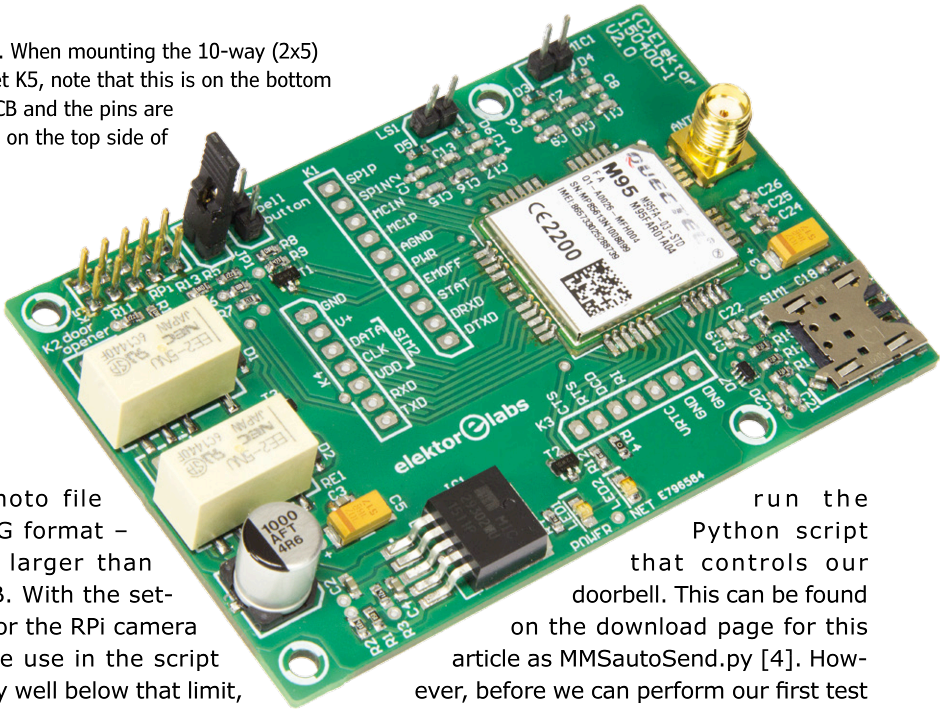


Figure 3. When mounting the 10-way (2x5) pinsocket K5, note that this is on the bottom of the PCB and the pins are soldered on the top side of the PCB.

Python script for the doorbell and first test

After setting up the Raspberry Pi and installing the Python libraries we are ready to run the Python script that controls our doorbell. This can be found on the download page for this article as MMSautoSend.py [4]. However, before we can perform our first test we first have to make some changes to this script. For this reason it will be convenient to connect a monitor, a keyboard and perhaps also a mouse while doing this testing. Once all the settings are correct the application will work without these peripherals.

At the top of the script these is a block of constant definitions that needs to be set appropriately:

- ThisNumber: the mobile number of the SIM card for the M95
- ThatNumber: the number of the smartphone to which the MMS-message is to be sent
- Settings for using the MMS service of the provider of the SIM card for the GSM-module, specifically: APN: Access Point Name

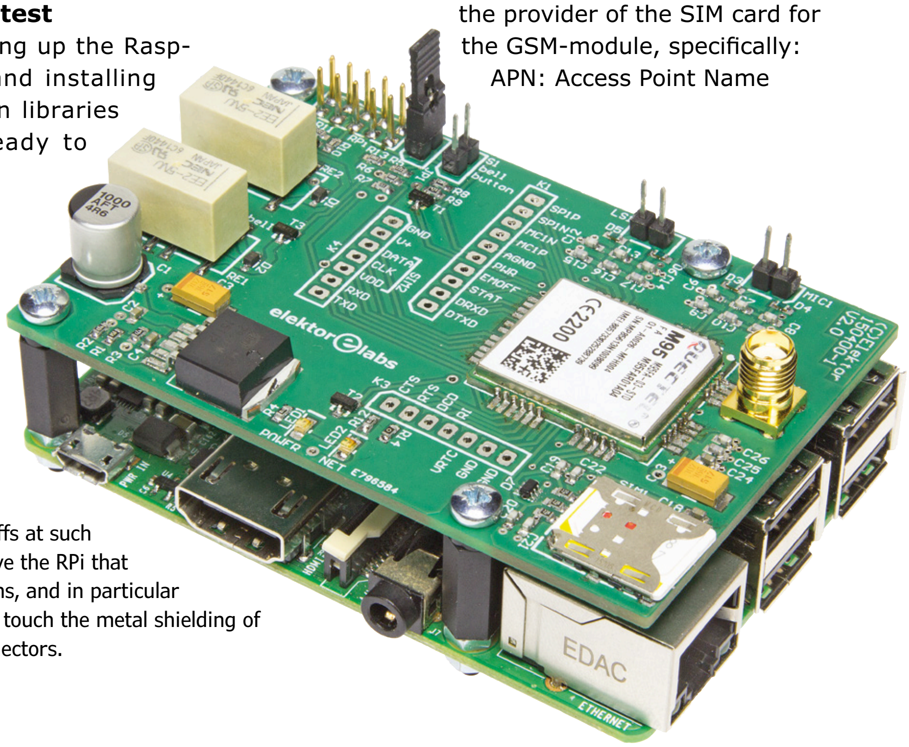
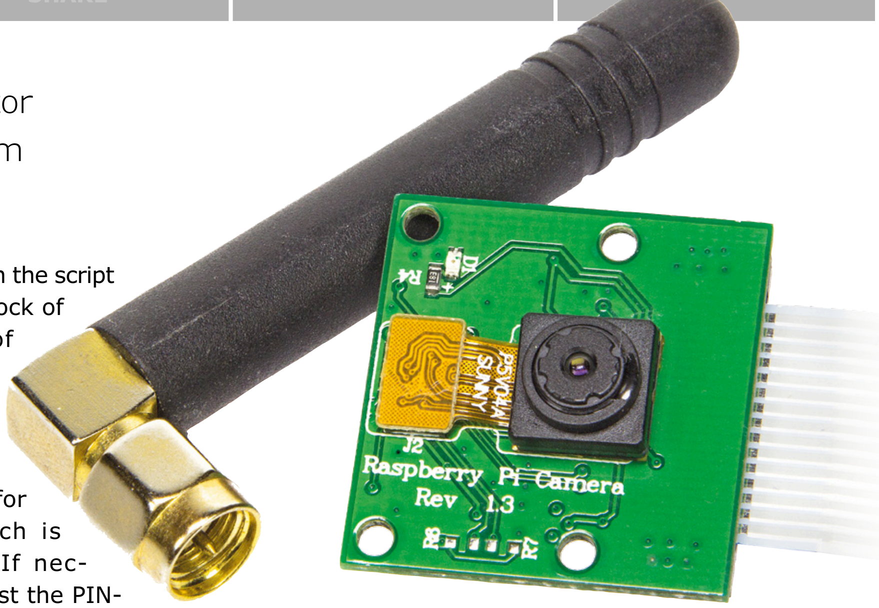


Figure 4. Mount the circuit board using standoffs at such a height above the RPi that the solder pins, and in particular ANT1, do not touch the metal shielding of the USB connectors.

You receive a snapshot of the visitor and you can then even talk to them



- MMSC: Multimedia Messaging Service Center
- MMSproxy: Multimedia Messaging Service proxy server
- MMSport: Multimedia Messaging Service port number

The correct settings can always be found on the website of the provider concerned. In the script are the MMS settings of the two Dutch providers that we used here in the lab.

After making the appropriate settings we are ready for the first test. Open the terminal window on the Raspberry Pi and start with the command:

```
sudo su
```

This is necessary to obtains the rights to be able to use the Python GPIO-library, or in other words, to be able to control the I/O-pins of the RPi.
The script can now be started using:

```
python MMSautoSend.py
```

The script itself is relatively straightforward. The RPi sends AT-commands to the GSM-module, where each time the M95 is given sufficient time to process the command with time.sleep. It begins with a range of settings, including parameters for the camera and volume settings for the microphone and loudspeaker, which incidentally, may have to be adjusted later, depending on the environment in which the doorbell will be installed.

To keep things simple we have turned off the PIN-code protection of the SIM cards that we used. The easiest way to do this is to insert the SIM card temporarily into a mobile phone and then use the appropriate menu to disable the PIN-code protection. In the script there is a block of four lines of comments in which the PIN-code can be entered for a card which is protected. If necessary, adjust the PIN-code according to the card that is used and remove the '#' at the beginning of the lines `port.write` and `time.sleep`.

After that the scripts ends up in an infinite loop, which can only be interrupted with CTRL+C to exit the script. But normally the RPi will continually check whether an SMS was received for operating the door opener OR checks whether the doorbell was operated or not.

In the event of an SMS, the script first checks where the message came from. If it came from the correct smartphone (ThatNumber) then it also checks that the contents of the message corresponds to the password that is set in the constant Very_Secret and if that is the case, relay RE2 will be activated for one second. Of course, this time can be changed in the script should that be necessary.

However, the receipt of an SMS message is usually preceded by another event: a visitor rings the doorbell. The Raspberry Pi then immediately takes a picture and prepares an MMS-message that will be sent to your smartphone. In addition to the photo, the message also contains a small amount of text, including the phone number of the doorbell, which will appear as a hyperlink on your smartphone. Depending on the person waiting in front of your door, you can use this link to call the doorbell; the M95 will answer automatically, so that you can converse with the visitor. Once the conversation ends on the smartphone, the M95 will also 'hang up' and be ready for the next cycle of the bell ringing or the receipt of an SMS-message.

Building in

The mechanical workmanship is perhaps the biggest challenge for this project. The most ideal — and the safest — is to mount only the doorbell pushbutton, camera, microphone and speaker outside and keep the remainder of the electronics inside the house. In some cases you may be able to have the camera look outside through a window.

As already mentioned, in the script there are a few parameters to configure the camera, where specifically the orientation and brightness may have to be changed depending on the environment in which the camera will be installed. The resolution only affects the quality of the photo, keep this as low as possible so the size of the JPEG that is to be sent — and with that the time it takes to be transmitted — is within reasonable limits. While experimenting with these settings it will be best if you turn off the sending of MMS-messages in the script and only view the image file (test1.jpg) that is stored on the RPi to judge the quality. ⏮

(150400)

Web Links

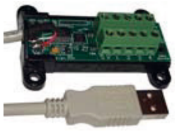
[1] www.elektormagazine.com/labs/gsm-breakout-board-150330

[2] www.quectel.com/UploadImage/Downlad/M95_Hardware_Design_V1.3.pdf

[3] www.raspberrypi.org/downloads/raspbian/

[4] www.elektormagazine.com/150400

Free
Front Panel
Designer

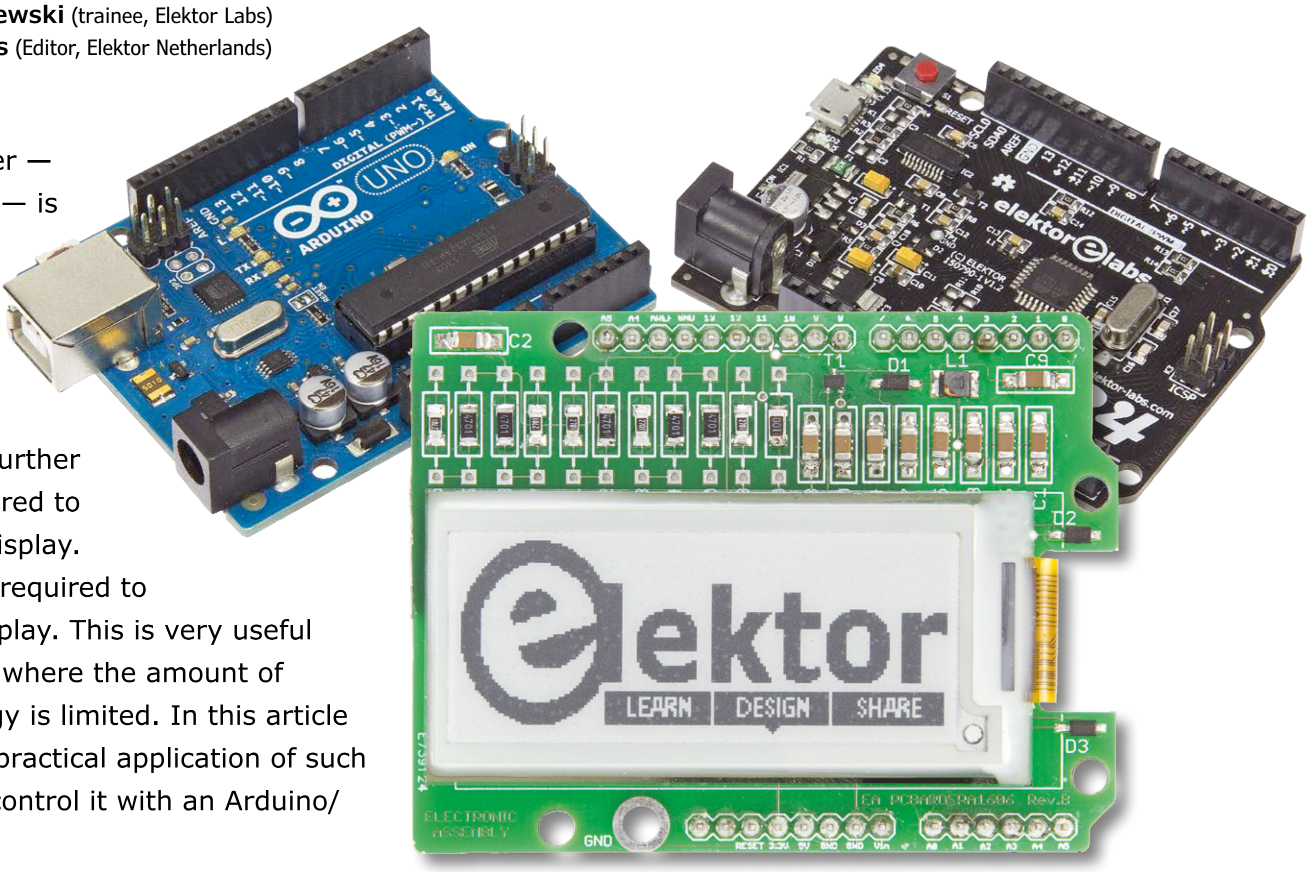


MOUSER
SEE THE NEWEST PRODUCTS
ADL5511
ANALOG DEVICES
SILICON LABS
TEXAS INSTRUMENTS
NXP
intel
Core i7

Using E-paper with the Arduino

A sketch for the EPA20-A

By **Niek Laskarzewski** (trainee, Elektor Labs)
and **Thijs Beckers** (Editor, Elektor Netherlands)



Electronic paper — a.k.a. e-paper — is an unusual type of display. Once a picture (or text) is displayed, no further energy is required to maintain the display. Energy is only required to change the display. This is very useful in applications where the amount of available energy is limited. In this article we describe a practical application of such a display and control it with an Arduino/Elektor Uno.

The e-paper display we're going to use is the EPA20-A from Electronic Assembly [1]. It has a resolution of 172 by 72 pixels and has an operating voltage of 3.3 V. Bear in mind that e-paper doesn't have a background light, just like 'real' paper so you can't read it in the dark — though it will be perfectly legible in full sunlight.

Requirements

We'll go in at the deep end and start with showing you how to get the display to work. The required Arduino libraries are freely available from [2]. Download the file 'Arduino meets EA EPA20-A.zip' and extract all the files. There will be three sub folders in the folder that was just created ('Arduino meets EA EPA20-A'). In the folder 'Arduino Library' is another folder called 'EPA20_A', which should be copied to the library folder of the Arduino development environment (free download from [3]).

In order to display an image you need to create a header file. This can be done with the program *BitMapEdit*, which is part of the *EA LCD-Tools* package (free download from [4]).

Schematic

The (very simple) schematic for the Arduino shield is shown in **Figure 1**. A booster circuit is built around T1/L1, which generates the voltage required by the e-paper from the 3.3V supply. The booster is driven by a controller that is part of the display module.

Resistors R1–R10 are configured as potential dividers, which limit the 5 V signals from the Arduino (Uno) to safe values. The circuit takes its supply from the 3.3 V supply of the Arduino.

Getting started

Before the functions of the added libraries can be initialized, they have to be defined. This is done by adding the line '`epa20 EPA20;`' *before* '`void setup()`' in the Arduino sketch. We can now initialize the display inside '`void setup()`' with the command '`EPA20.initialize(byte p_cs, byte p_si, byte p_so, byte p_clk, byte p_a0, byte p_res)`'.

The shield can be configured for either hardware SPI or software SPI. In this case we'll use hardware SPI. This is done by '`EPA20.initialize(10,0,0,0,9,8)`'. With the initialization complete, we should call '`EPA20.clear()`'. This function clears

the buffer of the display. It is also used before a new display is created.
The functions '`EPA20.string()`', '`EPA20.rectangle`' and/or '`EPA20.picture`' are used to show information on the display. When we want to show a new display we have to use the instruction '`EPA20.update()`', otherwise the changes won't be shown. As mentioned earlier, the display has a resolution of 172×72 pixels. This means that the coordinates must never be greater than these values. The coordinates start at the bottom-left of the display (0,0 is bottom-left and 171,71 is top-right).

Hello World

The command '`EPA20.string()`' is used to display a line of text. It is used as follows:

```
EPA20.string(byte x, byte y,
   const byte *font_adress, const
   char *str, byte fg_color, byte
   bg_color)
```

Where

- `Byte x`: the x-coordinate,
- `Byte y`: the y-coordinate,
- `Const byte *font_adress`: the type of font, with a choice of font_6x8, font_8x8, font_8x16, font_16x32nums,
- `Const char *str`: the text to be shown,
- `Byte fg_color`: the color of the letters, with a choice of BLACK, WHITE, GRAY1 en GRAY2,
- `Byte bg_color`: the background color, with a choice of BLACK, WHITE, GRAY1 en GRAY2.

For example: `EPA20.string(0, 0, font_8x16, "Hello world", BLACK, WHITE);`

`EPA20.rectangle(byte xs, byte ys, byte xe, byte ye, byte outline, byte filling);` creates a rectangle with an outline. This function has six variables:

- `Byte xs`: start position of the x-coordinate;
- `Byte ys`: start position of the y-coordinate;
- `Byte xe`: end position of the x-coordinate;
- `Byte ye`: end position of the y-coordinate;
- `Byte outline`: color of the outline (BLACK, WHITE, GRAY1, GRAY2);

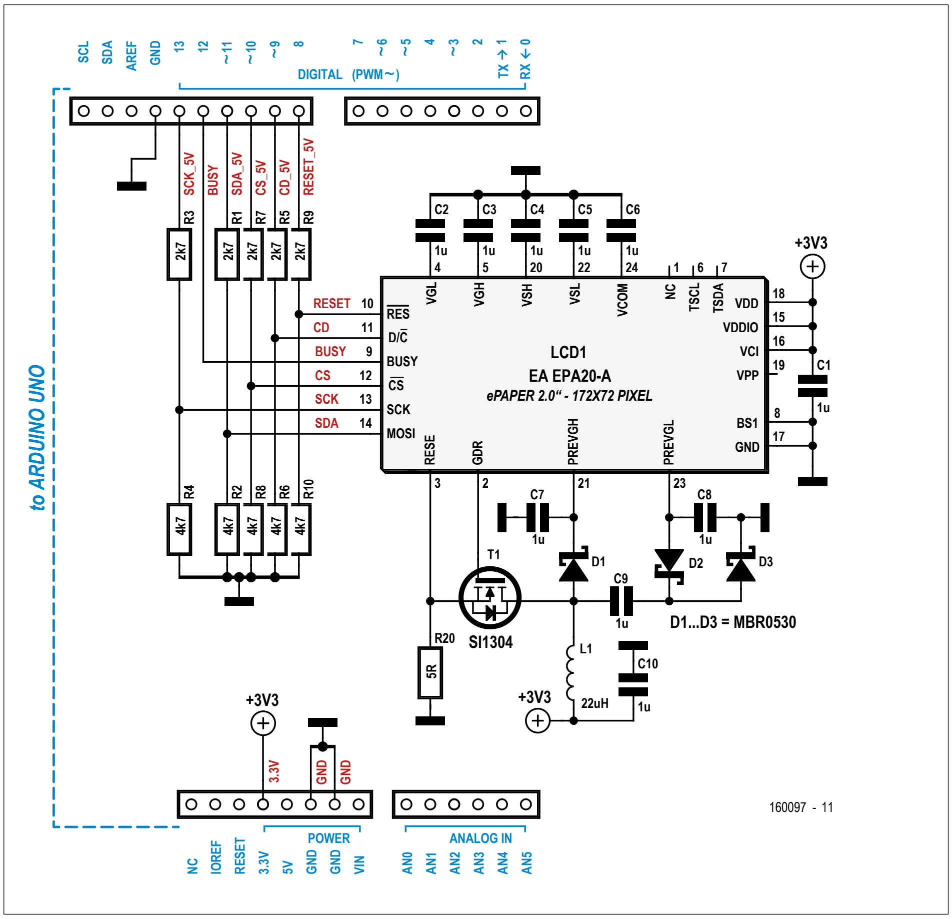


Figure 1. The circuit is fairly simple and clearly shows all the connections and the booster circuit.

- `Byte filling`: color inside the rectangle (BLACK, WHITE, GRAY1, GRAY2).

For example: `EPA20.rectangle(0, 0, 10, 10, BLACK, GRAY2);`

Graphical displays

`EPA20.picture(byte x, byte y, const byte *pic_adress, byte fg_color, byte bg_color);` shows a monochrome image with a maximum size of 172 by 72 pixels, where

Component List

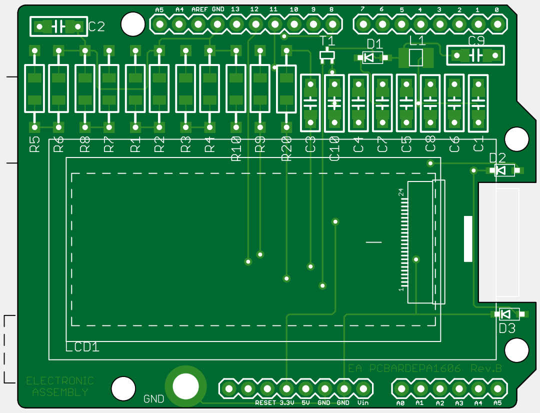
Resistors
R1,R3,R5,R7,R9 = 2.7kΩ
R2,R4,R6,R8,R10 = 4kΩ
R20 = 5Ω

Capacitors
C1–C10=1µF 50V, 1206

Inductor
L1 = 22µH 0.25A, SMD 1210

Semiconductors
D1,D2,D3 = MBR0530
T1 = SI1304

Miscellaneous
J1 = WF050-24T
Display = EPA20-A
Pinheaders, 0.1'' pitch, SIL: 10-pin, 8-pin (2 pcs), 6-pin



- `Byte x`: start point of the x-coordinate;
- `Byte y`: start point of the y-coordinate;
- `Const byte *pic_adress`: the name of the image;
- `Byte fg_color`: the foreground color;
- `Byte bg_color`: the background color.

Before an image can be shown, it must first be converted into a header file, as we mentioned earlier. This can be done using the program *BitMapEdit*. With this program you can create your own drawing, but you can also import an image. This image must not be greater than the screen, so the maximum is 172×72. The color depth of the image has to be black and white (refer to *Bitmap→color-depth→black and white (1bit)*).

Once everything is in order, you can export the image by clicking on *File→↖ Export*. It should be saved in the same folder where the Arduino sketch is. You should make sure that the *save as* type is set to 'C Header File vertical BLV (*.h)', then save the file.

Next, you'll have to open the header file of the exported image in a simple text editor, such as notepad or notepad++. In the header file is a line similar to the following: `unsigned char Image_example[Image_example_LEN] =`, where `example` is the name of the image. In this line you have to add `__attribute__((section(".progmem.data")))` *before* the '=' sign and after the final ']'. The line then becomes:

```
unsigned char Image_example[Image_
   example_LEN] __attribute__
   ((section(".progmem.data")))=
```

In the Arduino sketch you have to add the line `#include "example.h"`, where `example` is the file name of the image. When this image is used in the `EPA20.picture()function`, you should add '`Image_`' in front of the name (`example` becomes `Image_example`).
We can now display the image with an instruction like:

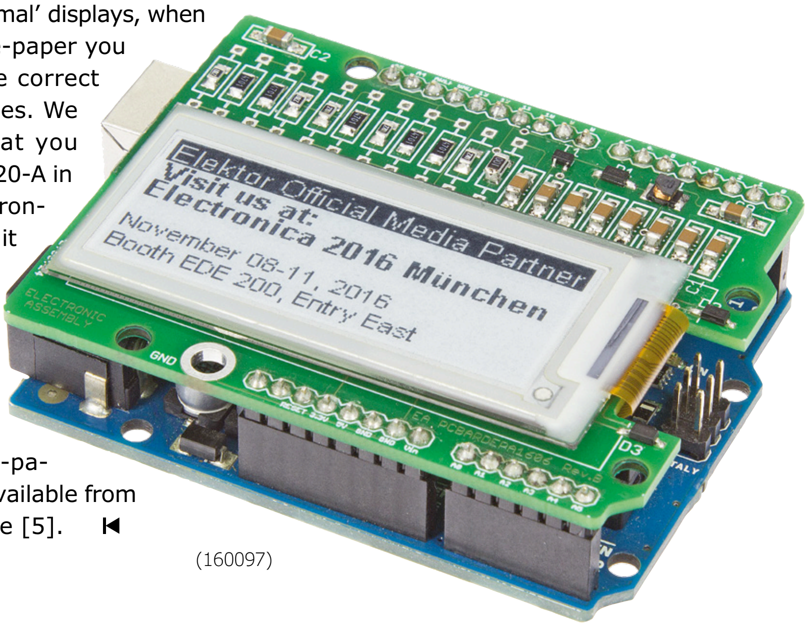
```
EPA20.picture(0, 0, Image_example,
   black, white);
```

Listing 1 shows an example of an Arduino sketch that displays an image.

Conclusions

Just as with 'normal' displays, when you work with e-paper you need to use the correct tools and libraries. We have shown that you can use the EPA20-A in an Arduino environment to provide it with a remarkable display. The Arduino shield for use with the EPA20-A (including the e-paper display) is available from the Elektor Store [5]. ◀

(160097)



Listing 1. An example of an Arduino sketch that displays an image.

```
#include <Arduino.h>
   #include <SPI.h>
   #include <epa20.h>
   #include <logo_ea.h>
   #include <font_6x8.h>
   #include <font_8x8.h>
   #include <font_8x16.h>
   #include <font_16x32nums.h>
   #include "example.h"
epa20 EPA20;
void setup()
{
     EPA20.initialize(10,0,0,0,9,8);   //initialize the screen
     EPA20.clear();
   }
void loop()
{
     EPA20.clear();  //clear the screen
     EPA20.picture(0,0,Image_example,BLACK,WHITE);
     EPA20.update(); //ensure that the screen is updated
   }
```

Web Links

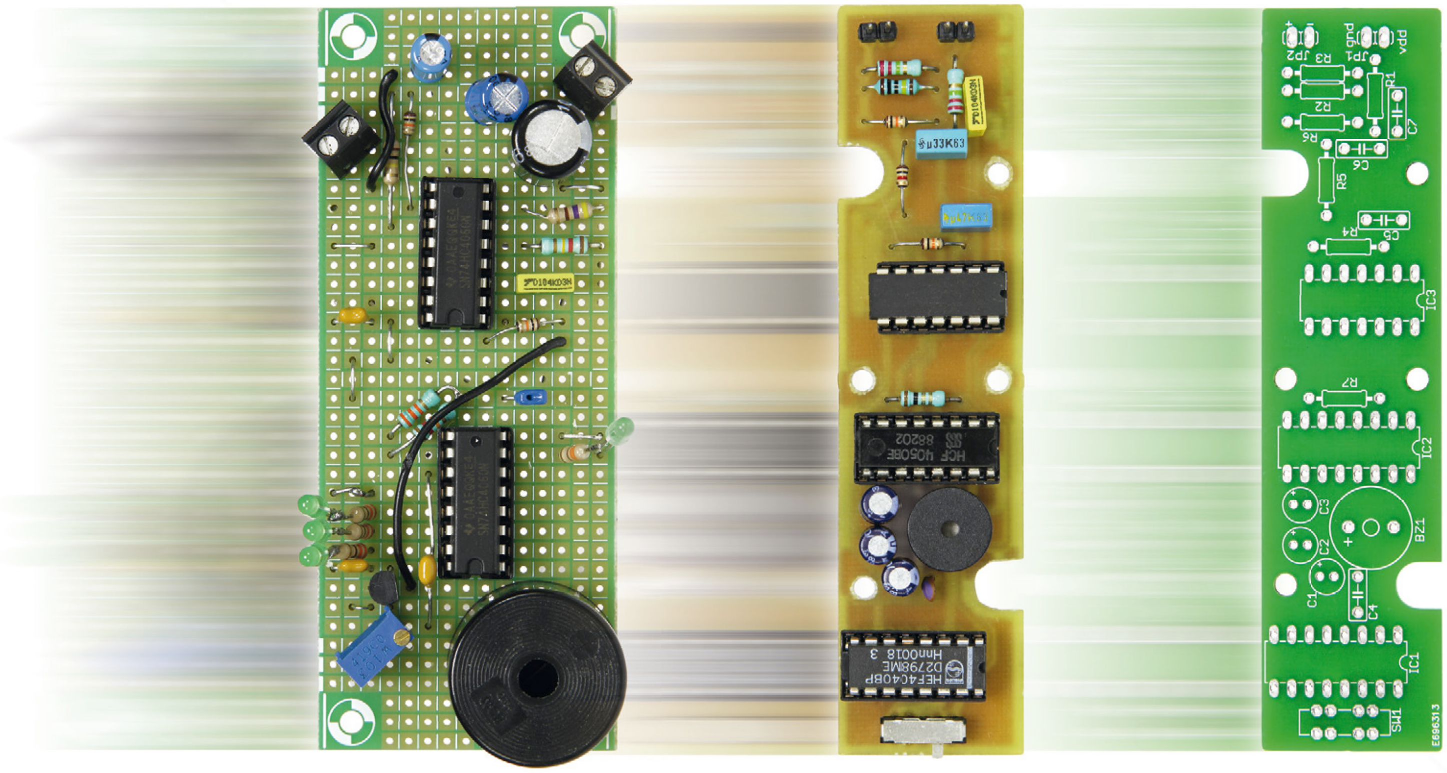
[1] Datasheet for the EPA20-A: www.lcd-module.com/fileadmin/eng/pdf/grafik/epa20-ae.pdf

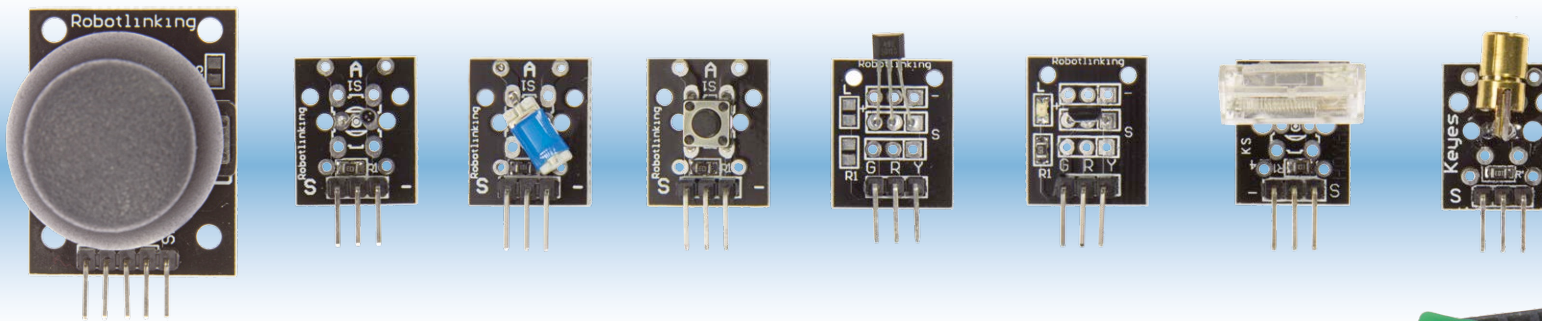
[2] Libraries: www.lcd-module.com/support/application-note/arduino-meets-epa-epaper.html?no_cache=1&sword_list%5B0%5D=arduino

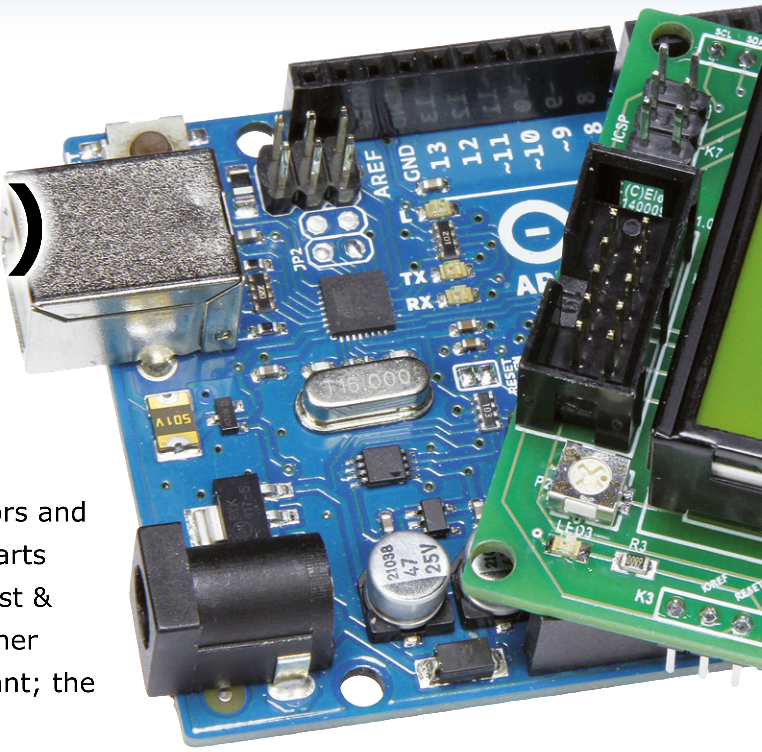
[3] Arduino development environment: www.arduino.cc/en/Main/Software

[4] BitMapEdit: www.lcd-module.com/support/driver-tools.html

[5] Project web page: www.elektor.com/160097







Sensors Make Sense (1)

For Arduino *et al.*

By **Burkhard Kainka** (Germany)

For some time now Elektor has offered a kit of 35 sensors and actuators [1]. It's mighty popular because with these parts you can produce all kinds of applications in fields like test & measurement, robotics or household automation. Whether you favor the Arduino or some other platform is irrelevant; the possibilities are endless.

With so many types of sensor (from joysticks to temperature and humidity monitors) and output devices (from relays to laser lights) it's no easy task to keep up with all the possibilities. This series of articles will make it easier for you by presenting practical applications. As our base point we'll use an Arduino Uno, which can be programmed both in Arduino C and in Bascom.

Overview: sensors and actuators

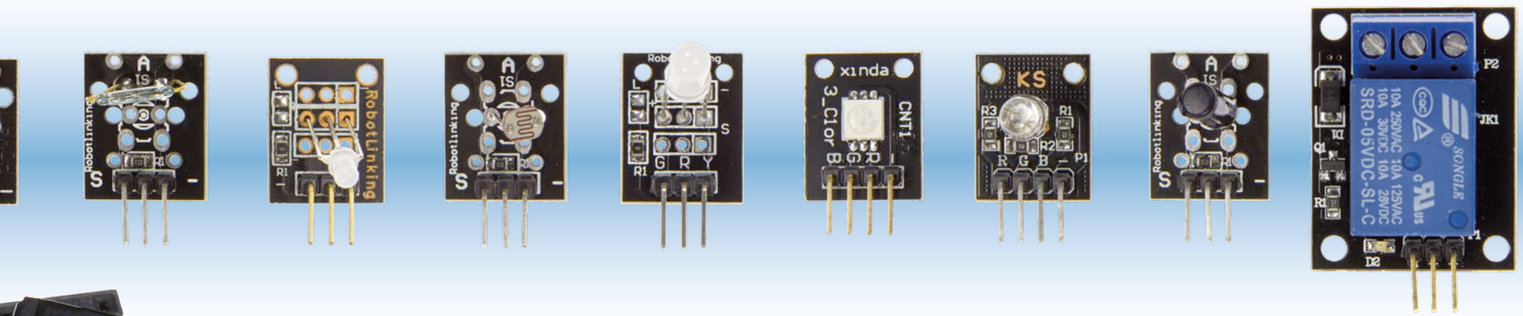
Analog sensors take center stage in this first part of the series. Sensors for light, temperature and magnetic field strength, together with a joystick with two potentiometers, are best examined alongside the appropriate actuators with which they work hand in hand. Here is where relays, dual-color LEDs, RGB LEDs and laser lights come into action. By way of comparison, we've checked out some digital sensors too.

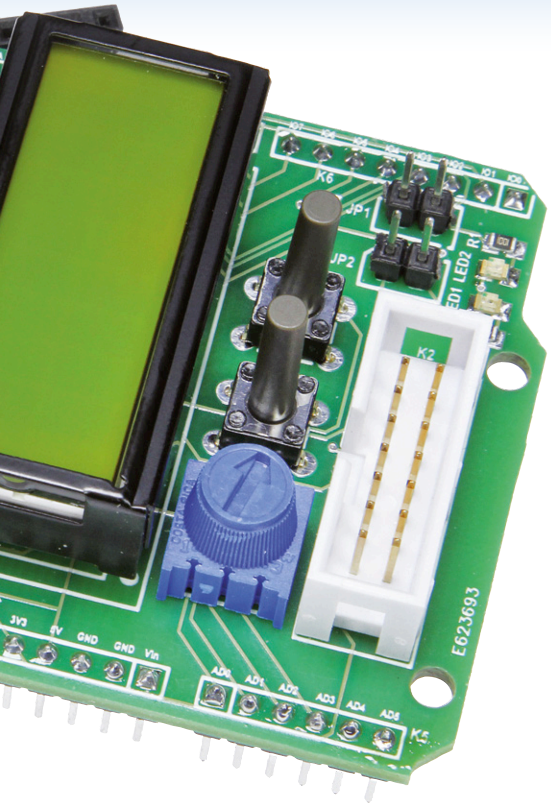
All the sensors and actuators mentioned here have their schematics and connection details shown in **Figure 1**. The designations correspond with those used in the sensor kit. The legends on the connections match those on the PCB too. In this way you can identify what you're looking at and get your bearings. On the right, next to the connection pins, you have in each case suggested connections to the Arduino Uno, as used in the sample programs.

If you take a close look at the multiplicity of sensors in the kit, you soon see that some of them use the same PCBs. This is not just cost-effective but also logical, because they share the same basic principles. A good example of this is the matching circuitry of the NTC temperature sensor (*Analog Temp*) and the LDR sensor (*Photoresistor*) [NTC = negative temperature coefficient; LDR = light-dependent resistor]. Both are sensors that vary their resistance according to the value measured (temperature or brightness). Usually they are both used in conjunction with a voltage divider or a fixed resistance.

The voltage divider is particularly simple, yet despite this, it also has the advantage of accuracy. This is because whenever the same voltage is used both for the voltage divider and as reference for the A-D converter, errors arise that can be traced back to inaccurate voltage. If you use the Arduino's 5-V connection, you have at this point a truly exact 5 V (when used with an AC power supply) but major variations can occur when you take power from a USB connection. However, the A-D converter provides the right result regardless of accurate supply voltage, as the value arises out of the relationship between the measured voltage and the reference. In this way you register the resistance ratio in the sensor circuitry direct, so to speak.

The way in which the two sensors are connected means we always get a falling voltage with more light and higher temperature. When programming we need to take this into account. But in this case it is also possible to transpose GND and VCC, in order to achieve a rising voltage when measured values increase.





All the same, perhaps you should better not do this, because other sensors cannot tolerate this and the diversity of sensors means that any potential source of errors must be avoided. For example the analog magnetic sensor (*Analog Hall*) is, in principle, connected exactly in this manner, albeit with an additional integrated measurement amplifier that requires a supply voltage of the correct polarity. Note the differing connections of the LDR and the NTC!

Looking at the digital sensors, it's readily apparent that they use the same PCB as the LDR and the NTC sensor. This holds good for the pressbutton switch (*Button*) and the reed switch (*Mini Switch*). Here again we find a sort of 'voltage divider' made up of a switch and a 10-kΩ resistor. Because this switch, according to its state, is either an infinite resistance or none at all, the voltage measured is either 5 V or 0 V. Consequently you can also interrogate this sensor to obtain a voltage measurement on an analog input, when

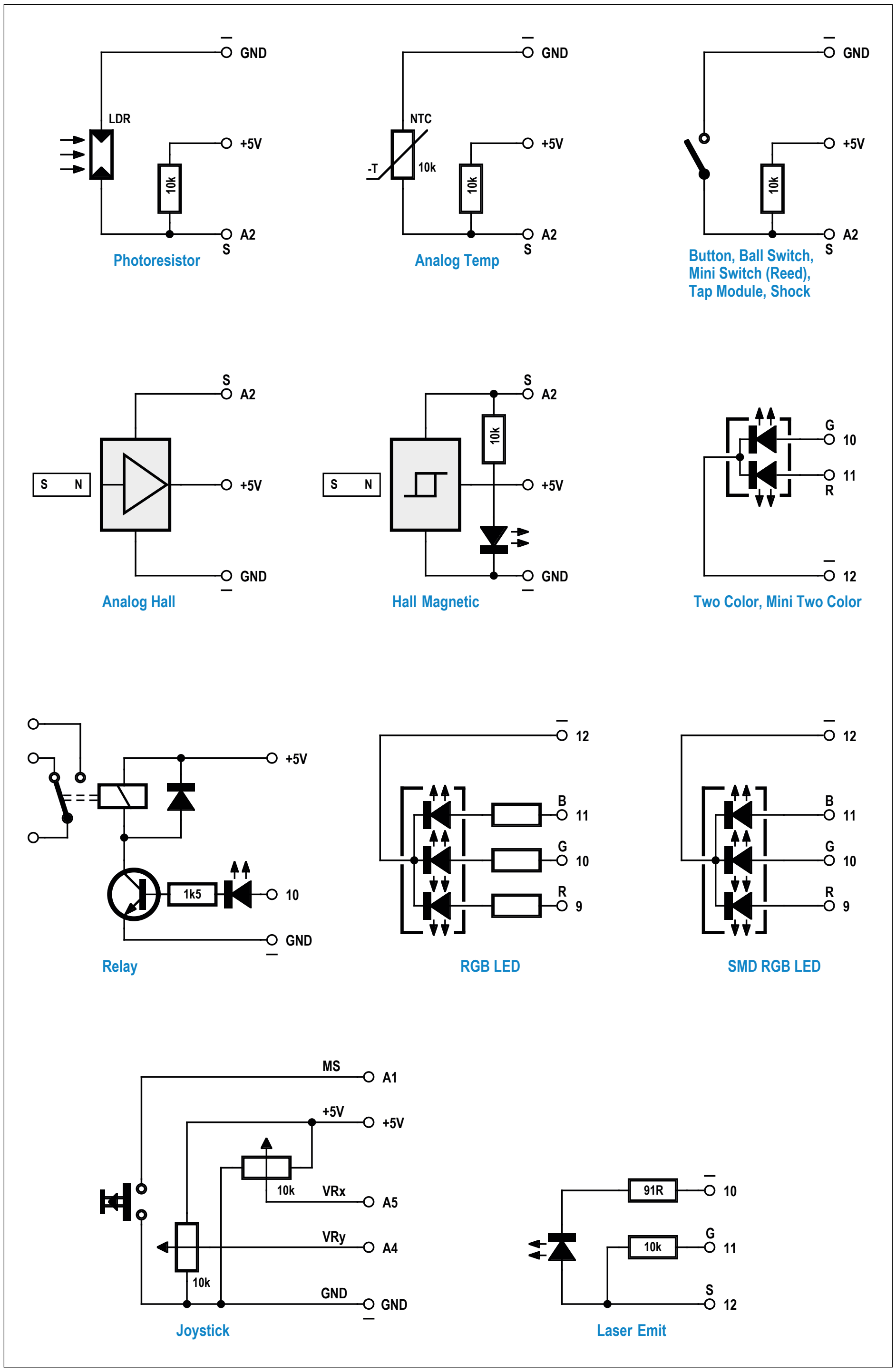


Figure 1. Schematics and connections for the sensors and actuators used.

in this case a digital input would also suffice.

Using analog sensors is generally quite straightforward: you measure a voltage with the A-D converter and then analyze or display it. For doing this you will find plenty of points of reference in our Bascom teach-in course from 2014 [2]. The way you use the Arduino Uno with Bascom is described in detail there. You can either control the Arduino Bootloader direct or flash the customized MCS Bootloader into the controller. Incidentally, during the preparation of this series of articles we used two Unos. One of them was left untouched, whilst the other was equipped with the MCS Bootloader, which worked trouble-free with Bascom.

Once again we set the Elektor Extension Shield [3] to work, exploiting its display and control functions (such as switches, LEDs and text display) to further expand our range of possibilities. This time all of the sample exercises are presented in Arduino C. This enables everybody to work effortlessly in their own preferred environment. You do nevertheless need to keep an eye on the connector assignment (configuration) of the Extension Shield, which is shown again in **Figure 2** as a reminder:

- Port B (Arduino pins 8 to 13): unassigned;
- Port B.2 (Arduino pin 10) LED2 via jumper, also used for PWM output;
- Port B.5 (Arduino pin 13) LED on the Arduino;
- Port D (Arduino pins 0 to 7): fully assigned to the UART and LCD;
- AD0 and AD1: Switches S1 and S2 versus GND, also usable as analog inputs;
- AD2: LED1 via jumper, otherwise unassigned (preferred input for analog sensors);
- AD3: Pot on the LCD shield;
- AD4 and AD5: Unassigned, but frequently used for the I^{2}C interface.

Do take care when using AD0 and AD1! Because the pressbuttons here are integral to the board and cannot be separated from it, it is possible to accidentally short-circuit the voltage being measured. Many sensors just ignore this but some react sulkily. Particularly at risk is the joystick fitted with two pots. If you connect the wiper of a pot (in this case via one of the pressbuttons) to GND at the same time as an external contact is at +5V potential, and you then rotate the wiper in the direction of this pin, the resistance becomes ever smaller and the current flow ever greater. A full short circuit would be only half as bad, as the Arduino has a self-resetting fuse. But there is a point near the end stop of the pot where it will get extremely hot and in this situation many a pot has given up the ghost in a puff of smoke. Consequently you will do better to steer clear of AD0 and AD1, also AD3 too, because there's an internal pot on the Shield in parallel and in the worst-case scenario you could end up destroying two pots. You'll do better to connect the joystick to AD4 and AD5, assuming you can manage without I^{2}C.

Displaying voltages in Bascom

Now comes our first sample program (all of these programs can be downloaded at [4]). A simple voltage display is handy for testing different sensors. Here we will use analog input ADC2. The jumper contact to LED1 on the Shield needs to be open (i.e. disconnected) for this. The raw data in the range 0

Listing 1. Voltage readout.

```
'----------------------------------------
'ADvolt.BAS    0...5 V
'----------------------------------------
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600
$hwstack = 16
$swstack = 16
$framesize = 64


Dim D As Word
Dim U As Single
Dim S As String * 10

S1 Alias Pinc.0
S2 Alias Pinc.1

Config Adc = Single , Prescaler = 64 , Reference = Avcc
Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6
  = Portd.6 , Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2
Waitms 50
Portc.0 = 1                         'Pullup
Portc.1 = 1                         'Pullup
Config Portb = Output


Cls
Cursor Off

Do
  D = Getadc(2)
  U = D * 5.0
  U = U / 1023
  U = U * 1000
  D = Int(u)
  'Print D
  S = Str(d)
  S = Format(s , "0.000")
  Print Chr(13);
  Print S ; " V     ";
  Locate 1 , 1
  Lcd S ; " V    "
  Waitms 500
Loop
```

to 1023 provided by the A-D converter is converted into voltages, output in serial format and displayed on the LCD.
A special feature of the Bascom program (**Listing 1**) is its serial output, which is customized specially for the Bascom Terminal. A frequent distraction on the Terminal is new lines of output constantly appearing one after another, caused by every print instruction being ended with Carriage Return (CR, ASCII 13) and Line Feed (LF, ASCII 10). In fact you can suppress these control characters with a semicolon at the end of the print instruction and by sending just a CHR(13) before the instruction. Each new output then appears on the same line, resulting in a static display (**Figure 3**).

Another special feature is the use of the Format function. An integer D is first transformed into a string S. Then we establish an edit mask '0.000', so that the last three places are placed behind the decimal point. To ensure everything works properly, a voltage like 1.234 V needs first to be multiplied by 1000, so that you end up outputting 1234 mV.
The string prepared in this manner is sent serially to the Terminal and also displayed on the LCD. If you wish to use the program without using the Shield, you can do this without any modification, as outputs to the LCD do not wait for any acknowledgment and simply disappear into the void if nothing is connected. Conversely, once you have loaded the program, you can also use it without needing the PC any more and just look at the LCD. If you feel like testing the program first without a sensor, simply bridge a wire between AD2 and AD3. Then it's up to the pot on the Shield, so that you can set any voltage of your choice between 0 V and 5 V and have it displayed. The universal voltage measurement device can now be used with a variety of sensors.

- The NTC sensor indicates a voltage of 2.5 V at 25 °C and a falling voltage with rising temperature. A still more exact analysis is displayed.
- The LDR sensor outputs a broad voltage range between almost zero (very bright) and almost 5 V (very dim).
- The pressbutton and the reed switch indicate 5 V in idle state. When you press the button or hold a strong magnet next to the reed switch, the voltage drops to 0 V.
- The analog Hall sensor delivers around 2.5 V in standby mode. Rotating it spatially makes no definitive change caused by the Earth's magnetic field. Bringing powerful magnets close to it cause readings of between around 1 V and 4 V, according to their direction.
- The digital Hall sensor is connected in the same way but includes a Schmitt trigger. In idle mode it measures over 3 V but when activated by a magnet approaching on the correct direction, the reading is almost zero.

If you need to process the measured data further, you can open a Log file in the Bascom Terminal and close this again at the end of measurement process. In this case it is better to use the commented output in the program 'Print D'. D employs whole integers and indicates the voltage in mV, so that no errors can arise with decimal points. All data collected is then stored and remains available for subsequent processing, for instance in a spreadsheet program. You can also produce charts and diagrams in this way (**Figure 4**).

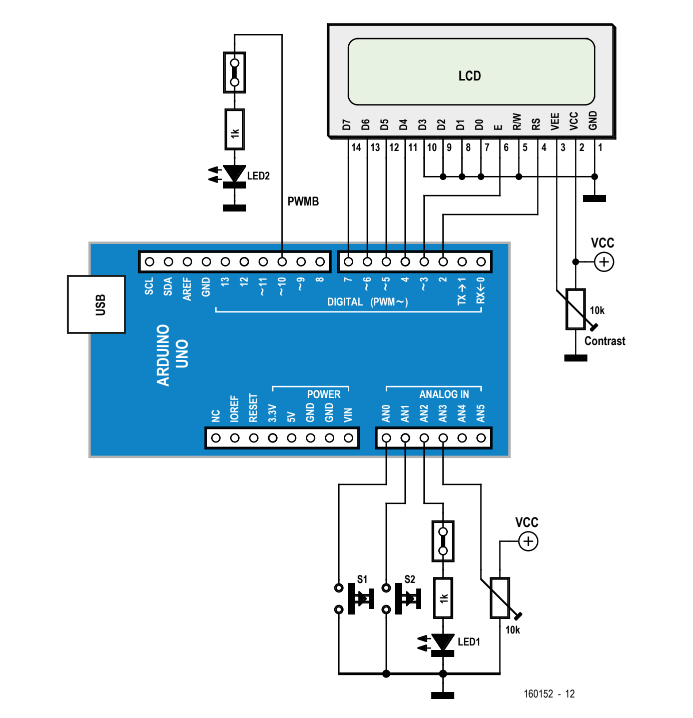
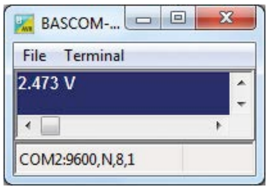


Figure 2. The Extension Shield supplements the Arduino Uno with LEDs, pushbuttons, a pot and a display.

Measuring voltages with Arduino

Programming in C with the Arduino IDE is not much different from programming in BASIC with Bascom. The universal-use program *VoltageAD2.ino* (**Listing 2**) again converts the raw



```
2.473 V
```

COM2:9600,N,8,1

Figure 3. Voltage readout in Bascom Terminal.

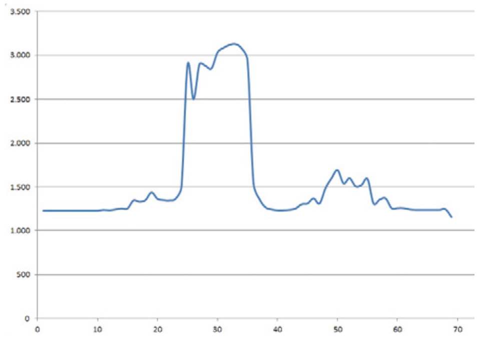


Figure 4. Excel graphic of a brightness curve.

output (value) from the A-D converter into the voltage in mV. At the same time this sample program shows how the LCD on the Elektor Shield is controlled to output the voltage in volts on the LCD. Without any additional formatting, the result is shown as two places after the decimal point.

You can see the measurement readings sent (voltages in mV) on the serial monitor (**Figure 5**). Starting with version 1.6.8, the Arduino software has been enhanced to handle serial plotters (**Figure 6**), enabling them also to receive and display simple sequences of numbers. In this way you can create a continuous diagram like on a plotter with an endless roll of paper. The range of measurement adjusts itself continually to the data sent. The smallest measurement range spans from −10.0 to +10.0.

Listing 2. Voltage measurement and LCD readout using the Arduino.

```
//VoltageAD2 0...5000 mV at AD2
#include <LiquidCrystal.h>
int sensorPin = 2;
int value = 0;
float voltage;
LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
}
void loop() {
  value = analogRead(sensorPin);
  voltage = value;
  voltage = voltage * 5000 / 1023;
  value = voltage;
  Serial.println(value);
  lcd.setCursor(0, 0);
  lcd.print(voltage/1000);
  lcd.print (" V   ");
  delay(500);
}
```

Listing 3. Analyzing NTC temperatures.

```
Dim D As Integer
Dim B As Integer
Dim R As Single
Dim T As Single
Dim S As String * 10
…
Do
  D = Getadc(2)
  B = 1023 - D
  R = D / B          'R = 1 @ 10k
  T = Log(r)
  T = T / 3900       'B25/85=3900
  T = T + 0.0033557  '1(273+25)
  T = 1 / T
  T = T - 273
  T = T * 100
  D = Int(t)
  S = Str(d)
  S = Format(s , "+0.00")
  Print Chr(13);
  Print S ; " " ; Chr(248) ; "C   ";
  Locate 1 , 1
  Lcd S ; " C   "
  Waitms 500
Loop
```

Listing 4. Measuring temperatures with the Arduino.

```
//NTCtempAD2 10 k NTC at AD2
#include <LiquidCrystal.h>
int sensorPin = 2;
int relaisPin = 10;
int value;
float fvalue;
float resist;
float temp;
LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(relaisPin, OUTPUT);
}
void loop() {
  value = analogRead(sensorPin);
  fvalue = value;
  resist = fvalue / (1023-fvalue);
  temp = 1/(log (resist) / 3900 + 0.0033557) - 273;
  Serial.println(temp);
  lcd.setCursor(0, 0);
  lcd.print(temp);
  lcd.print (" C   ");
  if (temp > 25 ) digitalWrite (relaisPin, HIGH);
  if (temp < 20 ) digitalWrite (relaisPin, LOW);
  delay(500);
}
```

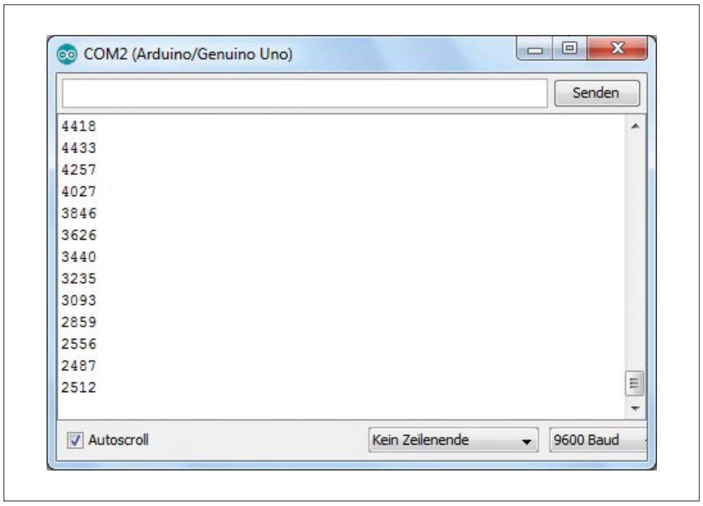
Temperature measurement using NTC sensors

The nominal resistance of NTC sensors is quoted for a temperature of 25 °C. Additionally a so-called ‘B value’ is stated, which defines the gradient of the sensor’s characteristic curve. The variation in resistance proceeds to a close degree exponentially according to the change in temperature. This means that the voltage change is sufficiently large as to provide very good resolution without the need for a measurement amplifier. Precision sensors are manufactured mainly with a nominal resistance of 10 kΩ and a *B* value of 3900. With this information we can evaluate temperatures (**Listing 3**). This is a two-step process. First we calculate the resistance *R* of the sensor from the measured value *D*, i.e. raw data in the range 0 to 1023. In the second step we determine from this the temperature in degrees using logarithms and further stages of calculation. The result of the calculation also provides a remarkably accurate temperature display without calibration (**Figure 7**). These sensors normally have an accuracy of 1%, meaning the maximum variation is less than 1 K. To round off the project, we’ll again create a static display in the Terminal with optional simultaneous output to the LCD. No deception is intended in matters of accuracy; we are not boasting to display hundredths of a degree but to provide a better view of small temperature variations. One A-D step stands for around 0.1 K at average temperatures. At very low or very high temperatures the resolution becomes coarser.

In Arduino C the conversion process (**Listing 4**) functions, in principle, exactly as in Bascom. The most obvious difference is that in C you can write the entire calculation on one line, whereas Bascom and other dialects of BASIC observe the restriction that only one arithmetic step can appear on a given line. The Arduino program includes yet another useful enhancement with control for the relay module in the sensor kit. The relay operates with 5 V and is switched via a driver transistor. The base resistance of 1.5 kΩ along with the series-connected status LED is applied direct to a Portpin. Here Pin 10 (B.2) is selected as output, as this enables LED2 on the Shield to be controlled at the same time. The relay is now used in conjunction with the NTC sensor for operating a ventilator fan. When the temperature rises above 25 °C, the relay switches on the fan. If the temperature drops below 20 °C, it is switched off again. This gives us a two-setpoint controller with a hysteresis of 5 degrees.

RGB LED with joystick control

The sensor kit contains two RGB LEDs, one in standard LED form with three dropper resistors (150 Ω each), and one SMD version without any dropping resistors whatsoever. You often hear the claim that it’s fine to connect LEDs to batteries or Ports without using dropper resistors (“No problem. I checked and nothing nasty happened.”) But doing this clearly exceeds the threshold values of the LEDs and the microcontrollers, and overloads of this kind reduce the LEDs’ life span. It’s far better using dropper resistors of 150 Ω, which at 5 V will limit the current to 20 mA. If you want to avoid every risk, you will also connect these series resistors externally to the SMD LED. But there’s yet another compromise, which will certainly not win any design prize, that still needs addressing. In this we connect the common (shared) cathode of the RGB LED not to GND but to a Portpin that is switched low. The current is then limited by the Port’s internal resistance. Each Portpin has an internal resistance of around 20 Ω. Because an LED is connected to two Portpins, we have effectively a series resistance of 40 Ω. With an LED voltage of 3 V and an operating voltage of 5 V the LED current is 50 mA. Oh well, it still works. And if all three LEDs are switched on, there’s a greater voltage drop to be borne by the Portpin at the cathode. A voltage of 1.2 V



```
4418
4433
4257
4027
3846
3626
3440
3235
3093
2859
2556
2487
2512
```

Figure 5. Measured values on a serial monitor.

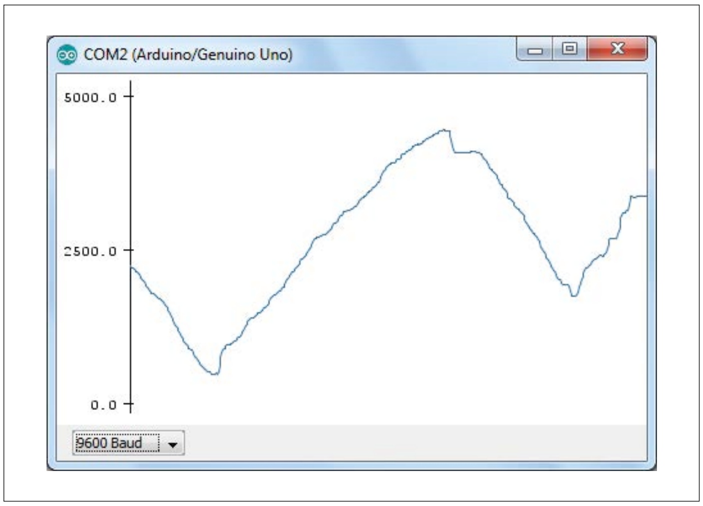
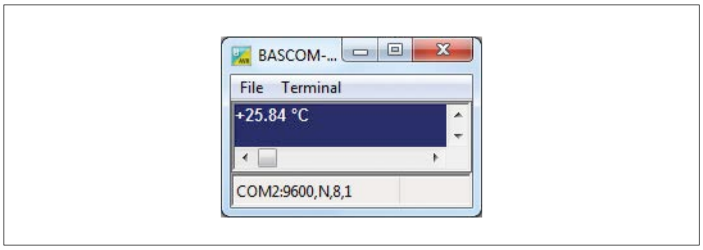


Figure 6. Measured values on a serial plotter.



```
+25.84 °C
```

COM2:9600,N,8,1

Figure 7. Temperature display using an NTC sensor.

Listing 5. RGB control with the joystick.

```
//JoystickRGB
#include <LiquidCrystal.h>
int joyX = A4;
int joyY = A5;
int red = 9;
int green = 10;
int blue = 11;
int cathode = 12;
int value;
int midX;
int midY;

LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  midX = analogRead(joyX);
  midY = analogRead(joyY);
  pinMode(cathode, OUTPUT);
  digitalWrite(cathode, LOW);
}

void loop() {
  value = analogRead(joyX);
  value = constrain(value, 35, 980);
  lcd.setCursor(0, 0);
  lcd.print(value);
  lcd.print("    ");
  if (value >(midX+200)) {
    analogWrite(red , 0);
    analogWrite(green , 0);
    analogWrite(blue , 0);
  }
  if (value <(midX-2)) {
    analogWrite(blue, (midX - value)/2);
  }
  value = analogRead(joyY);
  value = constrain(value, 35, 980);
  lcd.setCursor(0, 1);
  lcd.print(value);
  lcd.print("    ");
  if (value > midY) {
    analogWrite(red, value -midY);
  }
  if (value > (midY+2)) {
    analogWrite(red, (value - midY)/2);
  }
  if (value < (midY-2)) {
    analogWrite(green, (midY - value )/2);
  }
  delay(500);
}
```

Listing 6. Excerpt from the program *JoystickRGB.bas*.

```
Do
  D = Getadc(4)
  Locate 1 , 1
  Lcd D ; "  "
  If D > 980 Then D = 980
  If D < 35 Then D = 35
  D = D - 2
  If D > Midx Then
     Pwm = D - Midx
     Pwm = Pwm / 2
     Print Pwm
     Pwm = 200 - Pwm
     Print Pwm
     If Pwm < 0 Then Pwm = 0
     If Pwm < Red Then Red = Pwm
     If Pwm < Green Then Green = Pwm
     If Pwm < Blue Then Blue = Pwm
  End If
  D = D + 4
  If D < Midx Then
     Pwm = Midx - D
     Pwm = Pwm / 2
     If Pwm > Red Then Red = Pwm
  End If
  D = Getadc(5)
  Locate 2 , 1
  Lcd D ; "  "
  If D > 980 Then D = 980
  If D < 35 Then D = 35
  D = D - 2
  If D > Midy Then
     Pwm = D - Midy
     Pwm = Pwm / 2
     If Pwm > Blue Then Blue = Pwm
  End If
  D = D + 4
  If D < Midy Then
     Pwm = Midy - D
     Pwm = Pwm / 2
     If Pwm > Green Then Green = Pwm
  End If

  Pwm1a = Red
  Pwm1b = Green
  Pwm2a = Blue
  Waitms 10
Loop
```

is measured here, which points to a total current of 60 mA. In fact only 40 mA is permitted for an individual Portpin and 60 mA is more than this, although not enough to expect any harm to occur. The total load for all Ports together is 200 mA. The Arduino program *JoystickRGB* (**Listing 5**) uses the PWM outputs 9, 10 and 11 (B1, B2, B3) for controlling individual LEDs. The common cathode goes to Pin 12 (B4), configured as an output.

The intensity or brightness of each individual color can now be controlled with the joystick. For the reasons mentioned above, inputs A4 and A5 are used for polling the two pots. The control function uses the joystick in a way in which each axis is divided in two halves. Right, left, up, down together make four functions, so that as well as the three colors, we can also control the total brightness. In this case winding back to zero means every light goes out. In use it works like this: each color is adjusted for the desired intensity. If you release the control knob rapidly back to its rest position, the last value remains in force, because a new measurement takes place only every 500 ms. The fourth channel (down) switches all the LEDs off. At program start-up the neutral center positions *midX* and *midY* are measured. A certain variation up to 30 steps from the center of the measurement range (512 = 2.5 V) is normal and must be taken into account in the program. Helpful in this respect is the constrain(value, 35, 980) function, which prunes the range of values within 35 up to 980. Another thing to note in the control process is that each PWM output is controlled within the range 0 to 255, with half of the pot's range covering double the value range. Finally you need to allow some tolerance when comparing whether you are measuring above or below the center position.

Laser light control

The program also lends itself to testing dual-color LEDs. You can then adjust red and green separately and mix them together appropriately. Even the 'laser' can be controlled without difficulty using the same software. You can make adjustment to any desired degree of brightness and because the PWM signal produces rapid switching on and off, it is remarkably easy to 'paint' the walls with a kind of oscillogram effect. By varying the beam with suitably rapid motion you end up seeing red lines and gaps that demonstrate the PWM ratio. By default the Arduino software uses a low frequency to enable the switching operation to dissolve without difficulty. Contrastingly, the Bascom example uses the maximum PWM frequency, so that you would need an oscilloscope to recognize or distinguish the PWM signals.

The equivalent program in Bascom is called *JoystickRGB.bas* (**Listing 6**) and differs in its method of operation. The joystick movements assigned to colors drive the relevant LED only in the 'brighter' direction. Accordingly the maximum value is used for the actual value and setting in every case. As a result the measurement loop runs very quickly, because it no longer has to worry about returning back to the zero position. The control thus becomes more fluid and more delicate. At the same time the 'darken' function is now infinitely variable. Altogether this produces excellent adjustability of any desired color mix.

(160152)

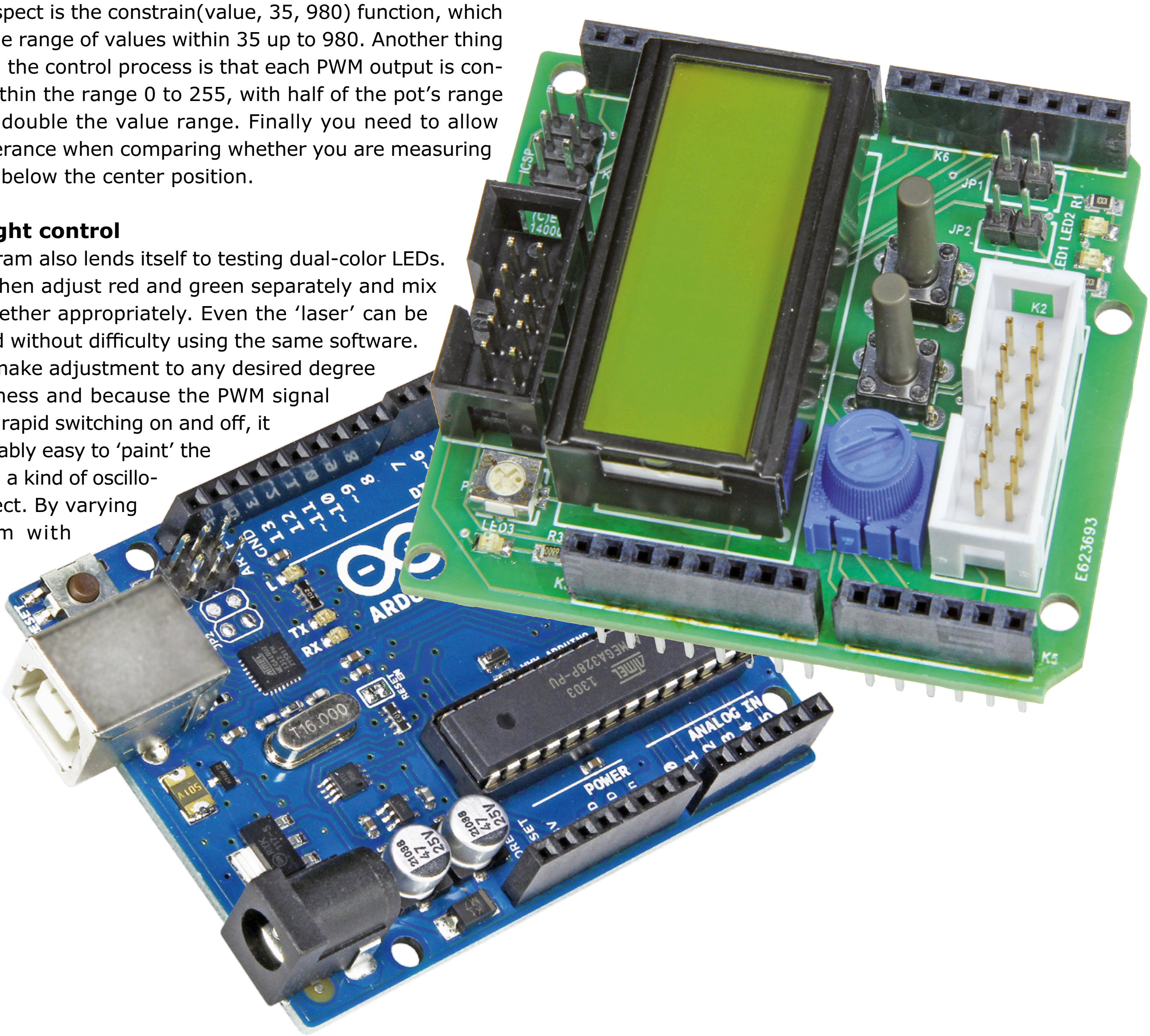
Web Links

[1] www.elektor.com/arduino-sensor-kit

[2] www.elektormagazine.com/120574

[3] www.elektormagazine.com/140009

[4] www.elektormagazine.com/160152



Wireless Power Converter

In cooperation with Würth Elektronik, in this article we present a pair of modules for high-efficiency wireless power transfer with a maximum capacity of 50 watts over distances up to a few centimeters. This set is available at a special price to enable Elektor community members to try out this highly promising technology.

By **Würth Elektronik, Ton Giesberts** and **Harry Baggen** (Elektor Labs)

Wireless transmission of power through the air is a very attractive idea. A lot of research has been carried out in this area in recent years, and the technology has evolved to the point that it is already being used commercially, for example in smartphone charging stations. However, the scope of potential applications is naturally much larger, particularly with power

Special offer for Elektor readers

In cooperation with component specialist Würth Elektronik, Elektor has put together a kit containing all the components and PCBs necessary to build a complete transmitter and receiver. Würth is sponsoring both PCBs and all components made by Würth in this kit, including the relatively expensive transmitter and receiver coils. Thanks to this sponsoring, for a short time we can offer Elektor readers a complete transmitter/receiver kit with a value of about €150 for **just €49.95** (plus shipping costs) or the equivalent in US$ or £. Please note that this is a limited-time offer and the number of kits available is restricted, so be sure to order your kit now in the Elektor Store.

Experience with soldering SMDs is necessary for building the transmitter and receiver boards.

Convey 50 watts through the air

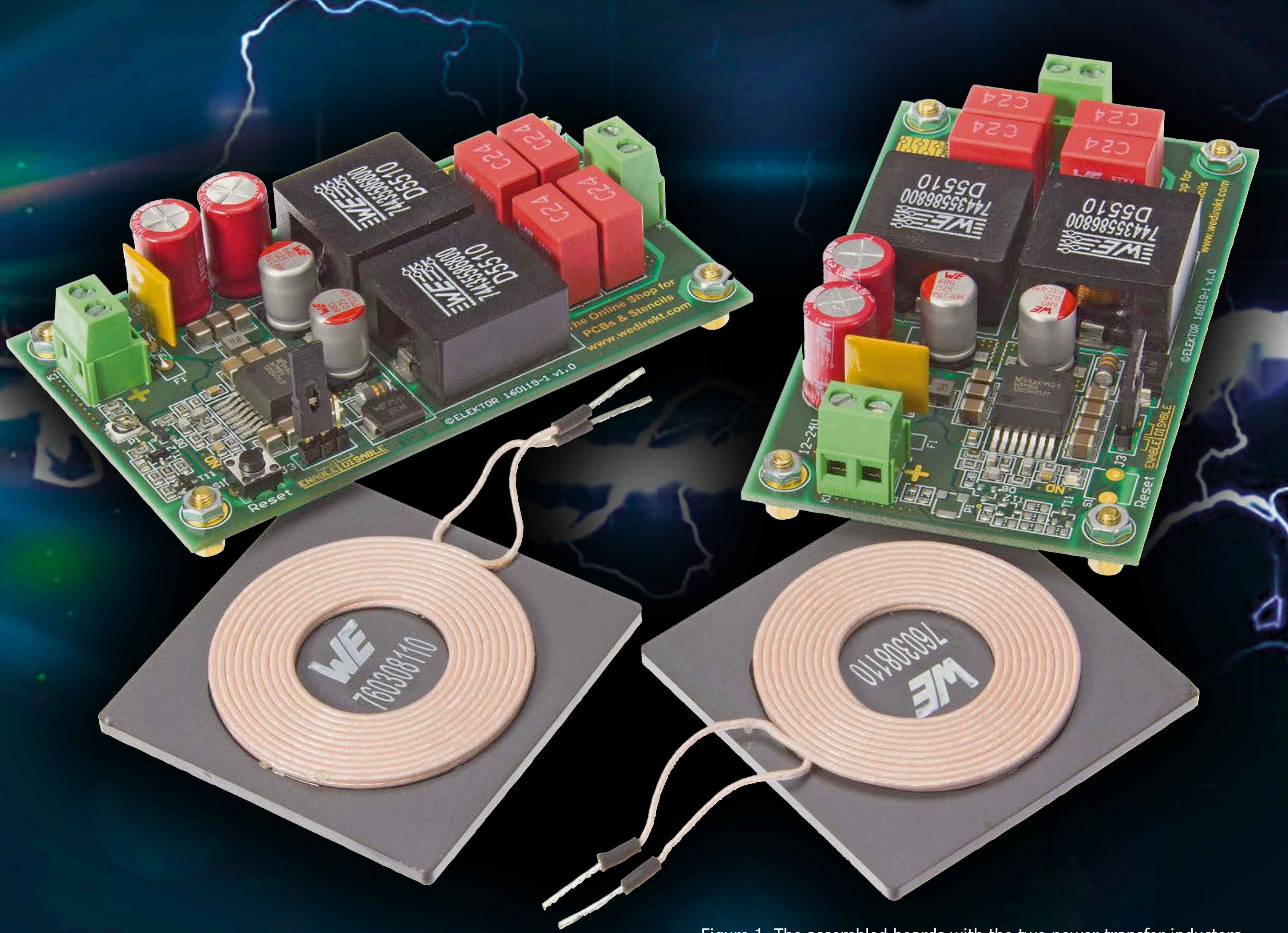


Figure 1. The assembled boards with the two power transfer inductors.

levels greater than just a few watts. That can be very advantageous in industrial and medical scenarios, for example when very high isolation is required or mechanical contacts and slip rings must be avoided.

Various manufacturers have developed specific components for wireless power transmission. One of the pioneers in this field is component specialist Würth Elektronik, which has a broad range of coils for wireless power applications in its product line. Würth also offers a variety of application circuits to help interested electronics enthusiasts get started and gain experience with this relatively new technology and the necessary components.

In cooperation with Elektor, Würth has taken the initiative to increase the awareness of wireless transmission amongst the Elektor community by means of the project described here, consisting of a wireless transmitter and receiver set for power levels up to 50 W. This is clearly an entirely different class of wireless power transmission than what is possible with wireless chargers compliant with the Qi standard, which have a maximum power of 15 W (most phone chargers are limited to 5 W).

To help you get hands-on experience with this technology, we have put together a kit containing all the components and PCBs necessary for a transmitter/receiver set. The PCBs and most of the components are sponsored by Würth, enabling us to offer this kit for less than €50 (in other words, just €1 per watt). You can experiment to your heart's content with this kit. However, we have to point out that the number of kits is limited.

Resonant circuit

How does wireless power transmission work? Actually it's very simple: two coils located a relatively small distance apart transfer energy from the one coil to the other by means of an alternating magnetic field (**Figure 2**). The trick is to ensure good coupling between the coils, so that as much as possible of the magnetic flux generated by the transmitter coil is transferred to the receiver coil.

Various topologies are possible for the oscillator that drives the transmitter coil. Zero voltage switching (ZVS) oscillators are often used for this purpose. **Figure 3** shows a highly simplified schematic diagram of a ZVS oscillator with two MOSFETs. The combination of the transmitter coil and the capacitor forms a parallel resonant circuit, which directly determines the oscillator frequency. The circuit starts oscillating spontaneously after a DC supply voltage is connected, because one of the two MOSFETs will always start conducting earlier than the other one. Positive feedback from the drain of each MOSFET to the gate of the other MOSFET produces a phase shift of 180 degrees, so the MOSFETs alternately drive each other into conduction. As a result, each end of the LC circuit is alternately connected to ground.

This type of oscillator has several advantages compared to other versions:

- self-oscillating, with just one DC voltage source necessary;
- low power dissipation because the MOSFETs automatically switch at zero voltage crossover points;
- virtually sinusoidal current and voltage waveforms.

Another advantage is that the same circuit can be used on the receiver side, where it acts like a synchronous rectifier. Where the power source is connected on the transmitter side, the transferred power can be tapped off on the receiver side.

In practice this sort of circuit is suitable for power levels up to about 200 W, depending on the dimensioning of the components. It should be pointed out that unlike Qi chargers for mobile phones, this converter does not have any intelligence on board. However, its greater capacity makes the circuit ideal as a wireless power converter for more elaborate electronic circuitry or lighting.

Features

- Maximum power: 50 W
- Recommended continuous power: 0–40 W
- Supply voltage: 12.5–24 V_{DC} / 3 A min.
- Recommended supply voltage: 19.5 V (laptop adapter)
- Distance between coils: 1 cm maximum for optimal power transfer
- Efficiency: 88% at 40 W with 0.5 cm coil separation

Practical circuit

Let's start with the block diagram of the converter, shown in **Figure 4**. Along with the LC circuit, the MOSFETs and the filters, there are two driver circuits for the MOSFETs. Their job is to ensure that the MOSFETs switch quickly. The driver stages are powered from a separate regulated supply voltage to ensure proper operation, because if anything goes wrong in the drivers the MOSFETs will go straight to the happy hunting ground. There is also an electronic protection circuit that disables the drivers if the current in the LC circuit rises above a specific (adjustable) level. This protection circuit is only necessary on the transmitter side.

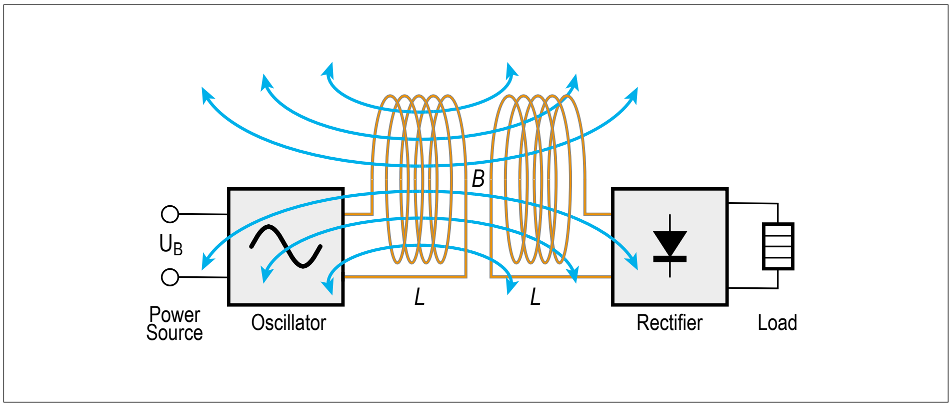


Figure 2. Operating principle of wireless power transfer.

Moving on to the full schematic in **Figure 5**, you can see that it contains quite a few components — much more than the simple design we initially described. The majority of the components are there for noise suppression and decoupling, which are necessary for proper operation of the circuit and keeping interference emissions at a reasonable level.

Let's do a walk-through of the circuit: At the top right you see the supply connector, which is connected to a regulated power supply. You can use a lab power supply for this, but a 19 V laptop AC adapter with a capacity of at least 70 W is also suitable. The PTC resettable fuse F1 is followed by the current sense resistor R18 for the current protection circuitry. Then the supply voltage passes through the large filter chokes L1 and L2 to the oscillator portion with the transmitter coil (L4). The LC resonant circuit consists of L4 and the series/parallel capacitors C8–C11. The resonant frequency is approximately 100 kHz. This frequency shifts slightly when the transmitter and receiver coils are close together.

The two MOSFETs T5 and T6 alternately connect the two sides of the LC circuit to ground. Each MOSFET has its own driver circuit to keep the switching edges steep and minimize power dissipation in the two MOSFETs. Each driver circuit consists of a push-pull stage (T3/T8 or T4/T9), a control section (T2/T7 or T1/T10), and a positive feedback network between the output and input (D3/R7 or D4/R8). The quality of the supply voltage for the driver circuits is crucial for proper driving of the MOSFETS, so they are powered from a separate regulated supply rail. The supply voltage is regulated by IC1, a switching step-down regulator from Würth's MagI³C line of power modules. The output voltage is set to 9.6 V by voltage divider R11/R14. The IC is surrounded by a number of buffer and decoupling capacitors. LED D7 lights up

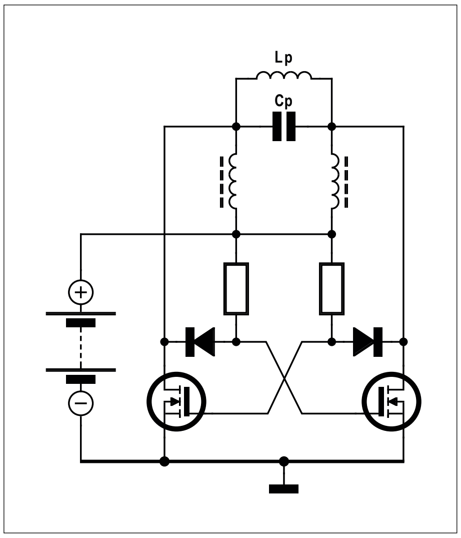


Figure 3. Basic circuit of a zero voltage switching oscillator.

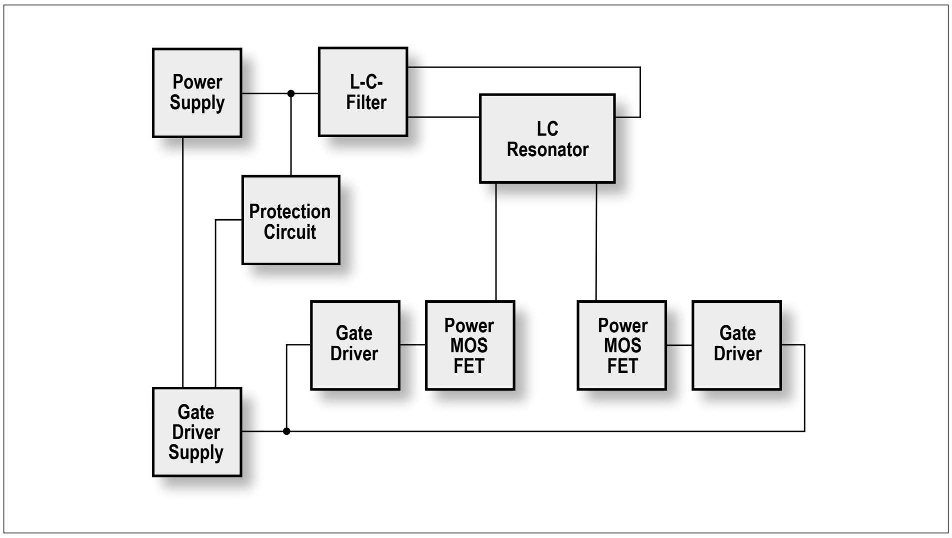


Figure 4. Block diagram of the wireless power converter.

Warning. **Never try to charge a Qi device (smartphone, etc.) with this circuit!**

when the voltage regulator is working. A diode (D5) and a TVSE overvoltage diode are include in the supply line to the driver circuits.
A jumper block (J1) is connected to the Enable input of the voltage regulator through voltage divider R9/R13. This allows the supply voltage, and with it the entire transmitter/receiver system, to be switched off manually. Normally the jumper is left in the Enable position and power is switched on and off at the connected power supply.

The current monitoring circuit consists of the components inside the grey box, which are only mounted on the transmitter PCB. Sense resistor R18 is present on both boards (for which reason there are two of them in the kit), but it could also be replaced by a wire jumper on the receiver board.

The voltage over the sense resistor R18 is measured by a high-side current monitor (IC2). Its output (pin 1) supplies a current that is linearly proportional to the voltage over R18 ($I_{out} = 0.004 \times V_{R18}$). This current generates a voltage over the voltage divider R20/R21/P1. A fraction of this voltage (adjustable with P1) goes to the input of a thyristor constructed from discrete transistors (T11/T12). When the input voltage at the junction of R23 and R24 exceeds a specific value (about 0.6 V), the thyristor goes into conduction and pulls the Enable pin of the step-down regulator to ground. This removes the supply voltage for the driver circuits, and the oscillator stops immediately.
The circuit remains in this state until the supply voltage is switched off or the reset button S1 is pushed. Resistor R27 limits the maximum current through D8 and T11, while diode D8 ensures that the undervoltage detection network around R9/R13 is not affected by R25/R26 when the thyristor is inactive.

The current limit level can be set between approximately 1.6 and 3 A, although the highest level is not recommended due to the risk of damage to the MOSFETs. During experiments at Elektor Labs, several MOSFETs bit the dust despite the various protection circuits, and they are a bit difficult to replace on the PCB. If you want to stay on the safe side, it’s a good idea to limit the transmitted power to about 40 W.

The protection circuitry will usually react immediately when the supply voltage is switched on. That can be seen from the blue LED D7, which lights up briefly and then goes dark. In that situation you have to press the Reset button to start up the circuit. The blue LED is continuously lit with the converter is working.

Construction

The double-sided PCB designed for the transmitter and receiver is shown in **Figure 6**. The board has a compact layout with components mounted on both sides. The large filter chokes and capacitors, as well as some other components, are located on the top side, with the MOSFETS and the rest of the components on the bottom side.

The components list is split into two parts: a general list which applies to both boards, and a separate list for the protection circuit components, which are only mounted on the transmitter board (inside the grey box on the schematic diagram). Experience with soldering SMDs is essential for assembling the PCBs. If you do not have suitable experience, you should not attempt this project.

If you have a reflow oven and a hot air soldering station, assembly is significantly easier. First apply solder paste to the top of the each PCB, inspect carefully, and then place the individual components. The kit contains a large number of resistors and capacitors with different values, so it is important to check the values carefully before fitting the components. That also applies to the various types of transistors. Also check the polarity of the electrolytic capacitors (you can use the photos of the assembled boards as a reference). Be careful with the separate components which are only mounted on the transmitter board. After all the components are fitted on the top side, you can put the board in the reflow oven.

The components on the top side can also be mounted using a hot air soldering station, but that can be difficult with the two large filter chokes (L1 and L2) because they have a lot of mass and therefore absorb a lot of heat. You can also try to do this with a normal soldering iron – the coil terminals are accessible from the side through openings in the ferrite cores. Voltage regulator IC1 has a heat sink pad on the bottom which is also the power ground. It is essential to solder this pad to the PCB. If you use solder paste and a hot air soldering station, the IC will drop slightly when the paste turns liquid, and then you know that the heat sink pad is properly soldered. With a normal soldering iron it is virtually impossible to see whether or not this pad is firmly soldered, so getting the desired result is a bit of a gamble.

Next you have to do the bottom side of each PCB. Since you can't bake a double-sided SMD PCB in a reflow oven without special measures, the bottom side of each board has to be assembled manually using a hot air soldering station or

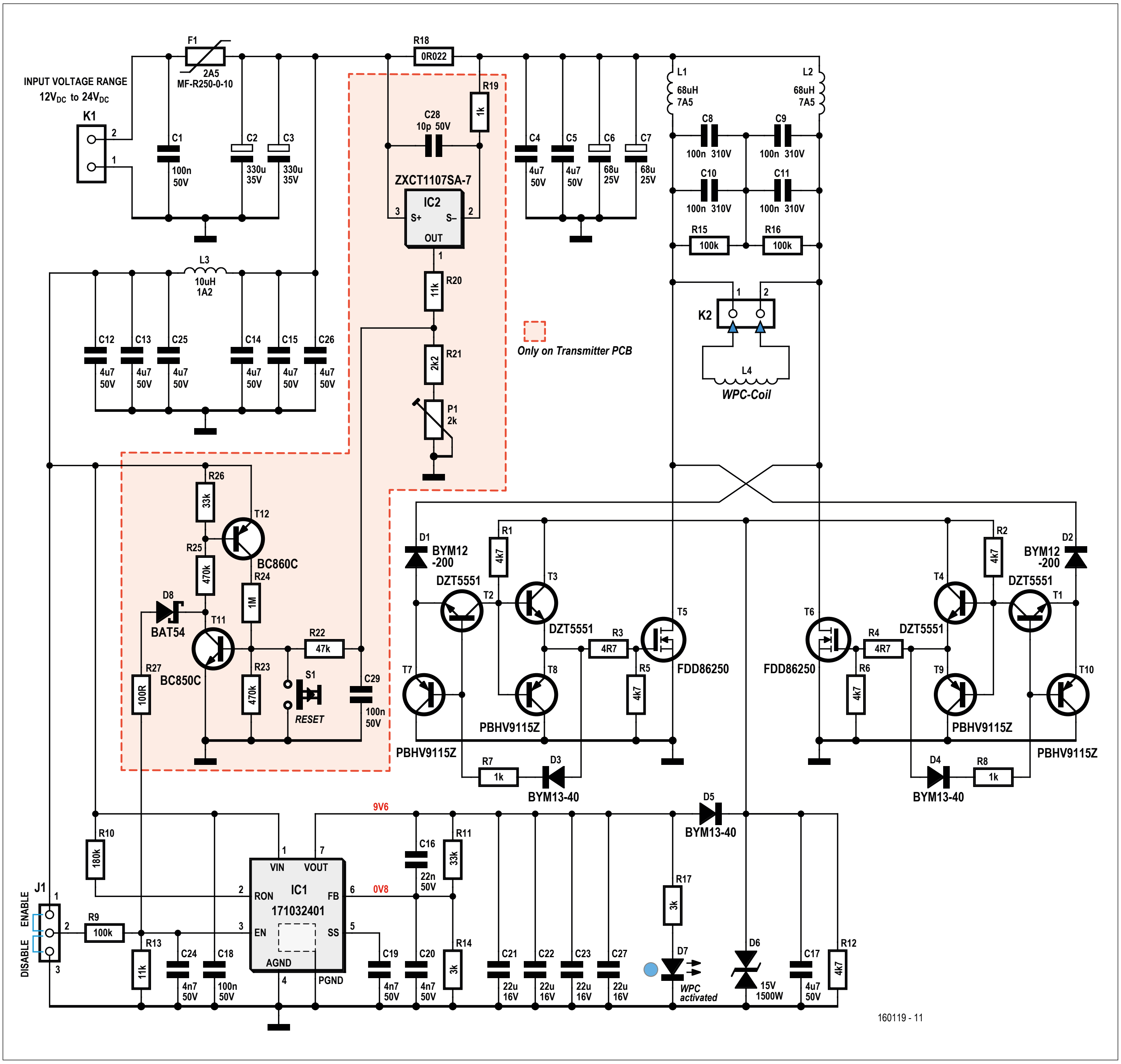


Figure 5. Complete schematic diagram. The receiver is the same as the transmitter with the exception of the components inside the grey box.

a normal soldering iron, depending on your personal SMD soldering experience. Here again there are some components that are only mounted on the transmitter board. They are simply omitted on the receiver board.

Once you have all the SMDs mounted, it's time to solder a few leaded parts on the PCBs: fuse F1, electrolytic capacitors C2 and C3, capacitors C8–C11, pushbutton S1, the two terminal strips K1 and K2, and jumper block J3.

Use the photos in **Figures 7 and 8** to check that everything has been correctly mounted. Now you're ready for the first test.

Hands-on

Connect the special coils with ferrite substrates to the K2 connectors of the two boards. Connect K1 on the transmitter board to a power source (for example, a laptop adapter), and connect K1 on the receiver board to a load. Bear in mind that the output voltage is nearly the same as the supply voltage for the transmitter. Start cautiously with a relatively small load, such as a 47 Ω /10 W power resistor. Set jumper J3 of each board to the Enable position. Then stack the coils on top of each other with the ferrite plates on the outside, as illustrated in **Figure 9**. You can use a small piece of wood or plastic about 5 mm thick to separate the two coils. The transfer efficiency remains fairly high up to about 1 cm separation; after that the efficiency and the output voltage drop off. Now you can switch on the power supply, and LED D7 should light up. If it does not, press button S1. If the LED goes dark again after you press S1, the current limit level is probably set

Component List – Covers 1 transmitter / receiver board

Resistors
R1,R2,R5,R6,R12 = 4.7kΩ 1%, 0.25W, SMD 1206
R3,R4 = 4.7Ω 1%, 0.25W, SMD 1206
R7,R8 = 1kΩ, 1 %, 0.25W, SMD 1206
R9 = 100kΩ, 1 %, 0.125W, SMD 0805
R10 = 180kΩ, 1 %, 0.125W, SMD 0805
R11 = 33kΩ, 1 %, 0.125W, SMD 0805
R13 = 11kΩ, 1 %, 0.125W, SMD 0805
R14,R17 = 3kΩ, 1 %, 0.125W, SMD 0805
R15,R16 = 100kΩ, 1 %, 0.25 W, SMD 1206
R18 = 0.022Ω 5%, 0.5 W, SMD 1206

Capacitors
C1,C18 = 100nF 50V, 10%, X7R, SMD 0805
C2,C3 = 330µF 35V, 20%, 0Ω15, 10x16 mm, radiaal
C4,C5,C12-C15,C17,C25,C26 = 4.7µF 50V, 10%, X7R, SMD 1210
C6,C7 = 68µF 25V, 0.015Ω, 8x12 mm, radial, Can - SMD
C8,C9,C10,C11 = 100nF 310 V_{AC}, 10%, polypropylene, 10mm pitch
C16 = 22nF 50V, 10%, X7R, SMD 0805
C19,C20,C24 = 4.7nF 50V, 5%, NP0, SMD 0805
C21,C22,C23,C27 = 22µF 16V, 20%, X5R, SMD 1210

Inductors
L1,L2 = 68µH 20%, 7.5A, 0.0273Ω, SMD (Würth 74435586800)
L3 = 10µH, 20%, 1.2A, 0.322Ω, SMD (Würth 74438336100)
L4 (not on PCB) = 24µH 10%, 6A, 0.1Ω (Würth Wireless Power Charging Coil)

Semiconductors
D1,D2 = BYM12-200, DO-213AB (MELF)
D3,D4,D5 = BYM13-40, DO-213AB (MELF)
D6 = WE-TVSP (15V 1500W), DO-214AB
D7 = LED, blue, SMD 0805
T1,T2,T3,T4 = DZT5551, SMD SOT-223
T5,T6 = FDD86250, SMD TO252
T7,T8,T9,T10 = PBHV9115Z, SMD SOT-223
IC1 = WE_VDRM_Power-Module SMD TO263-7 (Würth type 171032401)

Miscellaneous
K1,K2 = 2-way PCB screw terminal block, 5mm pitch
J1 = 3-pin pinheader, 0.1'' pitch
F1 = PTC resettable fuse, 0.048Ω, I_{hold} 2.5A/30V (MF-R250-0-10)
J1 = jumper, 0.1'' pitch

PCB # 160119-1

Additional parts on Transmitter Board
R19 = 1kΩ 1%, 0.125W, SMD 0805
R20 = 11kΩ 1%, 0.125W, SMD 0805
R21 = 2.2kΩ, 1%, 0.125W, SMD 0805
R22 = 47kΩ, 1 %, 0.125W, SMD 0805
R23,R25 = 470kΩ, 1 %, 125 mW, SMD 0805
R24 = 1MΩ 1%, 0.125W, SMD 0805
R26 = 33kΩ, 1%, 0.125W, SMD 0805
R27 = 100Ω 1%, 0.125W, SMD 0805
P1 = 2kΩ 25%, 0.2W, SMD trim pot (e.g. Bourns 3364W-1-202E)

C28 = 10pF 50V, 5%, C0G/NP0, SMD 0805
C29 = 100nF 50V, 10 %, X7R, SMD 0805
D8 = BAT54, SMD SOT-23
T11 = BC850C, SMD SOT-23
T12 = BC860C, SMD SOT-23
IC2 = ZXCT1107SA-7, SMD SOT-23
S1 = pushbutton with make contact, SPST-NO (FSM4JRT)

Kit of parts with all components and boards for 1 transmitter and 1 receiver: Elektor Store # 160119-71

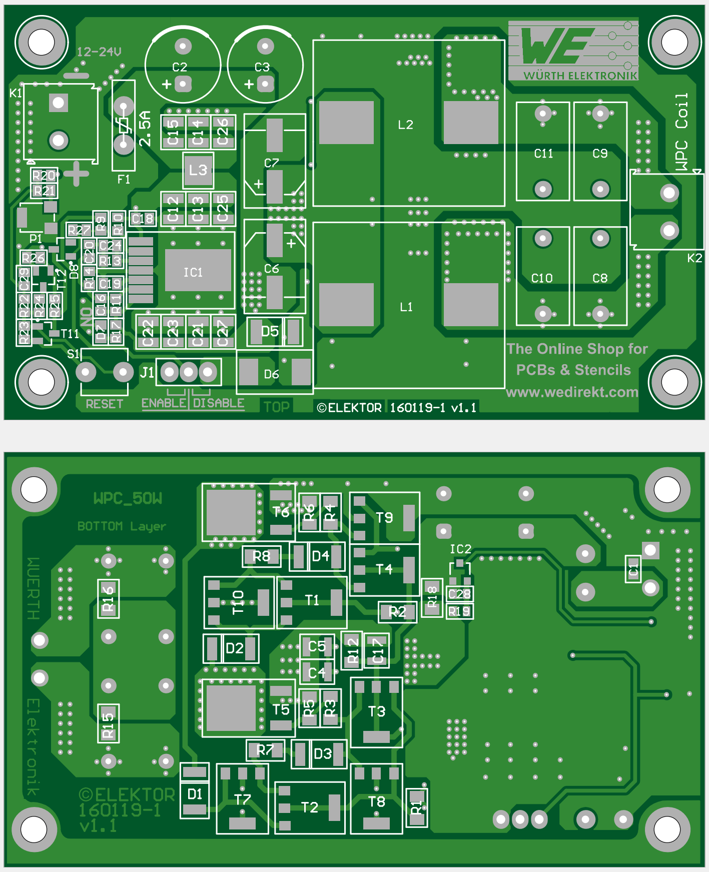


Figure 6. The double-sided PCBs for the transmitter and receiver have components (SMD and leaded) on both sides.

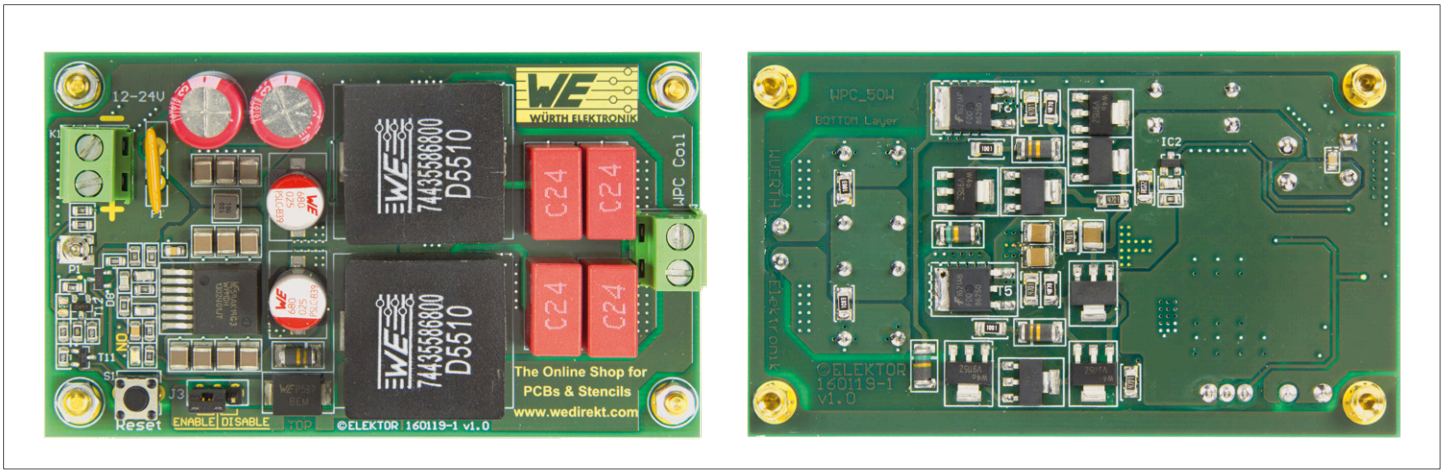


Figure 7. Top and bottom sides of the assembled transmitter board.

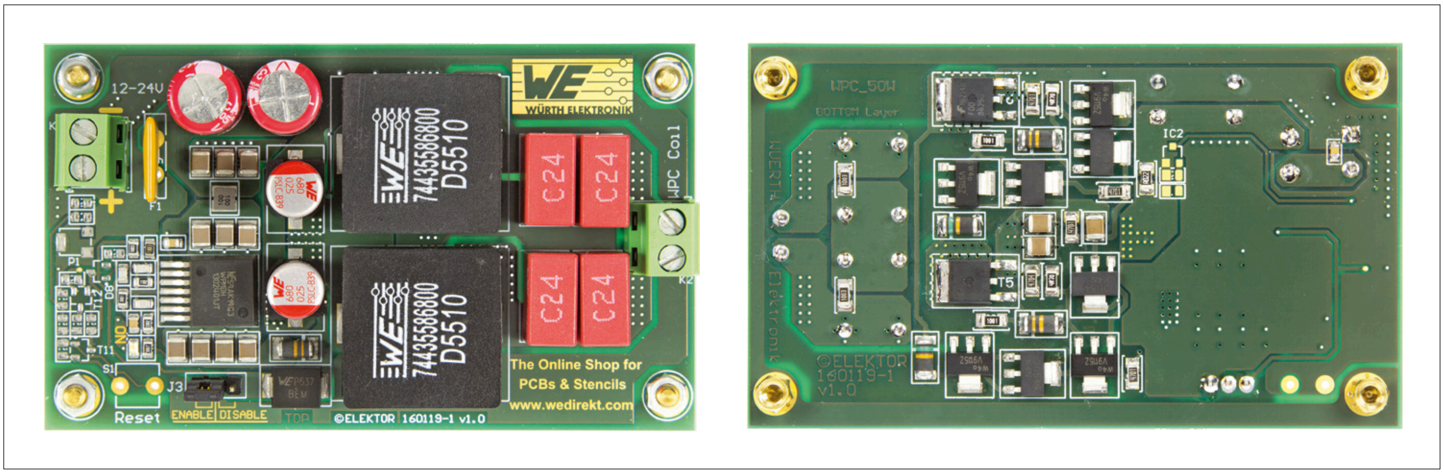


Figure 8. Top and bottom sides of the assembled receiver board.

too low. You can correct that by adjusting P1. If LED D7 lights up and everything appears to be okay, check the voltage over the load resistor. Once the converter is working properly, you can increase the load. To avoid the risk of having to replace the MOSFETs, we recommend that you keep the load under 40 W or so. Although the circuit is theoretically good for 50 W, with that sort of power level it's advisable be cautious, even though the circuit has built-in protection.

The position of potentiometer P1 for current limiting depends on the amount of power you want to transfer. The safest approach is to set it so the protection trips just above the desired power level.

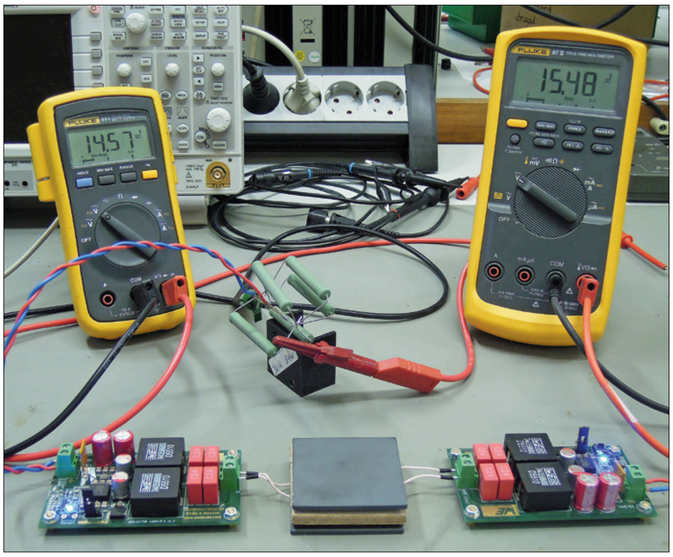


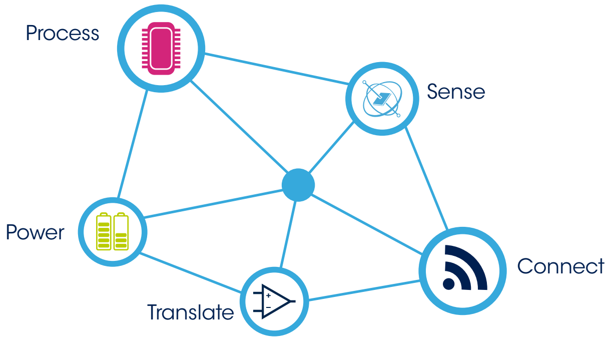
Figure 9. Test setup in the Elektor Labs. The electric load consists of five power resistors wired in parallel.

This set is intended to be used for experimenting with wireless energy transfer at relatively high power levels. For that reason we do not describe any practical applications here – we leave that up to you. We hope you have a lot of fun with the set!

(160119-I)

Web Link: [1] http://www.we-online.com/web/en/electronic_components/produkte_pb/application_notes/anp032.php

life.augmented


Process
Sense
Power
Connect
Translate

Return of the Wi-Fi Controller Board

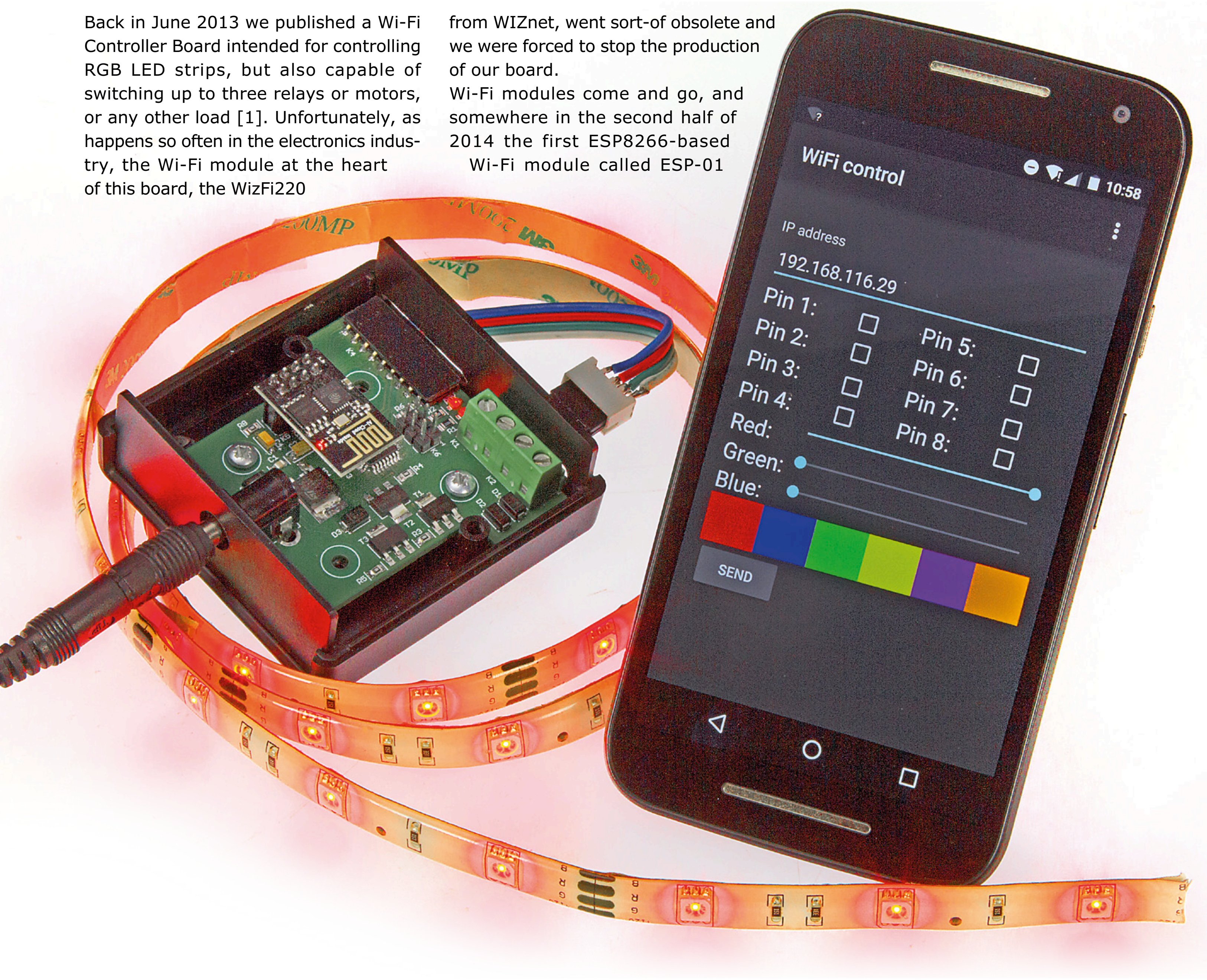
www-connect objects to your smartphone

By **Roy Aarts** (intern, Elektor Labs) and **Clemens Valens** (Technical Manager, Elektor Labs)

Connected Objects is what people covet today. Anything except the dog's temper has to be controllable from a smartphone or tablet and even more 'stuff' must report back via the internet. If that's what you really want, who are we to refute that desire? With our revamped Wi-Fi Controller Board you can control electric loads over the Internet and receive status information in return.

Back in June 2013 we published a Wi-Fi Controller Board intended for controlling RGB LED strips, but also capable of switching up to three relays or motors, or any other load [1]. Unfortunately, as happens so often in the electronics industry, the Wi-Fi module at the heart of this board, the WizFi220 from WIZnet, went sort-of obsolete and we were forced to stop the production of our board.

Wi-Fi modules come and go, and somewhere in the second half of 2014 the first ESP8266-based Wi-Fi module called ESP-01



(**Figure 1**) made its appearance. The ESP8266 is a low-cost chip manufactured by Chinese company Espressif and integrates a full TCP/IP stack together with a microcontroller. The modules that have quickly become very popular are made by AI-Thinker, while no-name clones are available too. Because these modules are dirt cheap we figured it was a good idea to redo our Wi-Fi Controller Board but based on an ESP-01 module this time.

The circuit

The original Wi-Fi Controller Board started out as an Arduino project with software developed for the ATmega328, and was then ported to a PIC18F14K50 microcontroller to add USB capabilities and to use a microcontroller with fewer pins. The design was fully through-hole. For the new design we went back to the ATmega328, but now in a surface-mount package, and the rest of the circuit was done with SMT parts too. The USB port was dropped (**Figure 2**).

The power supply was simplified also — the 24 VDC input capability was lowered to 12 VDC and the switching voltage regulator in 2016 is a linear device (IC1). This reduces EMC emissions somewhat and allowed for a more compact design.

Because the microcontroller (IC2) has more pins than the MCU of the previous design, the expansion port K4 was extended to eight bits.

The three power outputs consist of three MOSFET transistors (T1, T2, T3), each capable of passing up to 4 A ($R_{DS(ON)}$ = 100 mΩ), i.e. more than enough as we recommend keeping the load to a maximum of 1 A per output. Each output has a so-called freewheeling diode allowing inductive loads like relays to be switched too.

The ESP-01 module is plugged onto connector K5. Care must be taken to orient the module properly because the symmetrically placed pins on the module are not labelled. That's why we put the module on a board side so that there is only one way to insert it while still being able to close the enclosure.

To communicate with the Wi-Fi module we use the serial port. A separate signal is used to reset the module if needed. The remaining pins are not used. They must be wired like we did to avoid the ESP-01 entering special firmware-reprogramming modes where you do not want to go.

The software

To make the system work software is needed. In our case it consists of an Arduino sketch running on the MCU and an Android app to provide the user interface.

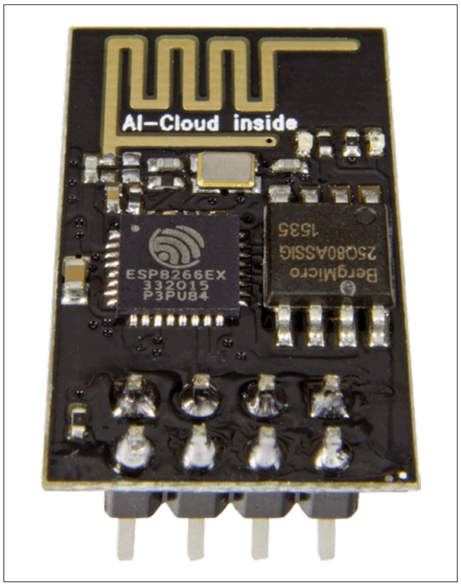


Figure 1. The ESP-01 module is a cheap, serial-to-Wi-Fi module with only eight pins, two of which are at loggerheads over being pin 1 (square pad).

Basically the sketch functions as a simple web server that waits for commands to come in. As soon as it receives something it can understand, e.g. a new value

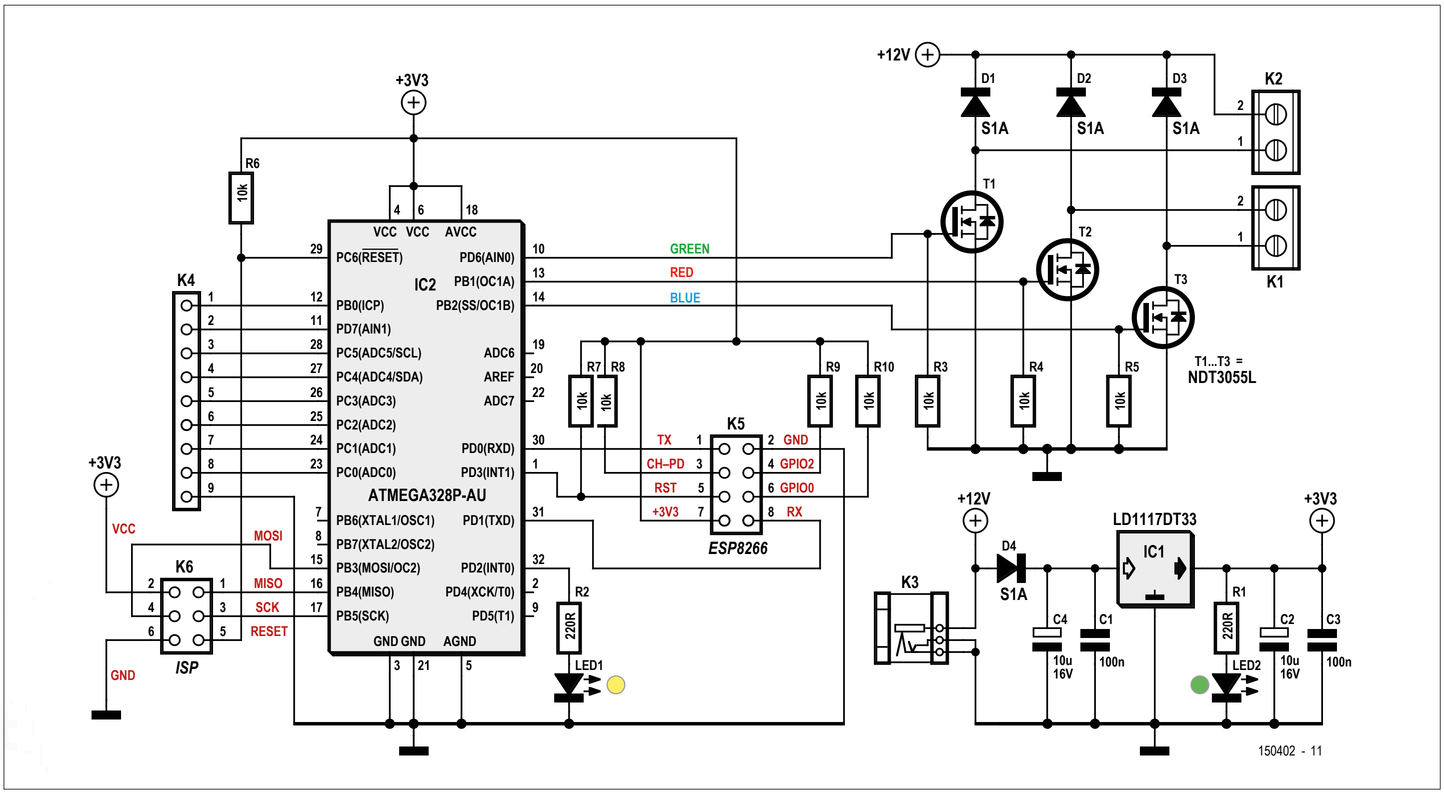
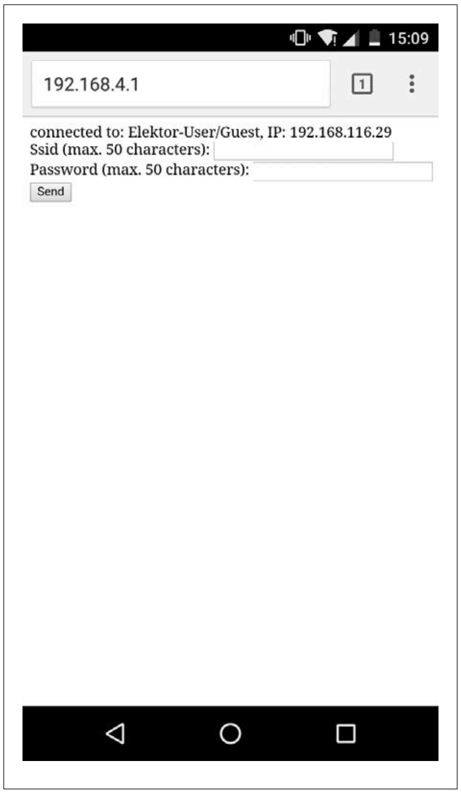


Figure 2. An ATmega328P microcontroller, three power MOSFETs and some more parts make up the circuit of the Wi-Fi Controller Board. It is the invisible module that plugs onto K5 that does all the magic.

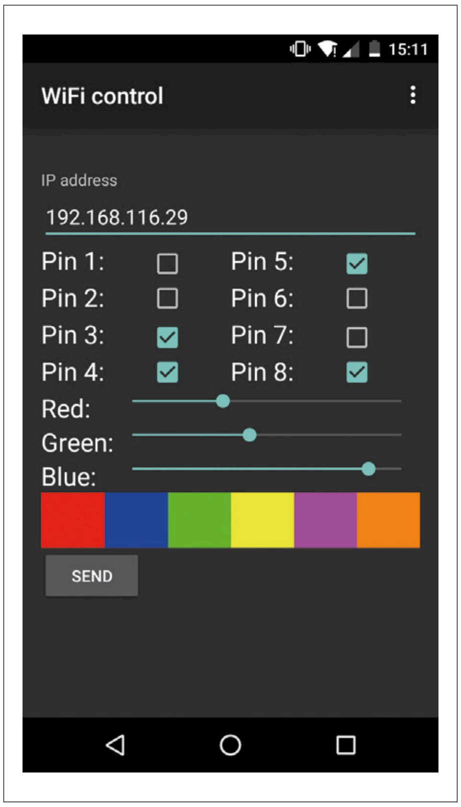


15:09

192.168.4.1

connected to: Elektor-User/Guest, IP: 192.168.116.29
Ssid (max. 50 characters):
Password (max. 50 characters):
Send

Figure 3. The configuration form served by the ESP-01's access point lets you set up the controller for your network.



15:11

WiFi control

IP address
192.168.116.29

Pin 1:	□	Pin 5:	☑
Pin 2:	□	Pin 6:	□
Pin 3:	☑	Pin 7:	□
Pin 4:	☑	Pin 8:	☑

Red:
Green:
Blue:

SEND

Figure 4. The main page of the Android app proudly showing its controls. Do not forget to tap the Send button to effectively transmit your commands to the www and from there to Wi-Fi Controller Board.

for the R, G & B outputs or for one of the extension port pins, the sketch will extract the parameter and act accordingly. It will then return the new output value. The R, G & B outputs will produce PWM signals with a 0 to 100% duty-cycle range, while the digital outputs will simply be High or Low.

When the board is powered on or reset, the sketch will execute its setup function. This function must first deal with (ignore) any serial data coming from the ESP-01 module after reset that's at a strange baud rate (76,800 baud) before the ESP module switches to 115,200 baud. From that point on the Wi-Fi module is ready for normal operation.

This high-ish baud rate is a bit inconvenient, because the error margin is very small. We found that every ESP module uses its own interpretation of 115,200 baud, obliging us to devise an algorithm to automatically adapt to it. This works by starting below 115,200 baud, send an AT command to the module and then trying to understand its answer. If it is not clear, the MCU increases the baud rate slightly. This is repeated until proper communication is established.

The sketch also listens for connections coming in at the special IP address 192.168.4.1. This is the module's Access Point (AP) address needed to configure the module for the network in which it is supposed to be integrated. If a connection request arrives at this address, from a smartphone for instance, the sketch will serve up a configuration form where the user can enter the SSID and password for the network in which the Wi-Fi controller board will be active. After saving the information, the ESP module will automatically try to connect to the configured network. To remove the controller from the network, simply reconnect your phone or computer to the special AP address and change the parameters.

The source code for the sketch and the app is available in archive file # 150402-11.zip you can download free of charge from [2] and sure, you are equally free to adapt it to your needs.

Using it

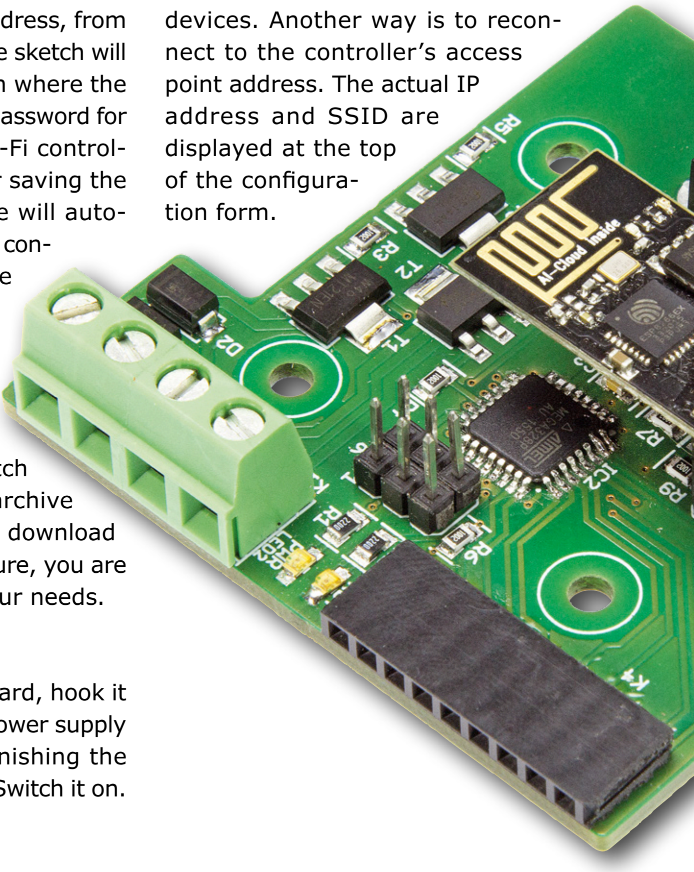
Install the Wi-Fi controller board, hook it up to its load and connect a power supply (5 - 12 VDC) capable of furnishing the current required by the load. Switch it on. If no smoke is apparent you are heading in the right direction. Continue.

On an Android (2.3 and up) device with Internet connection and Wi-Fi capabilities, go to the Google Play Store and install the app "WiFi control" by Elektor Team. Connect the Android device to the access point (AP) with a name like "ESP8266 something". If you are not sure, it's the Wi-Fi network with enormous field-strength when you are close to the module and that disappears when you switch the module off. The connection does not require a passphrase.

On the Android device, open a browser and point it to 192.168.4.1. If all is well you should see a form appear after a few seconds showing typical Wi-Fi network parameters like SSID and passphrase (**Figure 3**). Enter the parameters of the network that you want to use and save it. The page will then display the IP address attributed to the ESP module by the network. Make a note of it somewhere.

Now connect the Android device to the network for which you just configured the ESP module. Launch the app and enter the IP address you wrote down. You should now be able to control your loads. Remember to click the Send button after every parameter change (**Figure 4**).

If you have forgotten the IP address of your Wi-Fi Controller Board you can often discover it by connecting to your Wi-Fi router and consult its list of connected devices. Another way is to reconnect to the controller's access point address. The actual IP address and SSID are displayed at the top of the configuration form.



Component List

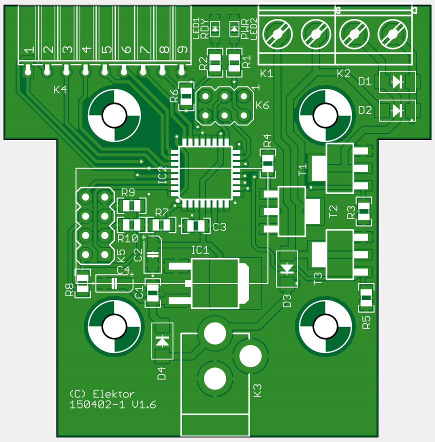
Resistors (0805)
R3,R4,R5,R6,R7,R8,R9,R10 = 10kΩ
R1,R2 = 220Ω

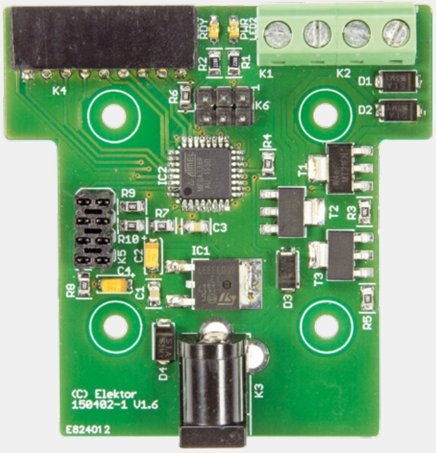
Capacitors (0805)
C1,C3 = 100nF
C2,C4 = 10µF 16V, case A

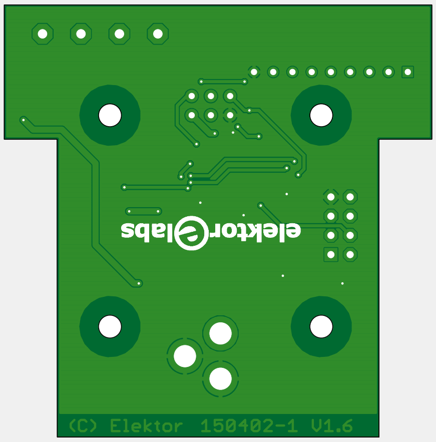
Semiconductors
IC1 = LD1117DT33
IC2 = ATmega328P-AU, programmed, Elektor Store # 150402-41
T1,T2,T3 = NDT3055L
D1,D2,D3,D4 = S1A

Miscellaneous
LED1 = LED, yellow
LED2 = LED, green
K1,K2 = 2-way PCB screw terminal block, 0.2'' pitch
K3 = barrel jack, 2.5mm
K4 = 9-way pinheader socket, 0.1'' pitch
K5 = 8-way (2x4) pinheader socket, 0.1'' pitch
K6 = 6-way (2x3) pinheader, 0.1'' pitch

Miscellaneous
ESP8266 Wi-Fi module, Elektor Store # 150445-91 (SKU-17326)
PCB v1.6, Elektor Store # 150402-1
ATmega firmware & fuse settings from [2]
App from Playstore: 'WiFi control' by Elektor Team [3]



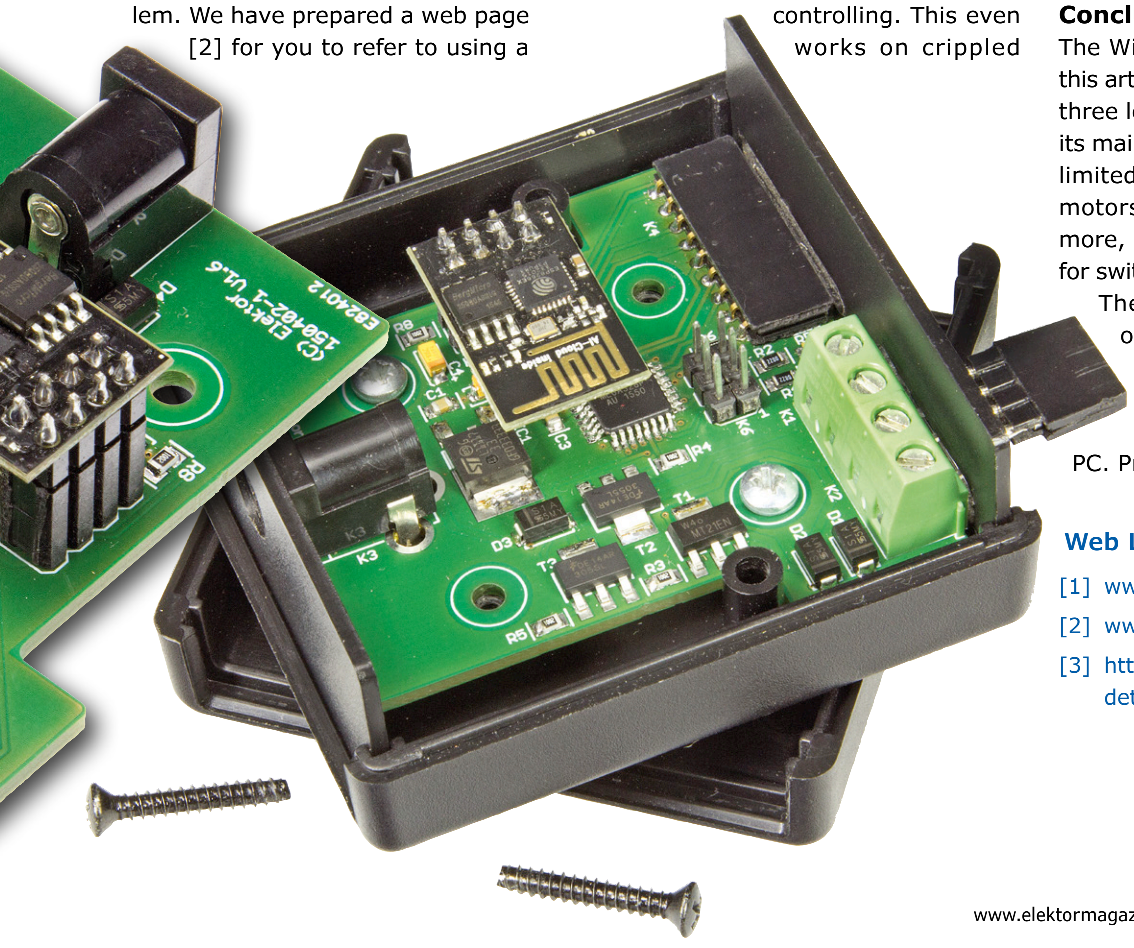




No Android?

What if you don't have an Android device? Or the app doesn't work on your phone or tablet for some obscure reason? No problem. We have prepared a web page [2] for you to refer to using a web browser capable of handling HTML 4.01 or higher.

Open the page, enter the module's IP address (not its AP IP address), and start controlling. This even works on crippled devices like iPhones and iPads (if you are capable of copying our HTML files onto your device).



Conclusion

The Wi-Fi Controller Board presented in this article allows the user to control up to three loads with a PWM signal. Although its main target is RGB LED strips, it is not limited to this type of load. Relays and motors can be controlled too. Furthermore, eight digital outputs are available for switching tiny loads like optocouplers.

The device is controlled wirelessly over Wi-Fi from either an Android device running the special app or from a web browser on any other type of device like an iPhone or a PC. Press to control — have fun!

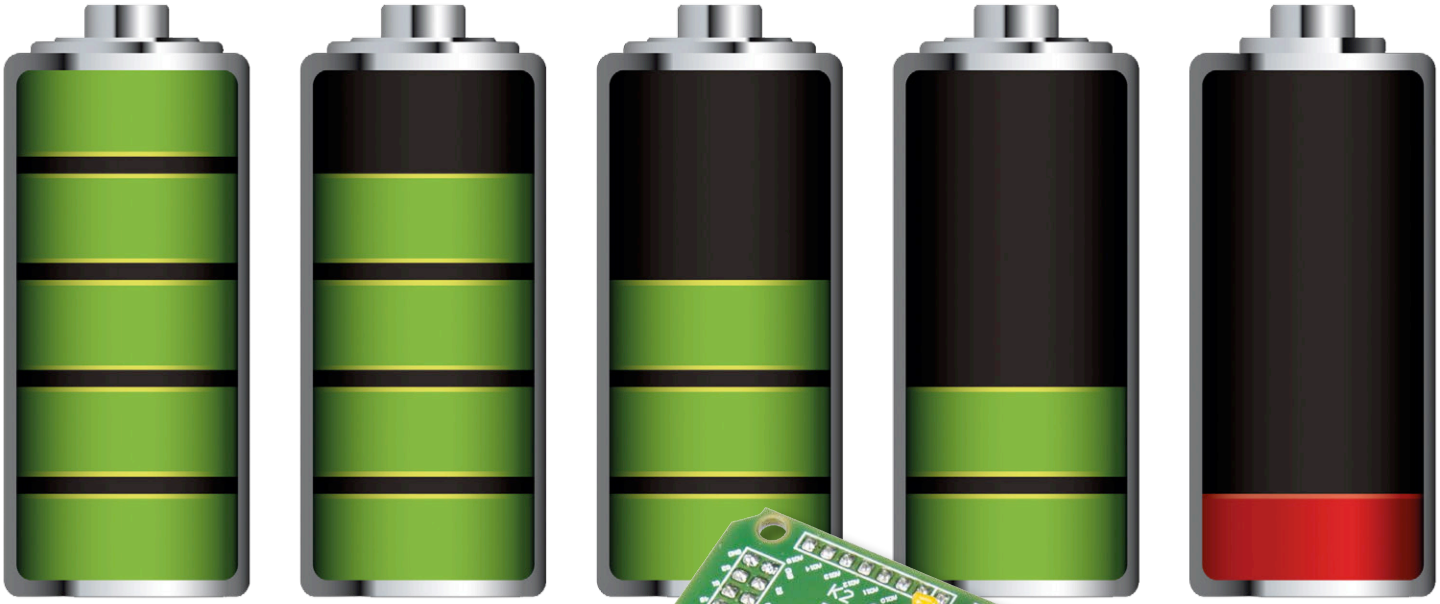
(150402)

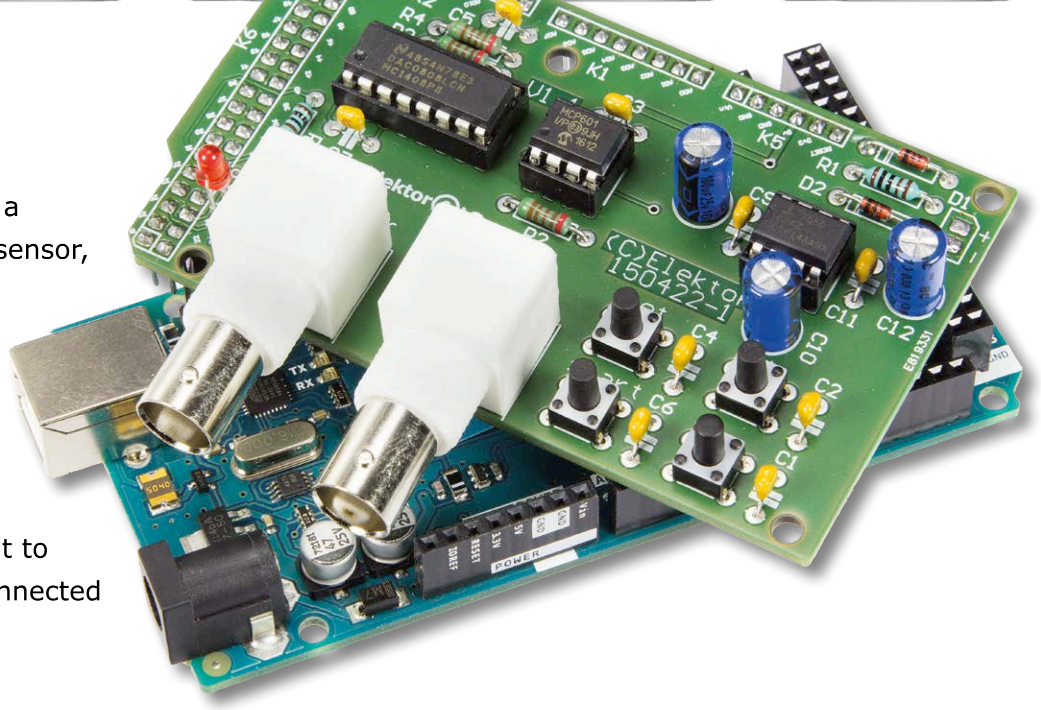
Web Links

[1] www.elektormagazine.com/120718
[2] www.elektormagazine.com/150402
[3] https://play.google.com/store/apps/details?id=com.elektor.wificontrol

Voltage Tracker for Oscilloscope

Slow-timebase measurements on your oscilloscope using an Arduino shield





By **Kurt Schuster** (Germany) and
Luc Lemmens (Elektor Labs)

Suppose you want to measure a slowly-varying voltage from a sensor, or perhaps plot the discharge curve of a battery. Is it really necessary to sit there and note down a reading from the multimeter every so many seconds or minutes? It is less tedious and much more elegant to use this Arduino-based unit connected to your oscilloscope.

Two questions motivated Kurt Schuster at the beginning of this project: 'is my battery any good?' and 'is this charger charging my battery correctly?'.

A graphical display of the battery voltage during charge and discharge would be very helpful in answering these questions. However, most modern oscilloscopes, despite their large storage capacity, are not up to the task: the slowest horizontal timebase is typically between 1 and 50 seconds per division, and so a screen 12 divisions wide can only show 12 seconds to 10 minutes of results at a time. Often that is not enough to monitor slow trends, which can occur over periods measured in hours. And a long record length is not much help: who wants to scroll through dozens of screens of data? The most convenient display would be to fit the entire trace on one screen.

The problem can of course be solved using a data logger or a multimeter with a logging facility. Some such meters have a built-in buffer memory, and some allow connection to a PC run-

ning suitable software. However, most meters do not offer these functions and those that do often aren't cheap.
And so it was decided to considerably extend the timebase range of an existing oscilloscope using low-cost microcontroller technology, and so the voltage tracker front-end circuit came into being. Before we get started, however, let us remark that despite the 'digital' user interface using pushbuttons, and the display of settings on the oscilloscope screen, our voltage tracker works perfectly well with old-school analog oscilloscopes as well as with their more modern digital cousins.

Preliminaries

Two questions now arise: 'how should the voltage tracker operate?' and 'what components do we need to build it?'.
The first question is easy to answer: an analog-to-digital converter will periodically sample the input voltage, and a microcontroller will store the results. Later, the microcontroller will cyclically output the stored results as a signal to the oscilloscope using a digital-to-analog converter, much faster than real-time. Now, given that we have a microcontroller available, it is an easy matter to add a user interface with pushbuttons and a graphical display of settings on the screen by outputting specially-constructed signals to the oscilloscope.

This leads us to the answer to the second question. The Arduino is a low-cost platform and has an established development environment in the form of the free Arduino IDE: why reinvent the wheel? If we use an Arduino board as a ready-made basis for the design, then the add-on hardware can be built as a shield that can be plugged in to it, and this will keep the complexity of the circuit board within reasonable limits.
Most microcontrollers have enough RAM for our application, as we only need to store as many data points as will fit in the horizontal resolution of the oscilloscope: any more would be an unnecessary luxury. Kurt Schuster used an Arduino Mega 2560 with a comfortable 8 Kbyte of RAM, of which only 1.8 Kbyte is used for storing readings and for the other functions of the program. But almost any Arduino board will do, even an Arduino Uno, as long as it can run at 16 MHz to provide the necessary processing power and meet timing requirements.
If you wish to build the project on a shoestring, it is possible to dispense with the Arduino board and just use the bare microcontroller. Every device in the ATmega microcontroller series offers at least one analog input, behind which sits a reasonably fast ADC. However, although we have a ready-made analog input, the analog output will have to be implemented via a digital output, for example using PWM (pulse-width modulation). Better, and more accurate, is to use a 'proper' dedicated external DAC chip, which converts a supplied byte directly to an analog voltage. An operational amplifier can be connected to its output to act as a buffer, or low-impedance output stage. With a symmetrical power supply a clean output signal referenced to ground can be obtained.

Components and circuit

We could simply choose an all-singing all-dancing hypermodern DAC, plonk it on a board with some buttons and the shield would be done. Unfortunately, however, such chips are not good value for money and moreover, invariably come in SMD packages. We have plenty of room on the shield, and so we decided to use through-hole components only in the interests of simplifying construction and making the project open to beginners in the art of soldering. Luckily, despite its advancing years, the type DAC0808 is still readily available: it is reliable, economical, and it even looks like a proper IC. It will not blow off the board in a light breeze or if you breathe while soldering it. The rest of the circuit is equally easy to assemble, especially as a ready-made printed circuit board (**Figure 1**, and see parts list) is available from Elektor. Then it is just a matter of loading

Component List

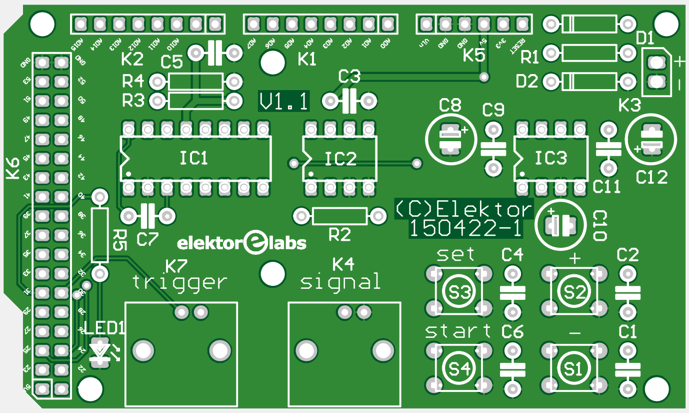


Figure 1. The printed circuit board, designed at Elektor Labs. The board is available from the Elektor Store.

Resistors
R1,R5 = 1kΩ 5%, 0.25W
R2,R3,R4 = 5.11kΩ 1%, 0.25W, metal film

Capacitors
C1–C7,C9,C11 = 100nF 50V, X7R, 0.2'' pitch
C8,C10,C12 = 100µF 25V, radial, 0.1'' pitch

Semiconductors
D1,D2 = 1N4148
LED1 = red, 3mm
IC1 = DAC0808LCN, DIP16
IC2 = MCP601-I/P, DIP8
IC3 = ICL7660CPAZ, DIP8

Miscellaneous
K1,K2 = 8-pin pinheader
K3 = 2-pin pinheader
K7,K8 = BNC socket, PCB mount, right angled
K5 = 6-pin pinheader
K6 = 36-pin (2x18) pinheader
S1–S4 = pushbutton, 6x6 mm, PCB mount
IC sockets for IC1,IC2,IC3
Two coax cables with BNC plugs
PCB # 150422-1 from Elektor Store
Arduino Mega, # 140566-93 from Elektor Store

Caution: all pinheaders to be mounted at PCB underside

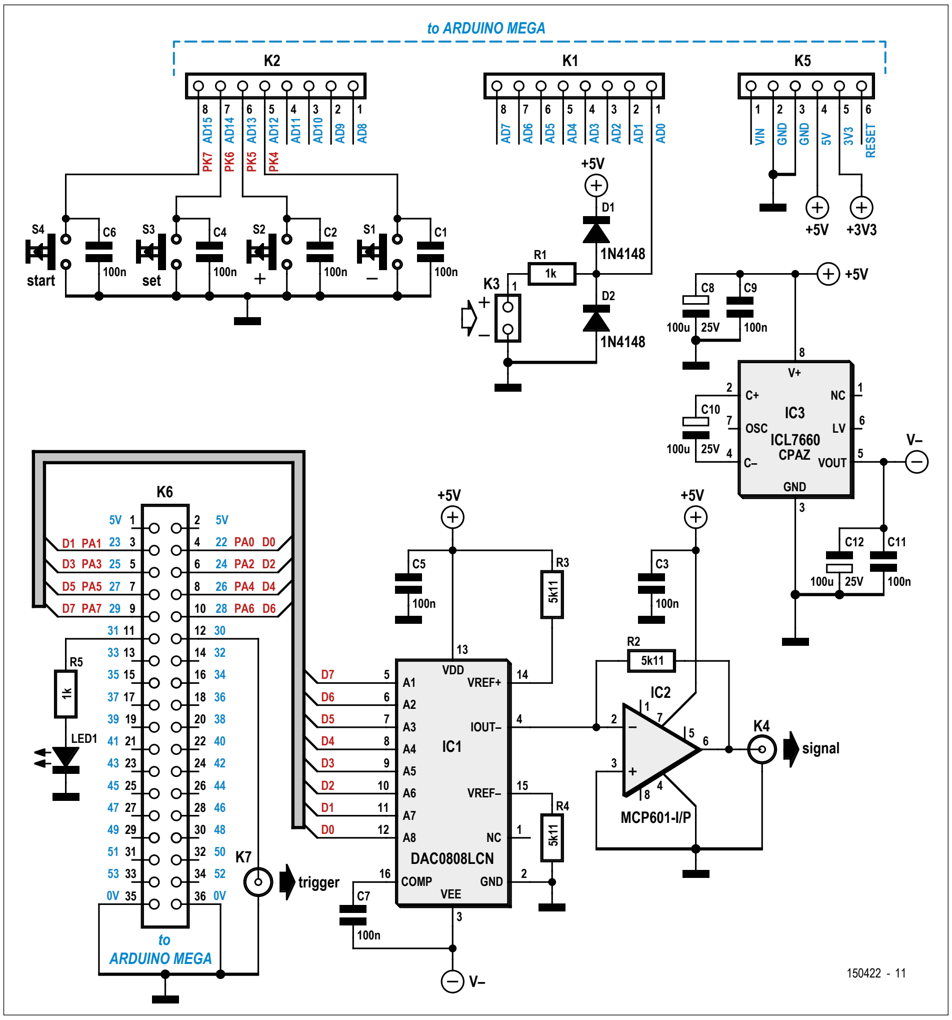


Figure 2. The circuit of the Arduino shield is very straightforward.

the firmware onto the Arduino board. The Arduino 'sketch' (program) file and the printed circuit board layouts are available for free download from the Elektor web page accompanying this article [1].

The circuit diagram in **Figure 2** shows how the DAC (IC1) is driven from port A of the ATmega, via K6. IC2 is the output buffer: we have chosen a rail-to-rail output opamp from Microchip. The electronics of the Arduino board is conventionally powered from +5 V, and this is available to the shield as well. We also require a negative rail for the DAC and opamp (but only at low current) and this auxiliary supply is generated by voltage converter IC3. As is so often the case with Arduino shields, the circuit is very straightforward. The only other components are four buttons to operate the device and an LED. The voltage to be measured comes in on K3. The analog input of the microcontroller on the Arduino board is not protected against damage from negative or excessive positive voltages, and so we have added protection resistor R1 and two diodes, which will limit the voltage appearing on the microcontroller's pin to a safe range.

Since we are using the built-in ADC of the microcontroller, the available voltage range is fixed at 0 V to +5 V. This is a convenient range for working with Li-ion rechargeable cells, which, depending on the exact type, operate at voltages of up to 4.6 V. If you wish to measure voltages on packs of cells or on a 12 V car battery, a simple voltage divider can be added. Resistor values in the region of 100 kΩ are suitable, as the analog input has a very high impedance of around 100 MΩ.

As mentioned above, the circuit is powered directly from the Arduino board. In turn, this will receive power either at 5 V over USB (for example, from a

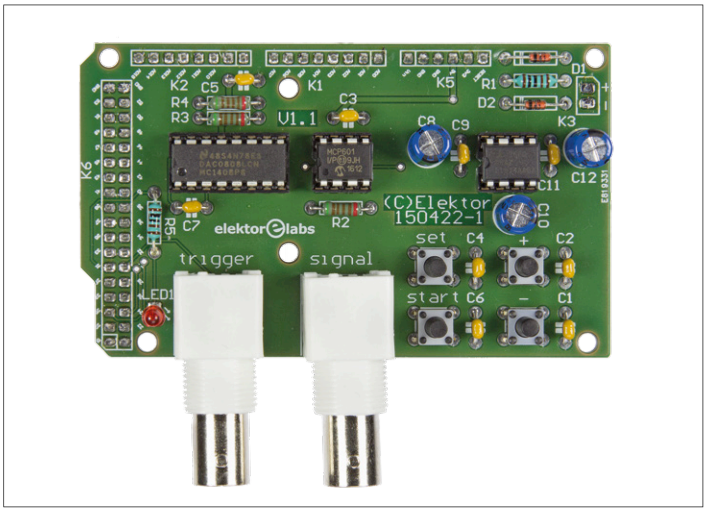


Figure 3. The fully-assembled board, ready to be plugged in to an Arduino.

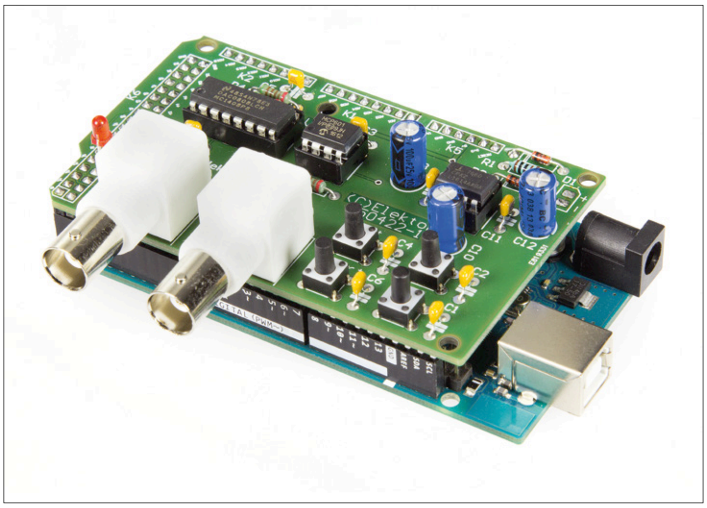


Figure 4. The sandwich of shield plus Arduino make the voltage tracker a compact unit.

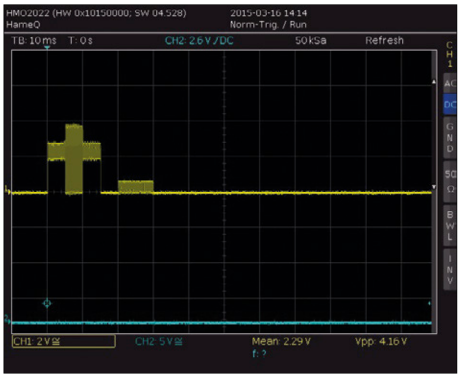


Figure 5. Display when setting the **timebase** parameter. The oscilloscope shows a ‘t’ and a staircase with one step, which corresponds to a timebase of one minute per division.

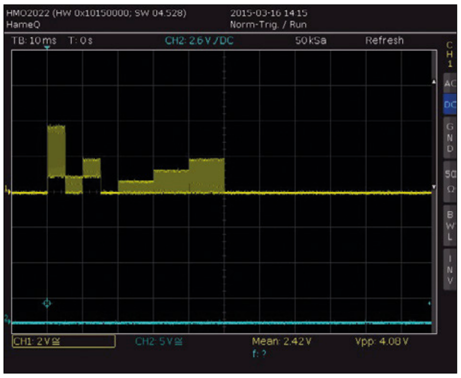


Figure 6. Setting the **level**. The display shows an ‘l’ and a three-step staircase, corresponding to an offset of 3 V.

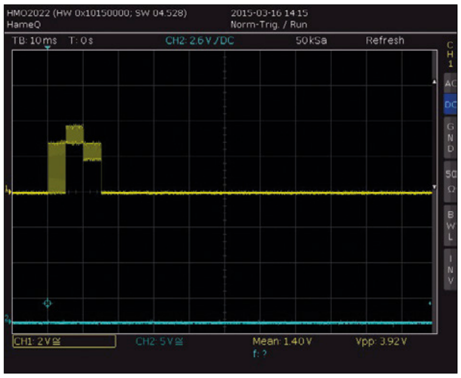


Figure 7. The **roll** parameter. The display shows an ‘r’ and no staircase, which means that a single recording will be displayed.

digital oscilloscope), or at 7.5 V to 12 V from an external mains power supply. The shield is connected to the oscilloscope using two cables. One of these (on K4) carries the analog signal, and the other (on K7) carries a trigger signal. BNC sockets are provided for each signal. Two standard coax cables with BNC plugs at each end can be used to connect the shield to an oscilloscope. **Figure 3** shows the assembled shield, while **Figure 4** shows the compact unit formed when the shield is plugged into an Arduino board.

Set-up and use

The microcontroller stores the recorded samples in its own memory, and so there is no need for the oscilloscope to be connected and kept powered up for hours when recording is taking place. However, the oscilloscope is needed when setting up the various options of the unit, as it provides feedback as to what is going on when the buttons are pressed. With the hardware fully assembled, sketch downloaded into the Arduino, and shield plugged in, the voltage tracker can be connected to the oscilloscope using BNC cables for the data and trigger signals, and power can be applied.

There are four buttons on the shield. Two of these are (or should be) marked ‘–’ and ‘+’. You can use these buttons to adjust an individual setting, and you can step through the settings using S3 (‘set’). A brief press of S4 (‘start’) starts or stops recording, while a longer press (at least 2 s) clears the current recording. So far so straightforward. But how do we see what we are doing?

The Arduino drives the DAC so as to create on the screen of the oscilloscope a representation of the first letter of the setting that can be adjusted, followed by a staircase which shows its current value.

The trigger level of the oscilloscope should be set between 1 V and 4 V, or it can be set automatically. The timebase should be set to 10 ms per division and the vertical scale to 2 V per division. Check that the oscilloscope is configured to expect a 1:1 probe on its input. Once this has been set up, a lower-case ‘t’ (for ‘timebase’) should appear on the display, with a staircase after it. The steps of the staircase should be exactly one division wide: if this is not the case, check the timebase settings of the oscilloscope again. If the oscilloscope display does not have twelve divisions horizontally, then the voltage tracker can be adjusted to suit. Press the ‘set’ button until a lower-case ‘n’ (for ‘number of divisions’) appears and then use the ‘+’ and ‘–‘ buttons to adjust the number of divisions the tracker will use from 10 to 16 (and see also below). Once that is configured, press the ‘set’ button until the ‘t’ appears again. You can now adjust the other parameters according to what you want to measure using the following menus.

The LED on the shield and the LED on the Arduino board remain dark while the device is being configured. They blink during recording and light continuously when the recording is complete.

The five settings that can be configured, and the ranges available for each setting, are as follows.

Table 1: There are seven possible ‘timebase’ settings (see **Figure 5**). The durations given in the table apply to an oscilloscope with twelve horizontal divisions. Since the staircase in the picture has only one step, the horizontal timebase has been set to 1 minute per division.

Table 2: The ‘level’, which is the voltage above which an input will be registered, can be configured in five steps (see **Figure 6**). The picture shows a ‘level’ setting of 3 V. Note that in this

Table 1. **timebase**

Step	Time per division	Recording time
1	1 minute	12 minutes
2	2 minutes	24 minutes
3	5 minutes	1 hour
4	10 minutes	2 hours
5	20 minutes	4 hours
6	30 minutes	6 hours
7	1 hour	12 hours

Table 2. **level**

Step	Offset
0	0 V
1	1 V
2	2 V
3	3 V
4	4 V

Table 3. **roll**

Step	Rolling
0	off
1	on

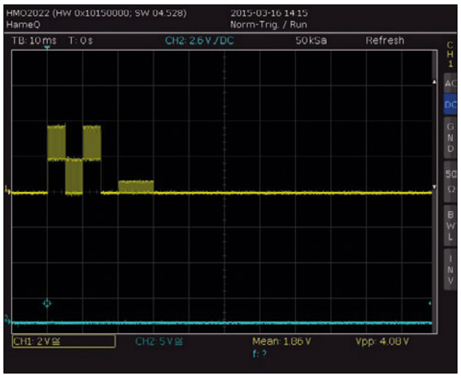


Figure 8. When setting the **Y-scale** (vertical scale) it is important to check the level setting as well. The display shows a ‘v’ and a single step, which corresponds to an allowable voltage range of 2.5 V.

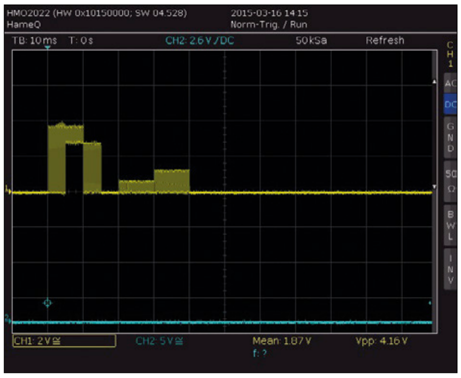


Figure 9. Display when configuring the **number of divisions** to match the oscilloscope in use. The ‘n’ is followed by two steps, corresponding to a twelve-division display.

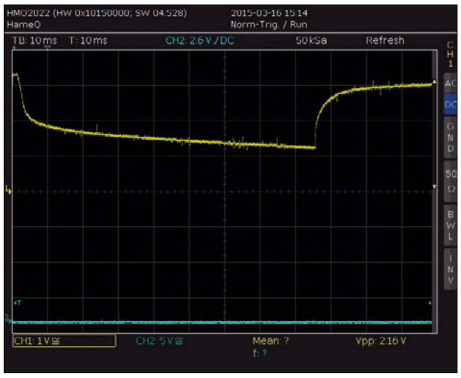


Figure 10. Example display of a discharge curve of a Li-ion rechargeable cell. The total duration of the displayed record is twelve minutes.

case the maximum input swing that can be used is 2 V, in order to stay within the 5 V input range of the microcontroller: see also the vertical setting.

Table 3: This is the ‘roll’ setting. With no steps in the staircase, as shown in **Figure 7**, data recording will stop after a complete screenful has been collected. Setting it to 1 gives a rolling display where a data point is deleted from the left edge of the display when a new one is plotted at the right.

Table 4: The ‘vertical’ setting is a little bit more complicated. In **Figure 8** it is set to 1, which corresponds to a vertical gain of 2. The maximum voltage swing that can be recorded is now 2.5 V, and ‘level’ can be set to a maximum of 2 V if clipping is to be avoided. If ‘level’ is set to 2 V then the input range is from 2 V to 4.5 V. With ‘level’ at zero there is no shift, and the available range is from 0 V to 2.5 V relative to ground.

Table 5: Not every oscilloscope has the same number of horizontal divisions on its display. This setting allows the number of horizontal divisions to be configured on the voltage tracker to match the oscilloscope in use. This affects the total recording time. **Figure 9** shows this value set to 2, which corresponds to the common case where the oscilloscope has twelve horizontal divisions. The highest setting corresponds to 16 divisions, which gives a maximum recording time of 16 hours.

An example

Figure 10 shows the discharge curve of a (used) Li-ion rechargeable cell. Discharging begins at the left-hand side of the screen, and when the voltage falls below 3.3 V discharging is stopped. It is immediately clear how the voltage rises again as the cell recovers from the discharge. An oscilloscope with twelve horizontal divisions was used. The voltage tracker was set to one minute per division, giving a total record length of twelve minutes. The ‘level’ was set to 3 V and the vertical scaling was set to 1/4. With the oscilloscope vertical amplifier set to 1 V per division, this gives a plot with a vertical scale of 0.25 V per division.

Final remarks

It is very easy to build an Arduino shield using only through-hole components. The Arduino is a low-cost platform and so the voltage tracker provides an economical solution to the practical problem of recording long-term signals and displaying them on an oscilloscope. The unit is easy to use, as the oscilloscope display itself is used to show its settings.

The voltage tracker never forgets: all settings are stored in EEPROM in the Arduino microcontroller and are preserved when power is removed. ◀

(150422)

Table 4. y-scale (vertical scale)

Step	Y-scale	Range	Mximum offset
0	1/4	1.25 V	4 V
1	1/2	2.50 V	2 V
2	3/4	3.75 V	1 V
3	1/1	5.00 V	

Table 5. number of horizontal divisions on display

Step	Divisions
0	10
1	11
2	12
3	13
4	14
5	15
6	16

Web Links

[1] www.elektormagazine.com/150442

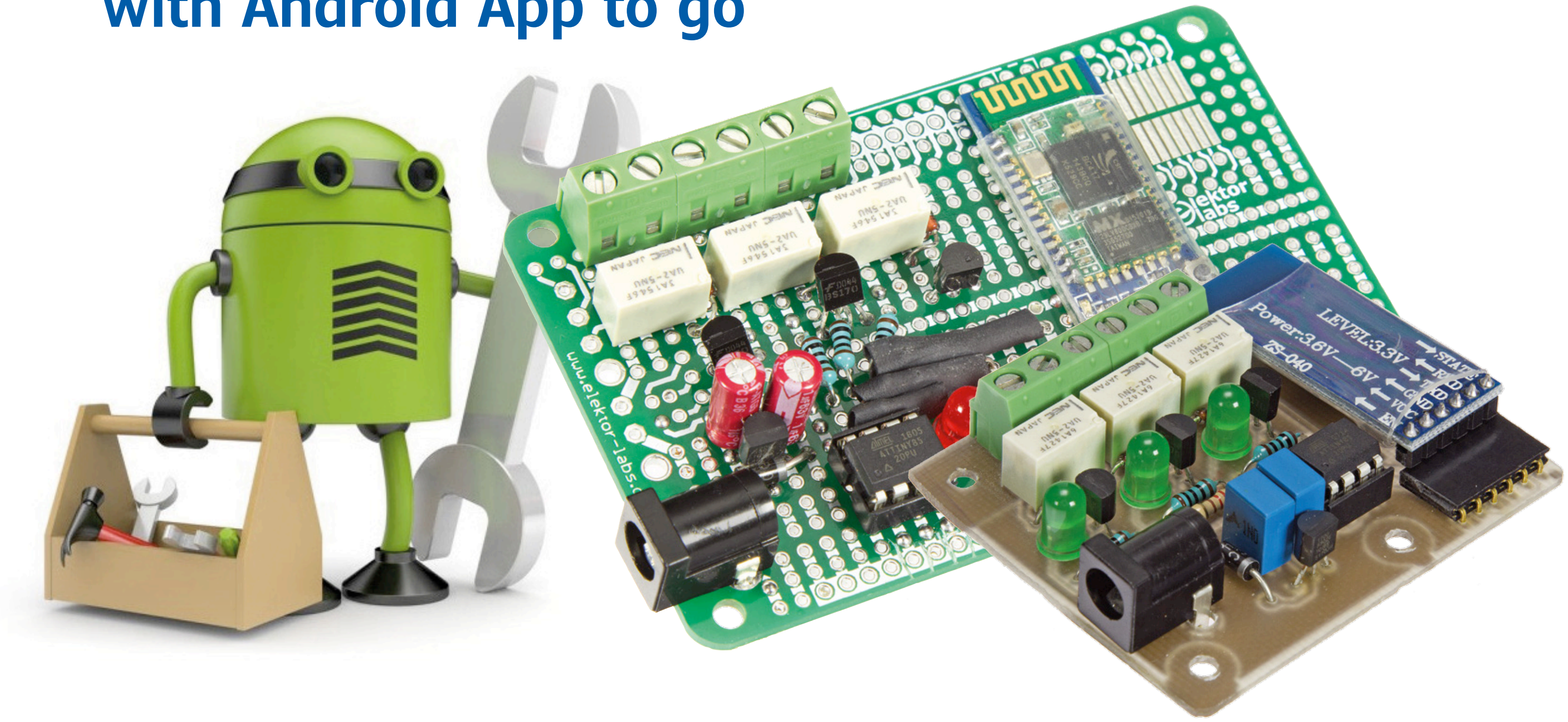
[2] www.elektormagazine.com/labs/voltage-tracker-for-oscilloscope

About the author:

Kurt Schuster works as a hardware and software developer, with a particular interest in AVR assembler. Feedback and questions are welcome at qrt@qland.de.

Bluetooth Controlled Relays

with Android App to go



A friend who knew I was handy with a soldering iron asked if I was willing and able to make a remote controller to operate his three (!) garage doors. I came up with this simple idea to control some relays via Bluetooth using an Android app.

By **Roy Aarts**, Elektor Labs

My initial thoughts for this project centered on the HC-06 Bluetooth module. My colleagues in the Elektor labs kindly pointed out that this was not exactly state-of-the-art. OK, so it may not be Bluetooth Low Energy but at least the older technology I plan to use enjoys support by Android version 4.3 and earlier which means it will operate with a greater range of Android devices. Added to that, this application is not mobile and battery-reliant so low energy devices offer no real advantage.

As you can see from the schematic shown in **Figure 1** the hardware is not at all complicated, you could say it's tiny, in

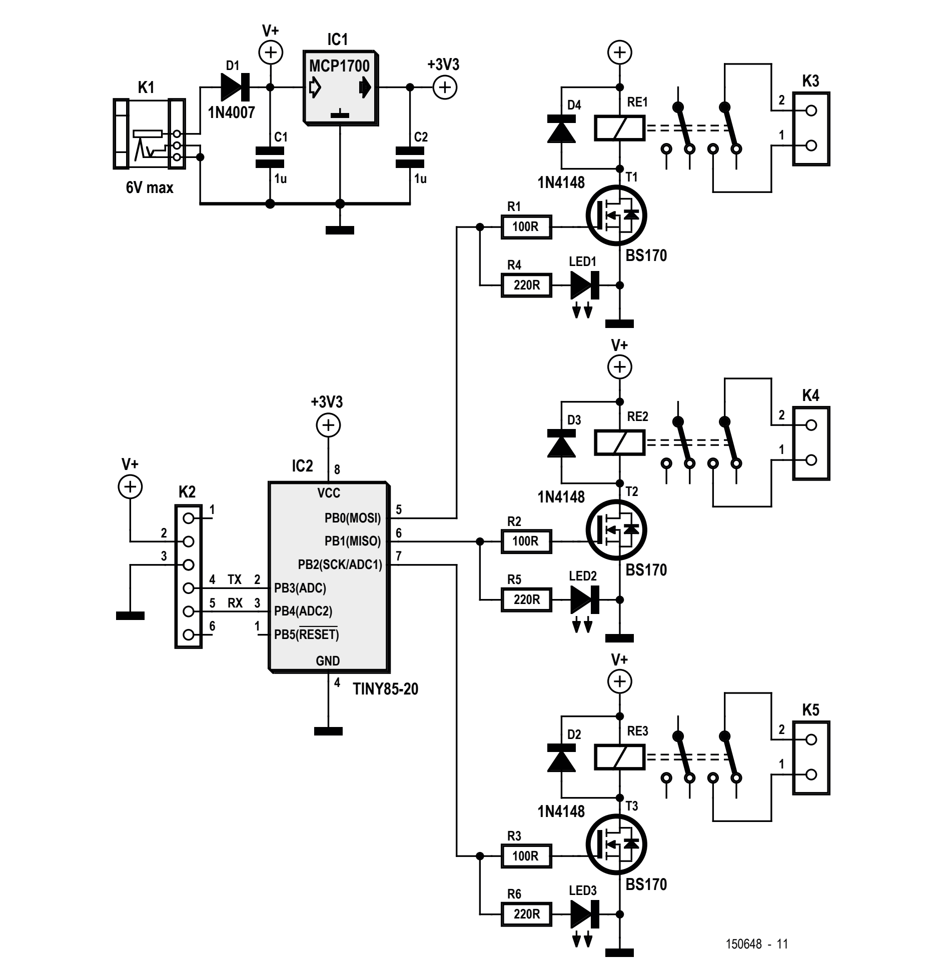


Figure 1. The Bluetooth relay schematic (Bluetooth module is not shown).

fact not much more than an ATtiny85-20 microcontroller and three identical relay driver circuits. The microcontroller I/O pins drive the gates of small-signal MOSFETs via series resistors. The BS170 type MOSFET is a popular choice to provide the additional current necessary to drive relay coils.

The MOSFET gate connection has very high impedance and draws virtually no current from the microcontroller I/O pin so we can drive a green LED directly from this pin also to indicate the state of the relay driver. A green LED begins to conduct when the forward voltage is between 1.9 to 2.5 V, in the worst case this only requires a total output current of just 9 mA when all three outputs are active which is easily supplied by the ATtiny controller I/O lines.

The relays RE1 to RE3 are very compact PCB mounted relays with two changeover contacts. Only one set of contacts is used on each relay in this application. The relay can switch 30 W DC or 37.5 VA maximum AC power at 1 A. The number 5 in the relay part number UA2-5NU indicates that its coil is suitable for use at 5 V DC.

Please Note: This project and the PCB shown below is not suitable for switching AC line voltages!

I chose 3.3 V as the operating voltage of the complete circuit; the Bluetooth module requires Rx and Tx signals at this level for communication with the microcontroller. The MCP1700-3302 low-drop voltage regulator from Microchip is ideal for this application. It can provide a maximum current of 250 mA and can operate with a minimum voltage drop of just 1 V between the input and output supply. This means we can power the circuit from an input voltage of 5 V applied at K1 (0.7 V will be dropped across the protection diode D1). Be sure the regulator input voltage does not exceed 6.0 V! There is in fact no reason why you couldn't power the whole circuit from a powered USB-hub… The HC-06 Bluetooth module (**Figure 2**) (not shown in the schematic) plugs into connector K2. In addition to the supply, the pins provide connections for the data lines TxD and RxD to the controller. Altogether five I/O pins of the ATtiny85 are used for this design which leaves just one unused.

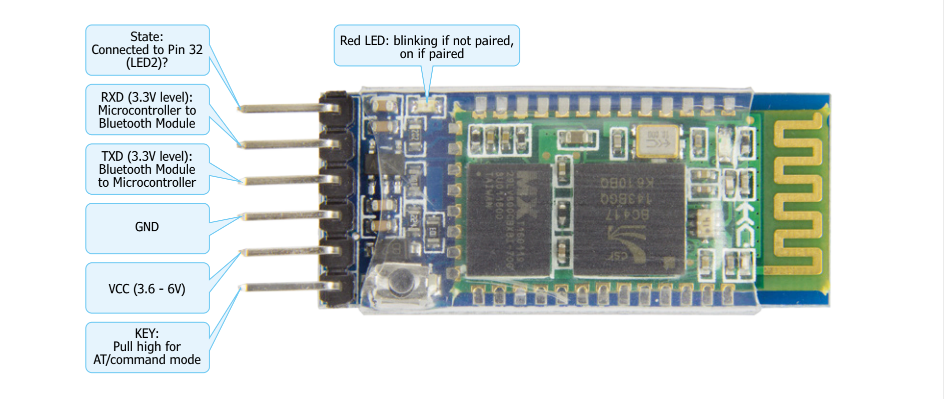


Figure 2. The HC06 Bluetooth module with its connector header pins along the shorter board edge.

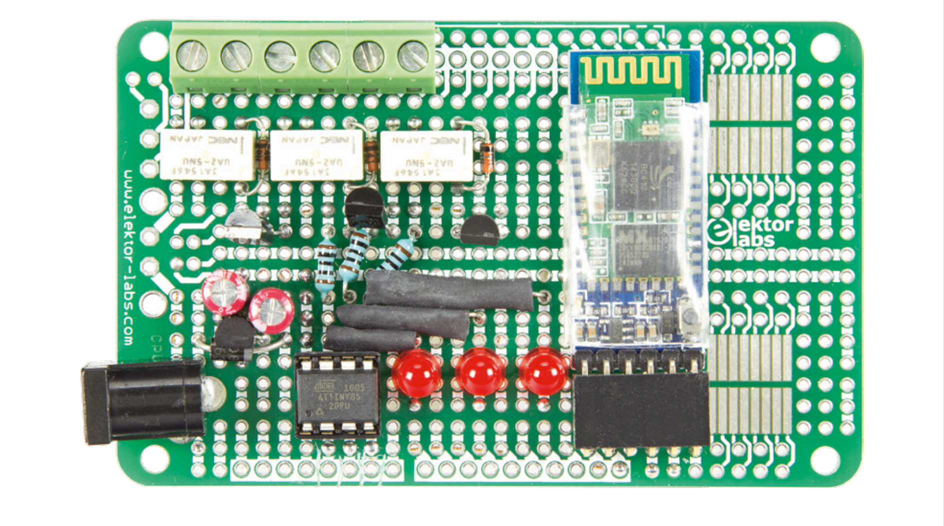


Figure 3. The Bluetooth relay board Prototype quickly built on an ELPB-NG prototyping PCB (the final design incorporates some changes to this prototype).

Build it and fire it up!

To save time and for more convenience I chose to build the prototype using an Elektor prototyping board (**Figure 3**). This ELPB-NG board has already been described in Elektor and you may have already used it yourself for prototype building. As you can see in the title image the finished circuit won't be picking up any prizes in a beauty contest (even with those shiny red LEDs in place of the green ones specified in the parts list).

In fact my colleagues in the Elektor lab (you know, the wise old duffers) were so impressed by the finished design they made a 'proper' board for the project and what a fine job they made of it! You can see from **Figure 4** it's very neat and compact (I need to emphasize again for your safety that the board has **not** been designed to switch AC line voltage).

Before you can start using the Bluetooth board you need to install the associated Android-App which we developed using Android Studio. Go to the Google Play Store and look for the 'Bluetooth relay control' app. After installation your Android phone or tablet needs to be paired with the Bluetooth relay board; go into the Bluetooth settings and search for any devices in range. The relay card should be identified as 'HC-06' in the list. The default PIN code is 1234. The relay board should now respond to commands from the app.
In the App click on the three dots in the upper right hand corner and select 'connect'. In the list of coupled devices select the relay board HC-06. Once your smartphone or tablet is successfully paired the red crosses next to the buttons will change to green ticks. Now you can control the relays using the buttons.

Component List

Resistors
R1,R2, R3 = 100Ω
R4,R5,R6 = 220Ω

Capacitors
C1,C2 = 1µF, 0.2’’ pitch

Semiconductor
D1 = 1N4007
D2,D3,D4 = 1N4148
LED1,LED2, LED3 = LED, 5mm, green
T1,T2,T3 = BS170
IC1 = MCP1700-3302E/TO (Microchip)
IC2 = ATtiny85-20PU, programmed, Elektor Store #150648-41 [1]

Miscellaneous
RE1,RE2,RE3 = 5V coil, 230VAC, 1A contacts (IMO Precision Controls UA2-5NU)
K1 = DC adaptor socket, 2.5mm center pin
K2 = 6-way socket for pinheader (1x6)
K3,K4,K5 = 2-way PCB screw terminal block, 5mm pitch
HC-06 Bluetooth module (JY-MCU V1.5)
PCB # 150648-1v1.3 or ELPB-NG prototyping board (150180-1)
Firmware 150648-11 [1]
Android App, free from Google Play Store [2]

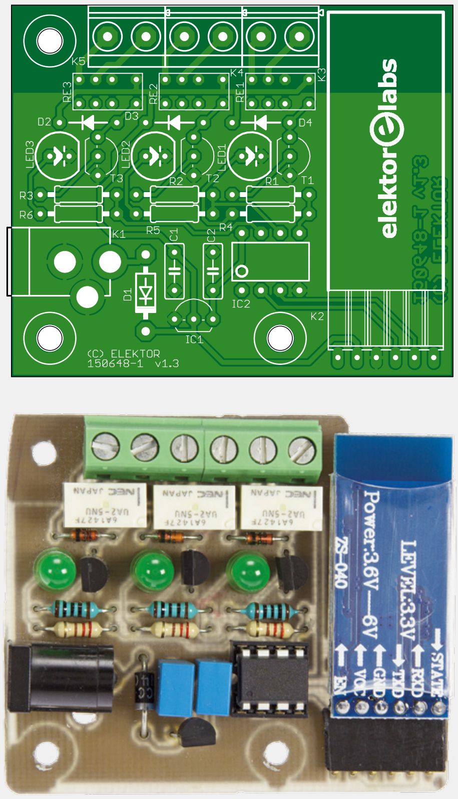


Figure 4. The Bluetooth relay assembled on its purpose-built PCB (The photo shows the prototype, the layout plan of the final version). Please note: the PCB is not designed or rated to switch AC line voltage!

Each relay contact can be made to close or open and also can be programmed to make or break contact momentarily. In this pulse mode the pulse length can be defined from 0.1 s to 5 s in steps of 0.1 s. Use the check box next to the button to select whether the relay acts in on/off (changeover or toggle mode) or in pulsed mode. The check box must be deactivated for operation in pulsed mode.
The relay contact will then either provide a close pulse or open pulse depending on the relay state when the check box is deactivated. The button color shows the relay contact state. When it’s grey the contacts are open and when it’s green they are closed.

By clicking on the three dots in the upper right hand corner you can get to the ‘settings’ page where it’s possible to change some parameters such as pulse contact period from 0.1 to 5 s, the module name and also the module PIN.

Software for the controller and smartphone

The software for this project consists of firmware running on the microcontroller and an app for the smartphone. The source files for both programs are available to download from the Elektor project web page [1]. You can also order a ready-programmed controller from the same page if you want to save the hassle of programming your own.
The **ATtiny Code** was written in C++ using the Arduino IDE. On power up the microcontroller’s firmware initializes the outputs and begins serial communication with the Bluetooth module. In the main loop the program looks for received characters. When data is detected it continues to read incoming characters until a linefeed (LF) character is detected. This tells the program that the received message is complete.
To find out the information contained in the received data the program uses a Switch-Case statement. In each case the software reads the data following the first character. The different ‘Cases’ and the meaning of the possible characters are given in **Table 1**.
The character ‘t’ is used to indicate whether the relay contacts will just changeover or provide a pulse. This configuration will be stored in the first three bits of the variable called ‘states’ while the next three bits contains the current state of the outputs. For example if the first bit is high output 1 operates as a changeover contact and the state of the fourth bit will be toggled and written to the output.
In pulse mode this first bit will be zero.

Table 1. Commands sent from the app interpreted in the controller firmware.

Command	Function	Example
c	Used when the smartphone is paired with the Bluetooth relay board. The board replies with the current state of its relay outputs, toggle status and pulse length.	c
p	Assigns a PIN code to the Bluetooth module.	p1234 (changes the PIN to 1234)
n	Changes the Bluetooth module name.	nHC-06 (changes the name to HC-06)
t	Defines toggled or change over contact.	t11 (output 1 changes over) t20 (output 2 is pulsed)
d	Used to define the relay contact pulse time (duration).	d1000 (sets the pulse duration to 1 s = 1000 ms)
b	Controls an output.	b1 (controls output 1)

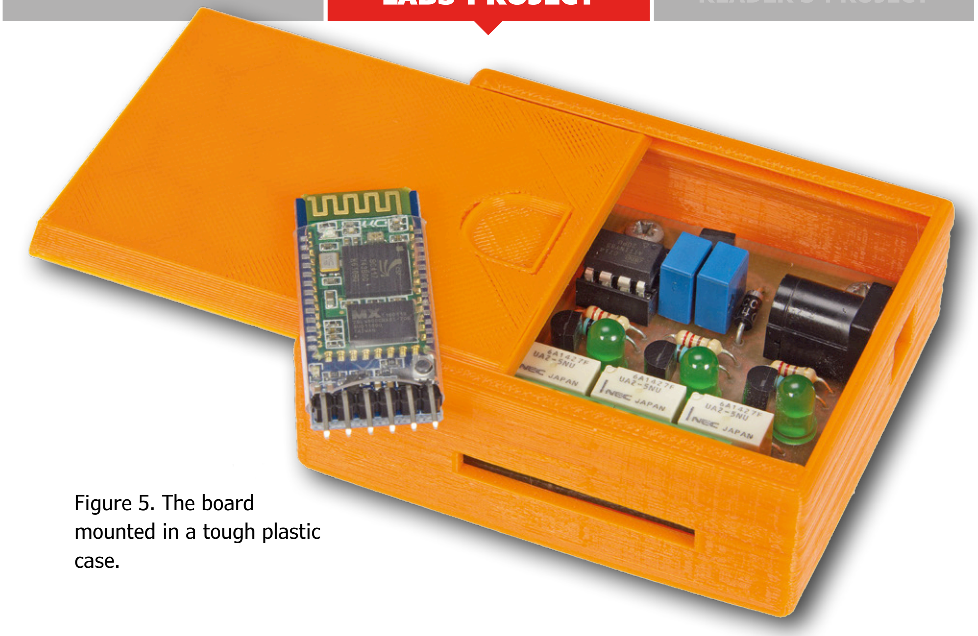


Figure 5. The board mounted in a tough plastic case.

The state of the fourth bit will be toggled and then written to the output. Now two other variables come into play: Firstly a bit is set in 'controllingOutputs' to indicate that the output is pulsed and secondly the pulse duration value is stored in 'pulseTime'.

Another function keeps track of the elapsed time of the pulse by monitoring the 'controllingOutputs' variable. When the bit is set it compares the elapsed time of the pulse. When this value is bigger than the set pulse duration value the function switches the output back to its original state.

The **Smartphone App** is written in Java and was developed using Android Studio. The app gives you the option to provide an output pulse or a continuous make/break level. To control the relay board the app sends a message containing a character followed by data and terminated by a line feed character (LF = `\n`, see source file `MainActivity.java`). When the pulsed output mode is selected the pulse length can be defined. The module's name and PIN code can also be changed (`settings.java`/`MainActivity.java`). In order to communicate with the board its first necessary to pair the Bluetooth module with the smartphone. The app can then select from the list of paired devices (`deviceList.java`) to control the board. ⏮

(150648)

Web Links

[1] Project page: www.elektor.com/150648

[2] Android App: https://play.google.com/store/apps/details?id=com.elektor.attinybluetoothio

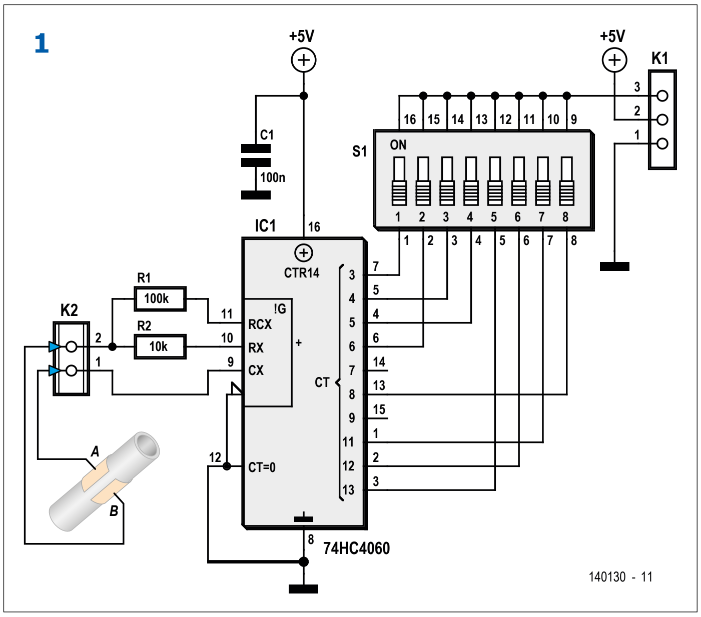
Capacitive Liquid Detection

With the help of Arduino

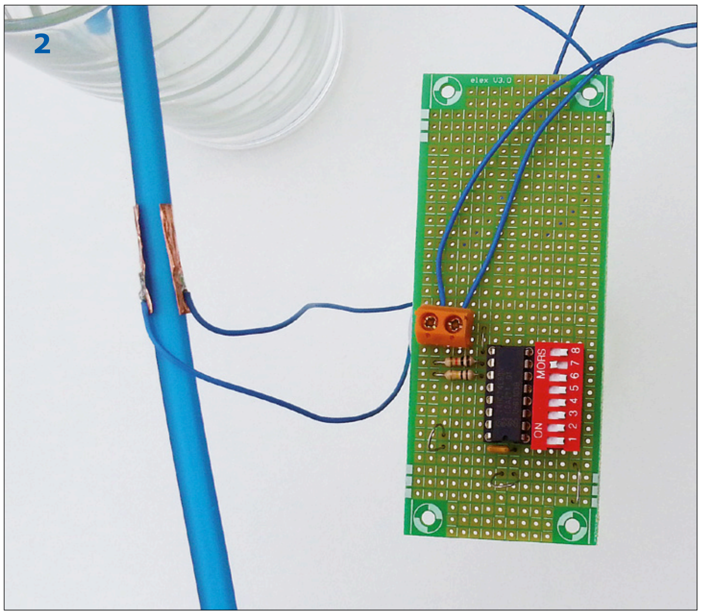
By **Paul Cordonnier** (Belgium)

Some time ago, Elektor published my siphonic rain gauge with no moving parts, in which the water that passes when a little siphon empties is detected by short-circuiting two electrodes [2]. At the same moment, articles were in preparation showing that capacitive detection would have a decisive impact [3]. Here I investigate if the "approaches" can be joined.

1



2



The idea of a technical bridge between the above publications only appeared in the comments posted on the Elektor Labs website. That was where I read the remark from a reader (I later learnt it wasn't just *any* reader) suggesting the capacitive approach [4] for detecting the water passing through the pluviometer drain instead of a short-circuit on the electrodes. Then I recalled that a few years earlier I had myself successfully built a capacitive-detection water level monitor using a Basic Stamp. Why didn't I think of this idea for the rain gauge? The sparks of creativity are capricious and don't just pop up to order. Whence the interest of working on ideas together. As it happens, my neighbor and the author of the comment was none other than David Ardouin, whose recent articles testify to the capabilities of capacitive.

I tested the idea successfully with the help of two small pieces of self-adhesive copper foil (which you can find from Farnell, in particular) stuck opposite one another on the flexible tube. The advantage of this ultra-simple process, which can be applied in other fields for detecting the presence or absence of liquids within pipes, is to eliminate drilling through the wall to insert the electrodes, as I had done for the rain gauge. No Stamp this time, I used an Arduino: a little "`pulseIn`" and a few lines of code later, and the die was cast! And the right-hand corner of my forehead is a little flatter than before.

I hope like me you'll learn this lesson of the rain gauge: instead of falling flat on your face all alone with some stubborn idea, take a few minutes to talk about it on Elektor Labs website.

There's water in the "condenser"

The type 4060 logic counter/divider (**Figure 1**) is cheerfully open to the world: its three pins RCX, RX, and CX are accessible from outside, and that's exactly where we connect the capacitor made from the two pieces of copper foil stuck on the walls of the pipe to be monitored. The capacitance of this assembly will not be the same, depending on whether there is water in the pipe or not. So with that, the oscillator frequency will change too. To find out what is flowing or not in the pipe, all we have to do is measure the length of the pulses at the counter/divider output. Do I need to point out that this will work with rubber or silicone, but not with metal pipes?

The oscillator signal is not used as is for discrimination, but its frequency is first reduced using the divider in the 4060. The 8-way DIP switch lets us pick off one of the division frequencies available. Note: only turn one of S1's eight switches on at a time. This squarewave, applied via K1 to an Arduino, will be "analyzed" using a little detector evaluation program (**Listing 1**).

The "cleanness" of the information obtained will depend on the construction of the detector, which affects the discrimination between the two extreme states (presence or absence of liquid). The absolute values are trivial, the only thing that counts is the detection threshold mid-way between them.

By way of an example, on the Elektor Labs prototype, using a 1 nF capacitor in place of the detector, the frequency noted on Q7 (pin 6 of IC1) was of the order of 2.5 kHz.

Using the detector in the photo (**Figure 2**) — the pipe is 6 mm OD and 4 mm ID, and the two self-adhesive copper strips are around 20 × 7 mm — the difference in frequency obtained was of the order of 1.5%. It's not much — but for a microcontroller, it's doable.

Longer strips ought to make it possible to increase the sensitivity. On my own prototype, tested using water and diesel, at appreciably lower frequencies, I obtained a clearer distinction (nearly 10%). The frequency *with no water* of 166 Hz gave `pulseIn` = 3128, while 151 Hz *with water* gave `pulseIn` = 3229. I adopted 3178 as the switching point.

To be quite honest, here we're only at the experimental stage; it's going to be necessary not only to refine this measurement, but also to carry out periodic calibration of the detector to compensate for likely drifts. Nonetheless, the project seemed serious enough for us to offer a PCB layout, the EAGLE format files for which can be downloaded [6].

Now it's your turn now to tell us [1] about your own experiments.

(140130)

Web Links

[1] www.elektormagazine.com/labs

[2] Siphonic Rain Gauge: www.elektormagazine.com/120554

[3] Touch-2-Switch, multi-purpose wall-mounting switch: www.elektormagazine.com/130272

[4] http://goo.gl/TcCE94

[5] Pattern Lock: www.elektormagazine.com/120579

[6] www.elektormagazine.com/140130

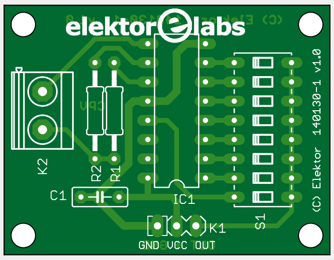
Component List

Resistors (0.25W)
R2 = 10kΩ
R1 = 100kΩ

Capacitors
C1 = 100nF

Semiconductors
IC1 = 74HC4060N

Miscellaneous
K1 = 3-pin pinheader
K2 = 2-way PCB screw terminal, 0.2'' pitch
S1 = 8-way DIP switch block
PCB # 140130-1 v1.0

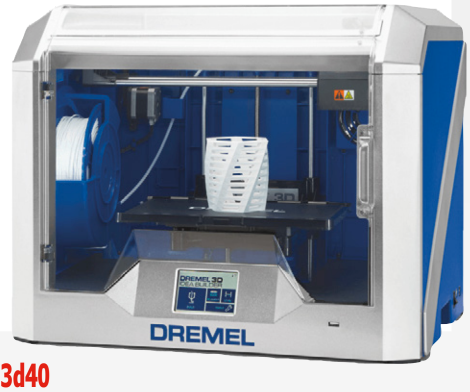


Listing 1.

```
int freqPin = 7;  // probe input
int beepPin = 6;  // buzzer
unsigned long duration;

void setup()
{
Serial.begin(9600);
pinMode(freqPin, INPUT);
pinMode(beepPin, OUTPUT);
}

void loop()
{
duration = pulseIn(freqPin, HIGH);
Serial.println(duration);

//  delay(1000);
if (duration >  3200)
{
tone(beepPin, 600, 100);
}

}
```

welcome in your ONLINE STORE

EDITOR'S CHOICE



The new 3D printer from Dremel is a big improvement on its predecessor. It uses the same type of filament used in Dremel's first 3D printer. The PLA filament material is 1.75 mm in diameter and biodegradable. The printing head has undergone a complete redesign and is now much larger and more robust. This produces a much improved filament melt phase making the Idea Builder 3D40 suitable for professional applications as well as for home use. Overall the Dremel 3D40 is the best 3D printer we have tested so far. It is very user friendly and the semi-automatic levelling works well and is an excellent addition to produce high quality printed models. With the 3D40 Dremel has built upon the strengths of its very good 3D20 printer and produced a worthy successor which is both user friendly and a robust, reliable 3D printer for PLA filament.

René Bohne
Maker on Duty

www.elektor.com/dremel-3d-idea-builder-3d40

Elektor Bestsellers

1. Red Pitaya for Test and Measurement
   www.elektor.com/red-pitaya-book

2. Elektor Uno R4
   www.elektor.com/elektor-uno-r4
3. Elektor SDR Reloaded
   www.elektor.com/sdr-reloaded
4. CAN Projects with ARM and Arduino
   www.elektor.com/can-projects
5. BBC micro:bit
   www.elektor.com/bbc-micro-bit
6. Swiss Pi
   www.elektor.com/swiss-pi
7. Bat Detector PLUS
   www.elektor.com/bat-detector-plus

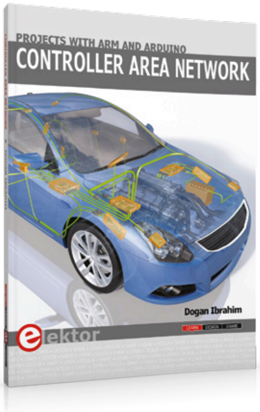
Arduino Uno – 45 Projects for Beginners and Experts

This book covers a series of exciting and fun projects for the Arduino, such as a silent alarm, people sensor, light sensor, motor control, internet and wireless control (using a radio link). Contrary to many free projects on the internet all projects in this book have been extensively tested and are guaranteed to work!

member price: £26.95 • €31.46 • US $36.00

www.elektor.com/arduino-projects

CAN Projects with ARM and Arduino



This book details the use of the ARM Cortex-M family of processors and the Arduino Uno in practical CAN bus based projects. Inside, it gives a detailed introduction to the architecture of the Cortex-M family whilst providing examples of popular hardware and software development kits. Using these kits helps to simplify the embedded design cycle considerably and makes it easier to develop, debug, and test a CAN bus based project.

member price: £28.95 • €33.75 • US $38.00

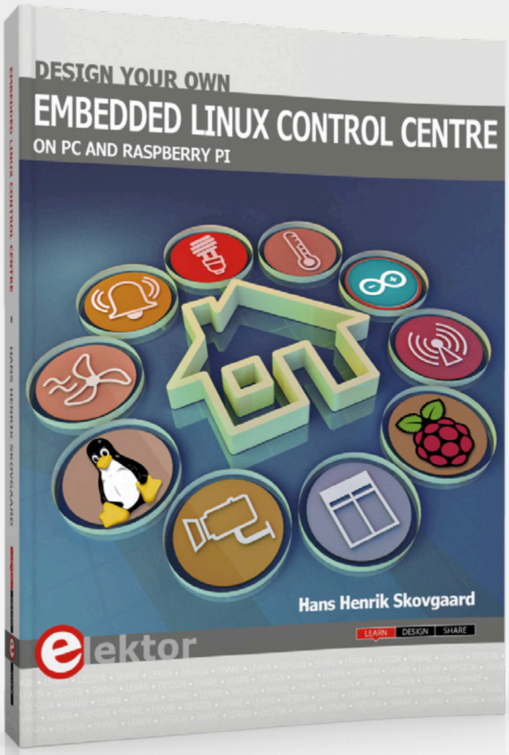
www.elektor.com/can-projects

DVD Elektor Magazine 2010 - 2014

This DVD-ROM contains all circuits and projects published in Elektor Magazine's year volumes 2010 through 2014. Since 1975, specialist magazine 'Elektor' has been in the forefront of electronics and computer technology through the publication of repeatable, professionally designed circuits and projects. Elektor constantly reports on technological and market-relevant developments, as well as on new products and technologies.

member price: £51.95 • €62.10 • US $70.00

www.elektor.com/elektor-2010-2014



This book is all about building your own DIY home control system. It presents two innovative ways to assemble such a system: By recycling old PC hardware - possibly extending the life of an old PC, or by using Raspberry Pi. In both cases, the main system outlined in this book will consist of a computer platform, a wireless mains outlet, a controller and a USB webcam - All linked together by Linux. By using the Raspberry Pi in conjunction with Arduino (used as an advanced I/O system board), it is possible to construct a small, compact, embedded control system offering enhanced capacity for USB integration, webcams, thermal monitoring and communication with the outside world.

MEMBER PRICE: £29.95 • €34.35 • US $39.00

www.elektor.com/elcc3

Build Your Own Home Control System

Limited time offer for GREEN and GOLD Members:
20% Discount
plus free shipping!

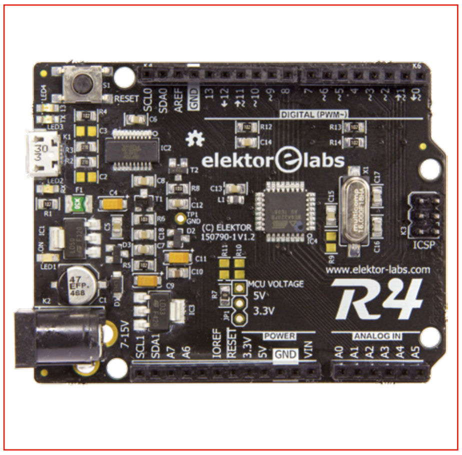
pi-top

DIY Laptop Kit for Raspberry Pi!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

Elektor Uno R4



When a manufacturer releases a B version of an existing product the differences between the old and the new are usually small. Not so in the case of Atmel's ATmega328P, the processor at the heart of the Arduino Uno R3. The B-type of this MCU has so many extra features that a new revision of the R3 is justified: Elektor Uno R4. The most striking difference between the Elektor Uno R4 and the Arduino Uno R3 are the four extra pins on the "power" connector and its second USART.

member price: £22.95 • €26.95 • US $31.00

www.elektor.com/elektor-uno-r4

C Programming with Arduino

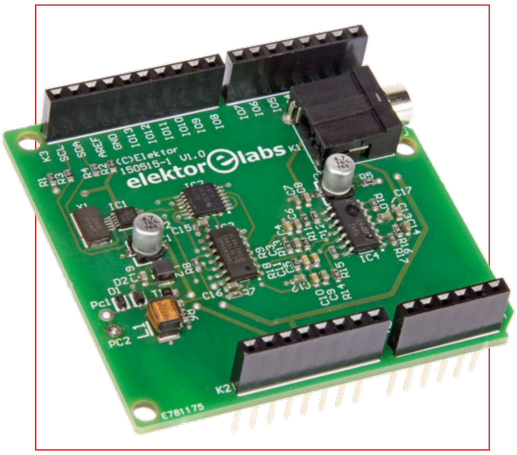
AVR MICROCONTROLLERS AND ATMEL STUDIO FOR
C PROGRAMMING WITH ARDUINO
Warwick A. Smith
elektor

If you would like to learn the C language to program microcontrollers, then this book is for you! Arduino is the hardware platform used. They contain the popular AVR microcontrollers from Atmel. Atmel Studio is used as the development environment for writing C programs for AVR microcontrollers. It is a full-featured integrated development environment (IDE) that uses the GCC C software tools for AVR microcontrollers and is free to download.

member price: £34.95 • €40.46 • US $46.00

www.elektor.com/c-programming

Elektor SDR Reloaded



This Arduino shield is a remake of our famous SDR project published in 2007. A Software Defined Radio is a universal tool in RF technology circles, one that can also be put to use for making measurements. The characteristics of the receiver are defined in software, which now gives us the opportunity to use an Arduino Shield as a front-end.

member price: £22.95 • €26.96 • US $31.00

www.elektor.com/sdr-reloaded

Join the Elektor Community

Take out a GOLD Membership now!

elektor
LEARN DESIGN SHARE
Starter Kits Paradise
a survey of embedded platforms and dev kits
Elektor Uno R4
Four pins make all the difference
ARCHOS

GOLD MEMBERSHIP

- ✔ 6x Elektor Magazine (Print)
- ✔ 6x Elektor Magazine (PDF)
- ✔ Access to Elektor Archive (Thousands of Articles)
- ✔ Access to over 1,000 Gerber files
- ✔ Elektor Annual DVD
- ✔ 10% Discount in Elektor Store
- ✔ Exclusive Offers

Also available:

The all-paperless GREEN Membership!

GREEN MEMBERSHIP

- ✔ 6x Elektor Magazine (PDF)
- ✔ Access to Elektor Archive (Thousands of Articles)
- ✔ Access to over 1,000 Gerber files
- ✔ 10% Discount in Elektor Store
- ✔ Exclusive Offers

www.elektor.com/member

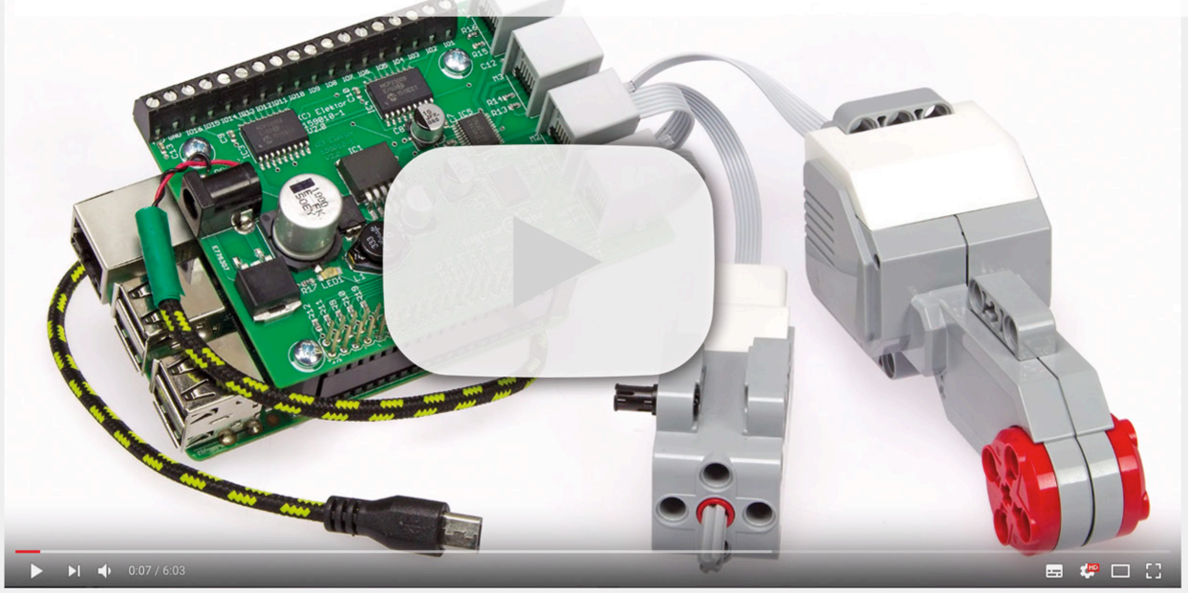
layar
INTERACTIVE PRINT

1 Download the free Layar App

2 Scan this page

3 Discover interactive content

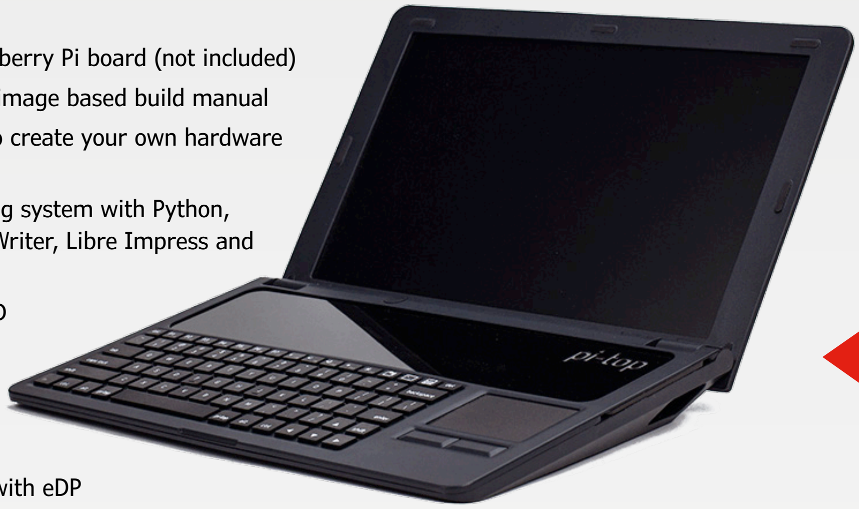
SCAN THIS PAGE AND WATCH THE VIDEO



Put your Raspberry Pi board into a pi-top and you're ready to start building your own customized laptop. The pi-top takes little time to build and does not require a soldering iron. You can start to experiment with electronics or create printed circuit boards immediately.

Features

- Compatible with any Raspberry Pi board (not included)
- Kit component guide and image based build manual
- Learn transferable skills to create your own hardware projects
- Raspbian (Jessie) operating system with Python, Scratch, Minecraft, Libre Writer, Libre Impress and Libre Calc software
- Chromium, 3D Slash, CEED Universe, Now software
- Accessible from anywhere, anytime pi-top Kit Contents
- 13.3 inch HD LCD Screen with eDP interface with 1366 x 768 resolution
- Smart battery with 10+ hours run time, 2-Wire SMBus V 2.0 interface
- US Keyboard (fully programmable) and trackpad
- 8 GB SD card with pre-installed pi-top OS



Please note: The pi-top is optimised for use with Raspberry Pi 3, and does not include a WiFi dongle.

MEMBER PRICE: £236.95 • €278.95 • US $311.00

www.elektor.com/pi-top

Build Your Own Home Control System

Limited time offer for GREEN and GOLD Members:
20% Discount
plus free shipping!

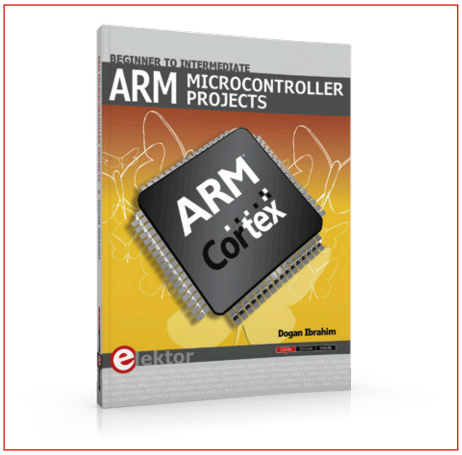
pi-top

DIY Laptop Kit for Raspberry Pi!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

ARM Microcontroller Projects



This book makes use of the ARM Cortex-M family of processors in easy-to-follow, practical projects. It gives a detailed introduction to the architecture of the Cortex-M family. The architecture of the highly popular ARM Cortex-M processor STM32F107VCT6 is described at a high level, taking into consideration its clock mechanisms, general input/output ports, interrupt sources, ADC and DAC converters, timer facilities, and more.

member price: £30.95 • €35.96 • US $41.00

www.elektor.com/arm-projects

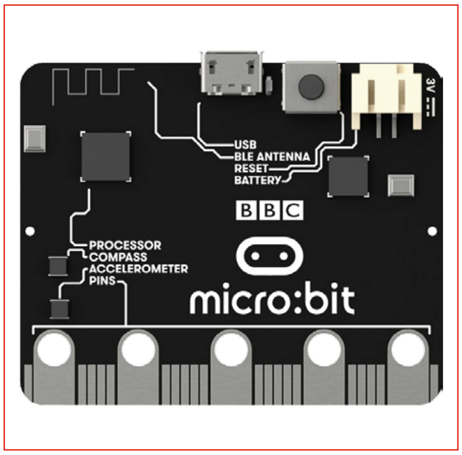
Red Pitaya for Test & Measurement

The Red Pitaya is a credit card-sized, open-source test and measurement board that can be used to replace most measurement instruments used in electronics laboratories. This book aims to teach the principles and applications of basic electronics by carrying out real experiments using the Red Pitaya. Many fun and interesting experiments are included. The book also makes an introduction to visual programming environment.

member price: £26.95 • €31.45 • US $36.00

www.elektor.com/red-pitaya-book

BBC micro:bit



BBC micro:bit is a small, codable device that can be programmed to light up using a series of tiny LEDs. A non-intimidating introduction to programming and making – switch on, program it to do something fun – develop new ideas. BBC micro:bit combines a pocket-sized coding device featuring several sensors and LED's, with a website full of coding languages, helping you get creative – from making your own games to taking selfies, the possibilities are endless.

member price: £12.95 • €15.26 • US $18.00

www.elektor.com/bbc-micro-bit

Welcome to the SHARE section

By **Thijs Beckers** (Elektor Netherlands)

Nixie but not Nixie #2

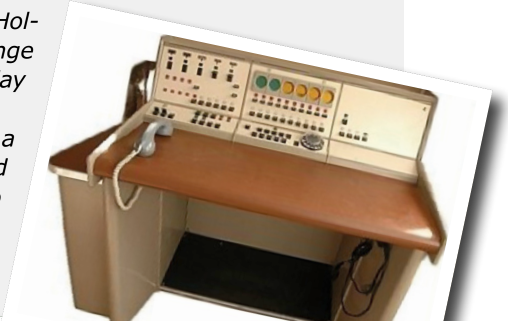
In my SHARE opening words in the previous edition I shared with you a small project by translator Kees de Groot: a Nixie clock without Nixie tubes. Instead of the tubes he wanted to use digits that are laser-cut from acrylic sheet and illuminated from the side. I have had a number of responses from readers on this and in particular I would like to share a contribution from Mr Welschen.

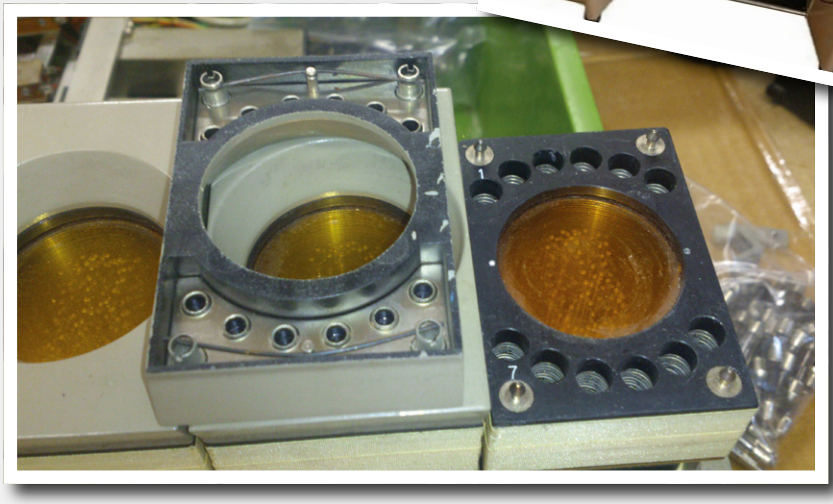
"The idea of obtaining the same effect as Nixie tubes using another method is not new. Some time in the 1960's or 70's Philips Telecommunication Industries in Hilversum, Holland, produced an exchange console for telephone operators with such an imitation Nixie display for displaying the telephone numbers. It operated as follows:

Ten round perspex (acrylic sheet) plates, stacked together with a small air gap between the plates. Each plate had small dimples drilled in the shape of a digit. The dimples where positioned such that no two dimples where behind one another. Ten small lamps were positioned around the plates with a mask such that one lamp would illuminate the side of one plate. By illuminating the plate, the light was scattered by the dimples, which made it visible from the front and the corresponding digit was shown. A number of these units adjacent to each other formed the telephone number. The colors were different for incoming and outgoing numbers. This was realized by using Perspex of different colors (green and amber).

How the reflection on the sides was enhanced I do not know. From products from a later date, which also operate with the internal reflection of Perspex, I know that simply painting the smooth side white works well. These days we could use transparent Perspex and illuminate it with RGB LEDs, but the color saturation would still be somewhat weak.

If you would like to use the effect that captures the light that falls on the surface and becomes concentrated at the edges then you can use acrylic sheet that has a pigment filler. The pigment uses the incoming light and re-radiates it again (in its own color). This technique is used, for example, in solar panels. A strip of solar panels is mounted along the edge of a plate, which now receive more light than if they had been placed in direct sunlight. If you would like to know more about this: http://po.st/share-nixie2. On the we I already found evidence of successful experimental demonstration of a few square meters."





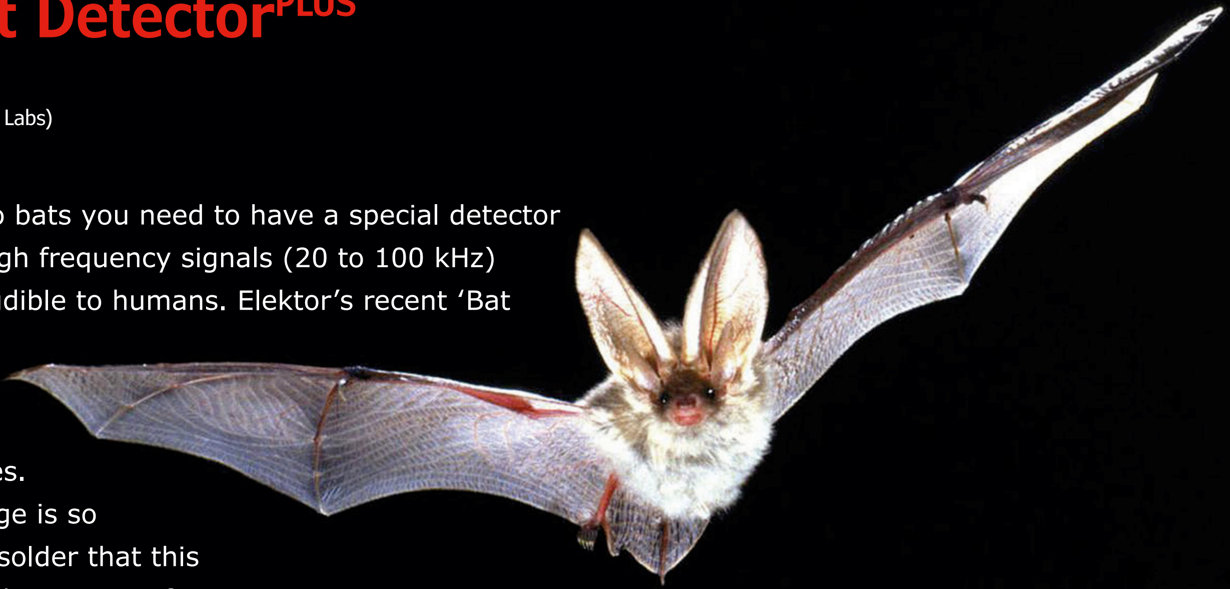
In any case, we have received another project idea. More about this soon.. ⏮

(160183)

Minuscule MEMS Microphone

for the Bat Detector^{PLUS}

By **Harry Baggen** (Elektor Labs)

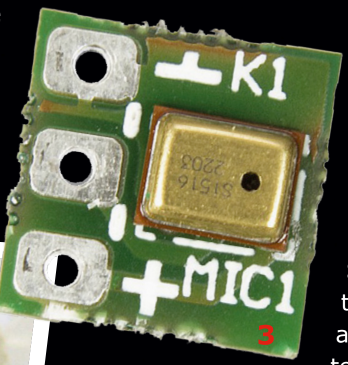


To be able to listen to bats you need to have a special detector that translates the high frequency signals (20 to 100 kHz) to sounds that are audible to humans. Elektor's recent 'Bat Detector^{PLUS}' uses a special microphone that is suitable for these high frequencies. But the device package is so small and difficult to solder that this requires a considerable amount of care.

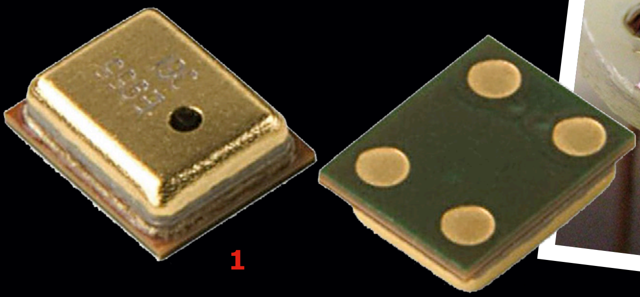
In the Bat Detector^{PLUS} published in edition 4 / 2016 the builder can choose from several types of microphones, namely a 2- or 3-wire electret type or a MEMS type (Micro-Electro-Mechanical System). Microphones specified by the manufacturer for use at frequencies above 20 kHz are very difficult to find, therefore the typical approach is simply to try whether a particular type is suitable for this purpose. The MEMS microphone type SPU0410LR5H made by Knowles is one of the few devices that does in fact have a sensitivity curve that goes up to 80 kHz in the datasheet. In the bat detector we used a comparable type, the SPU0410HR5H. While the device looks eminently suitable for this application, it's housed in a package that is extraordinarily unpleasant for manual mounting (**Figure 1**). On the top is a small hole in the metal housing for the microphone element and on the bottom are four solder pads. This has obviously been designed for SMD assembly, but not every hobbyist has experience with these. The author of this circuit, Kenneth Horton, has developed a very clever solution by making a small circular circuit board with an opening in the center. He then mounted the microphone upside down in this opening, so that the hole in the microphone package has an unobstructed 'view' from the other side of the board. He subsequently threaded a few pieces of solid copper wire through the nearby through-hole pads and bent them around towards the solder pads (**Figure 2**). And now all that remains is to solder it all together! A very clever solution indeed, but unfortunately this method is not particularly user friendly.

In order to make the construction of the Bat Detector^{PLUS} a little easier, a separate circuit board for the MEMS microphone was designed here at Elektor Labs, and at the same time a decoupling capacitor was added (**Figure 3**). A panel of 11 x 11 of these tiny circuit boards were populated by hand in our lab and then soldered in an SMD oven. This was ready in a jiffy, after all, our lab has by now accumulated quite a bit of experience with SMD soldering! Once this was done, all the assembled boards were tested for proper operation, after which they were sent to our warehouse to be sent to our readers who would like to build the Bat Detector^{PLUS}. They now only need to assemble the circuit board with the through-hole components, and that is not nearly such a difficult job! ⏮

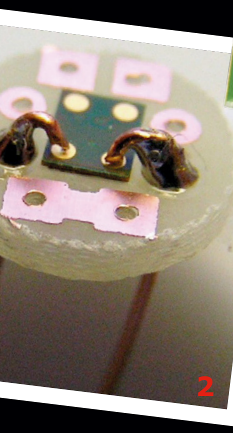
(160083)



3



1



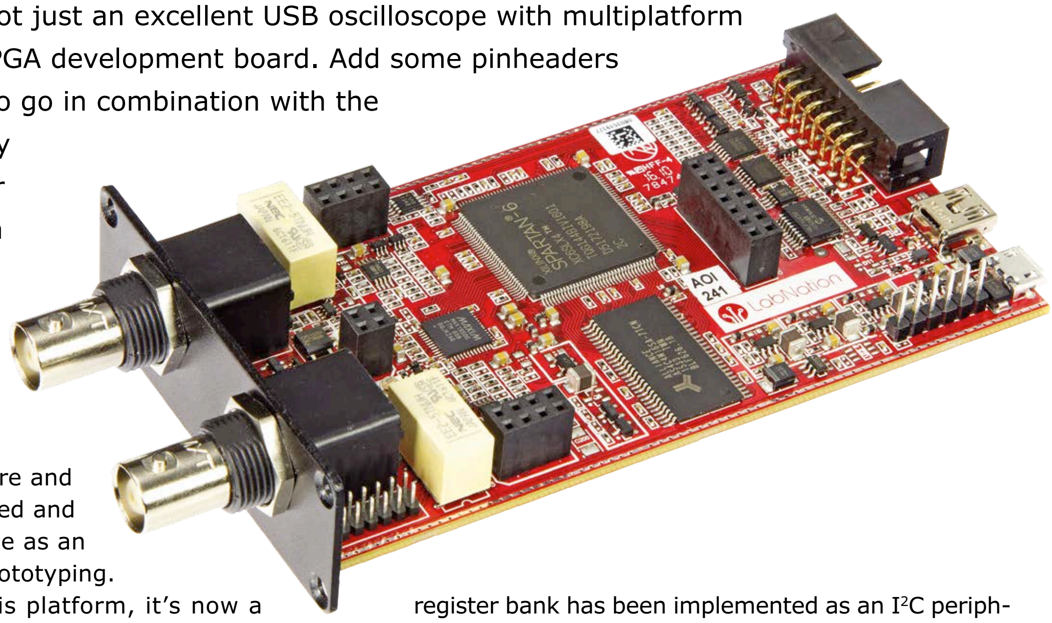
2

SmartScope Maker Kit

Open-source platform for FPGA development

By **Riemer Grootjans**, LabNation (Belgium)

The renowned SmartScope is not just an excellent USB oscilloscope with multiplatform support, it's also great as an FPGA development board. Add some pinheaders on the board and you're good to go in combination with the user-friendly software especially written for this purpose. Elektor offers a special Maker kit with a preconfigured Smartscope, two programmers and the necessary cables.



Recently, LabNation made its software and firmware stack open-source, optimized and presented to position the SmartScope as an ideal hardware platform for FPGA prototyping.
With access to the VHDL files of this platform, it's now a breeze to compile and upload your own FPGA code into the SmartScope, and use its USB bridge to control it from your PC, tablet or phone.
In cooperation with LabNation Elektor has put together a SmartScope Maker Kit, which consists of a unique SmartScope version with preassembled pinheaders, two programmers and several connection cables. This limited-edition kit is available at a special price in the Elektor Store.

Who can benefit from this kit?

Electronics enthusiasts with experience in CPU-based development kits who are interested in taking the step towards FPGA development. There are numerous tutorials on the web to get you started with writing your first 'Hello world' VHDL module (this article actually contains one). However, what you need next is an FPGA to upload your program to. This is where the SmartScope Maker Kit comes in, as it contains an FPGA as well as the required programmers to upload the program to that FPGA.
Additionally, using the SmartScope open-source platform you start off with a full communication chain from within your FPGA — over USB, all the way to an app running on your PC/tablet. This allows you to adjust parameters of your VHDL module in real time.
Experienced electronics engineers will recognize the need for such control of registers in their VHDL modules, and will therefore appreciate the ease of integration of this USB link. Moreover, by using the open-source PC libraries you can script access to these VHDL registers, for which the code of the app can be used as a starting point. Finally, because access to this register bank has been implemented as an I^2C peripheral, your VHDL project can be ported to other boards easily.

Why would you be interested in FPGA development?

CPU-based kits allow you to control most simple peripherals, such as I^2C sensors, a DC motor or small LCD displays. However, if you want to take the next step and control higher-bandwidth peripherals such as RGB sensors, ADCs or a full-HD display, you will need something else to interface with them. This is more the area of an FPGA: it is typically being used as glue logic between various high-speed interfaces. All I/O pins of an FPGA are 100% flexible: it's up to you to program what you want to do with them, which is done in VHDL.
As an example, you could connect the high-bandwidth output of an image sensor kit to the headers of the SmartScope, read out a 5-megapixel image at 48 MB/s, and send the result over the USB link. Because the USB link is slower, you might want to add in a small FIFO buffer, which can be done inside the FPGA, or you could use the much larger RAM inside the SmartScope. Congratulations, you have just made a 5-MP webcam!

Benefits of using the SmartScope open-source platform

Even though numerous FPGA development kits exist, the SmartScope open-source platform offers a couple of benefits:

- You start off with a full USB chain providing a register bank in your FPGA, which you can control from an app on your phone. Your custom VHDL modules have full access to these registers and you can control them from any program on your PC/tablet.

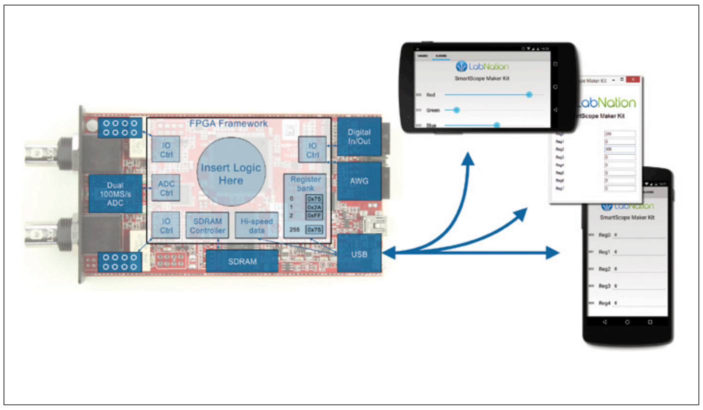


Figure 1. Open-source architecture of the SmartScope FPGA platform.

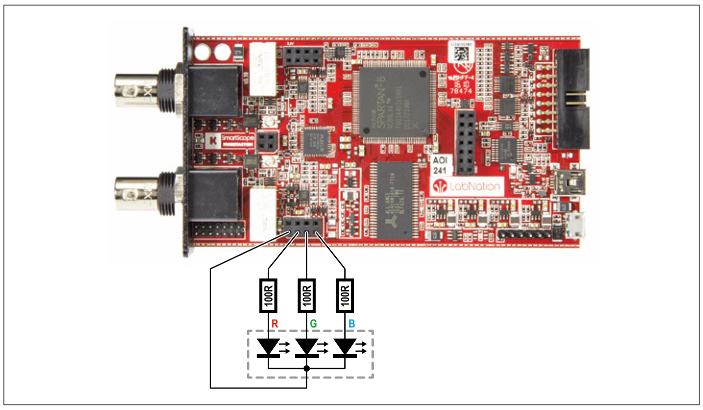


Figure 2. Connection of the RGB LED to the AUX connector of the SmartScope.

- This USB bridge can also be used to upload your VHDL project to your FPGA, so you don't have to connect any additional programmer cables.
- It comes with the SmartScope Maker Kit app running on Win/Lin/OSX/Android/iOS, which allows you to control these registers without writing a single line of code.
- Power your circuit through the SmartScope from your phone/tablet!
- It comes with drivers and a *full-blown and proven software stack* which runs on virtually any OS to kickstart the coding of your own app.
- Simply create your VHDL modules, and control them from C#, MatLab, LabView and more.
- Your VHDL module has direct access to the other hardware inside the SmartScope, such as a powerful dual ADC stage, RAM and USB connectivity. VHDL controllers for these parts are already present in the framework, so you can skip writing time-consuming chip interfaces.
- Completely free if you already own a SmartScope oscilloscope, with no danger of overriding the original firmware of the SmartScope. Open up the SmartScope if you want, but you can also keep your SmartScope closed since you can flash the FPGA over the USB link as well.

Goal of this article

The focus of this article is on getting you up and running with the SmartScope open-source platform, and therefore does not intend to be a tutorial on FPGAs or VHDL. However, in order to keep the required FPGA knowledge to the absolute minimum, this article will aim at the simplest of peripherals to control: an RGB LED (see connection diagram in **Figure 2**).
More advanced examples which make use of other components on board of the SmartScope can be found at the special Maker Kit page on the LabNation website [1].

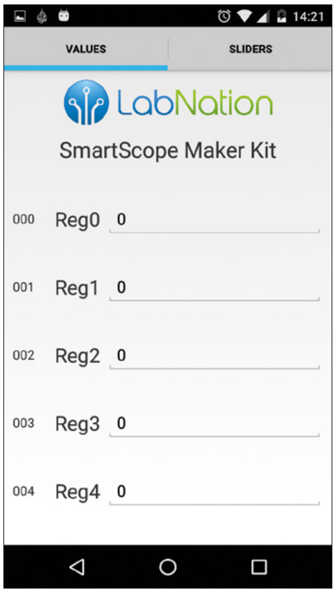
Open-source architecture overview

At the heart of the SmartScope platform (**Figure 1**) is a Xilinx Spartan 6 FPGA. By uploading your own project you can read

Dedicated SmartScope Maker Kit

In order to protect the FPGA inside the regular SmartScope, its AUX port provides pins which are either input (Logic analyzer ports) or output (Digital generator ports).
However, some use-cases such as an I²C master require bidirectional access. Elektor now offers the **SmartScope Maker Kit**, which contains a **special version of the SmartScope**. In this version all internal headers have been assembled, exposing more FPGA pins which provide both read and write access. These headers are matched to a 0.1" grid, allowing you to align your breadboards with ease. Another header provides access to both analog inputs, so the SmartScope can digitize any voltages from your breadboard without the need of using patch wires.
Furthermore, the Maker Kit comes with a **JTAG programmer**, allowing you to hook straight into the FPGA. The JTAG programmer can be used to flash the FPGA and to use Xilinx' ChipScope tool. With ChipScope you can look in real-time at the behavior of all the signals inside the FPGA, providing you the functionality of a logic analyzer built into your FPGA. This visualization offers a debugging capability which is mandatory for more complicated designs. The JTAG programmer can also be used on other Xilinx chips and dev boards.
Additionally, the Maker Kit comes with a **PICKIT3 programmer** for the USB controller, allowing you to experiment with the on-board USB controller as well.

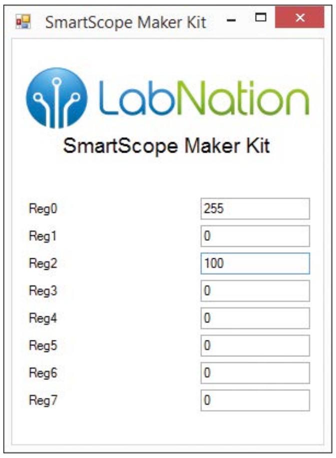
The SmartScope Maker Kit contains a special version of the SmartScope with preassembled headers



SmartScope Maker Kit

000	Reg0	0
001	Reg1	0
002	Reg2	0
003	Reg3	0
004	Reg4	0

Figure 3. The SmartScope Maker Kit app gives you direct control to the registers in your FPGA. This is the app for Android.



SmartScope Maker Kit

Reg0	255
Reg1	0
Reg2	100
Reg3	0
Reg4	0
Reg5	0
Reg6	0
Reg7	0

Figure 4. The registers in the app for Windows.

from or write to the I/O headers, control the SmartScope's logic analyzer port and digital outputs, generate a signal on its AWG or make use of its dual-channel ADC and RAM. All of these components have been made readily available in the top level VHDL file, inviting you to add your module.
On the other side is the USB link, through which you can upload your FPGA configuration and stream data to a PC/tablet. Moreover, notice the register bank which exposes a set of 8-bit registers to your VHDL module. The value of these registers can be directly controlled from your PC/tablet. In order to do so, you don't have to create a single line of code: The platform comes with the pre-compiled SmartScope Maker Kit app (**Figures 3 and 4**), running on PC/tablet/phone, which allows you to read and control the values of these registers in your FPGA. In case you want to give the registers in the Maker Kit app a more meaningful name, simply change the human-readable YAML file and your register's names will be shown the next time you start the Maker Kit app.

Preparation

Install Xilinx' ISE

If you want to compile your VHDL files targeting the Spartan 6 inside the SmartScope, you will need to install Xilinx' free ISE WebPack. Simply google for it or use the link at [1]. Make sure you download the 'Full Installer for Linux/Windows'. After the rather lengthy download, extract everything into a temporary directory (you might need a tool like 7zip [2] for this) and install ISE by running the *xsetup.exe* file (on Windows). During the setup process, the only change you need to make is to select the free 'ISE WebPack' edition, as this will allow you to generate a free license later on.

After the installation you need to get a free license. To do so, in the License Manager window select *Get Free WebPack license*, after which you'll be forwarded to the Xilinx website. In the *Certificate Based Licenses* list, select *ISE WebPack License* and hit the *Generate* button. Find the Xilinx.lic file in your mailbox and save it to your disk. Finally, back in the License Manager open the *Manage Licenses* tab, hit the *Load License* button and point to the license file you just downloaded.

Download open-source files

While ISE is installing, you can already browse to the SmartScope open-source files at the LabNation github pages [3]. Click the *Clone or download* button to get the files on your PC, either by cloning the git repository or by downloading everything as a zip file.

Examining the VHDL framework files

Once downloaded, head to the hld folder which contains all the VHDL source files. The only file of interest is the SmartScopeMakerKit.vhd file, which is the toplevel file where you can fit in your module. Before doing so, let's examine its contents from top to bottom. You will notice the following parts:

Port declarations

You'll find a list of logical names for all pins of the FPGA on the SmartScope, together with a brief description of their purpose. Here's a quick rundown of the different ports, for deeper information you can rely on the schematics of the SmartScope posted at [3].

- General section: incoming oscillator clock and I^2C control pins from the USB controller
- RAM port: signals interfacing the RAM IC
- PIC port: data channel and 'ready signal' for interfacing to the USB controller
- ADC port: signals interfacing the dual-channel ADC
- Analog channel A: all signals for configuring Channel A (division/gain/coupling)
- Analog channel B: all signals for configuring Channel B (division/gain/coupling)
- AWG/LogicAnalyzer/DigitalOut/Trigger
- Misc: additional I/O, such as the internal headers and both signals on the micro-USB port

Signal definitions

Some signals you might find useful to use in your own VHDL module, such as clocks, the register bank and data interface to the USB controller. Notice the register bank (the `sig_regs_user` variable), which contains 256 bytes you can use in your module.

PLL

Presenting 4 clocks of different frequencies (12 MHz, 24 MHz, 48 MHz, 100 MHz) which you can use in your VHDL module.

Reset generation

Provides a short pulse at startup, essential to reset your logic to a known state.

I^{2}C slave

Communication between the USB controller and the register bank on the FPGA is achieved using a standard I^{2}C interface. This means that your FPGA essentially becomes a I^{2}C device, increasing the portability of your code to whichever platform you might use in a next phase.

I^{2}C communication from the PC/tablet over USB towards the FPGA is handled by this component. It allows the USB controller to read from or write to your register bank (`sig_regs_user`).

Control for ROM, RAM, analog channels

In this simple example, all these signals have been tied to fixed values.

ADC Controller

The ADC requires some configuration before it will function as expected. This module interfaces the USB I^{2}C controller to the ADC, so the ADC can be configured from the PC/tablet. See the 'ADC Controller' example on the Maker Kit page [1].

USB PIC Controller

The last module provides a high-bandwidth data path from your FPGA to your PC/tablet. See the 'USB transfer' example on the Maker Kit page [1].

Installing the Maker Kit app

The Maker Kit app comes precompiled for all platforms, meaning it is installable without problems. For Android devices you can download the app from Google Play store, while for other platforms you can download the Maker Kit app from [1]. The same page also includes links to all source files of the Maker Kit app for the various platforms.

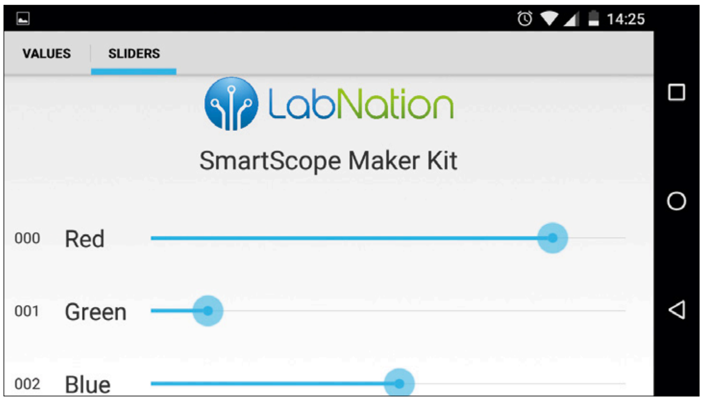


Figure 5. Controlling the three colors of the RGB LED in the Maker Kit app.

Upload your first FPGA project to the SmartScope platform

Build the FPGA firmware

In this section you'll open the example ISE project and build a FPGA firmware (aka bitstream) out of it.

Note: On Windows 8 and onwards ISE contains a bug resulting in a crash whenever you want to browse the file system. To solve this, right-click the desktop ISE icon, select Properties and change settings64.bat to settings32.bat and \ISE\bin\nt64\ise.exe to \ISE\bin\nt\ise.exe

From the FPGA repository directory, open *SmartScopeMakerKit.xise* with ISE.

Hit *Process -> Implement Top Level Module*. Since this is the first time you compile this project, you might need to confirm the generation of the .xco cores.

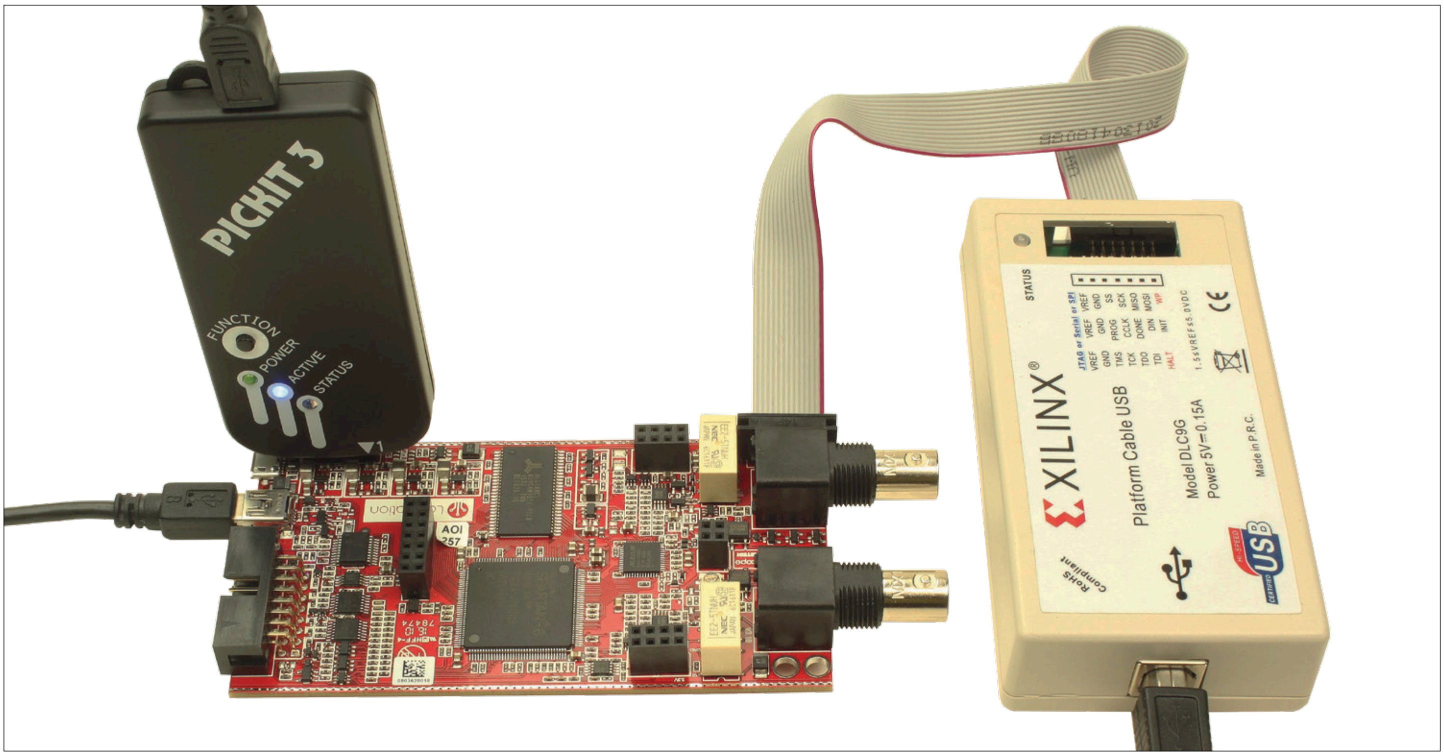


Figure 6. The JTAG programmer and the PICKIT3 programmer connected to SmartScope board.

Add in your module

If you’ve been following until this point, you now have a project which compiles just fine. However, it will not do that much. To solve that, we’ll add in a small module which generates three PWM signals for controlling the color intensities of our RGB LED. We’ll make a counter which keeps incrementing from 0 to 255, and compare it to 3-byte values of our register bank to generate the PWM signals. This will allow us to change the duty cycle of each PWM at runtime from our PC/tablet.

Start by opening the *SmartScopeMakerKit.vhd* file by double-clicking on it at the top-left and adding the following line to your Signal definition block, which creates an 8-bit value:

```
signal sig_counter   : unsigned(7 downto 0);
```

Next, since we’ll be assigning to the `out_digital` port (the digital outputs of the SmartScope’s AUX connector), find and remove/comment the following line:

```
out_digital        <= (others => ‘0’);
```

Finally, add the process from **Listing 1** to your architecture implementation (eg: before the very last ‘end’ of the document). This module will increment the `sig_counter` value at each rising edge of `sig_clk_input`, a 100 MHz clock generated by the PLL. Since it has 8 bits available, after reaching 255 it will automatically overflow to 0. Simultaneously, its value is being compared to three registers inside your register bank, resulting in a low or high output for three digital outputs of the SmartScope’s AUX connector.

Listing 1.

```
process(sig_clk_input, sig_reset, sig_regs_user)
begin
 if (sig_reset = ‘1’) then
   sig_counter <= (others => ‘0’);
   out_digital <= “0000”;
 elsif rising_edge(sig_clk_input) then
   sig_counter <= sig_counter + 1;

   if (sig_counter > sig_regs_user(0)) then
     out_digital(0) <= ‘0’;
   else
     out_digital(0) <= ‘1’;
   end if;
   if (sig_counter > sig_regs_user(1)) then
     out_digital(1) <= ‘0’;
   else

     out_digital(1) <= ‘1’;
   end if;
   if (sig_counter > sig_regs_user(2)) then
     out_digital(2) <= ‘0’;
   else
     out_digital(2) <= ‘1’;
   end if;
 end if;
end process;
```

Upload and test

With the code ready, hit *Process → Implement Top Module* to compile the entire project. Finally, at the bottom-left of your screen find the *Generate Programming File* command and click it to generate the actual bitstream which needs to be flashed into the FPGA.

The latter can be done under full control using the standard Xilinx programs and the JTAG programmer included in the SmartScope Maker Kit. In order to do so, simply connect the JTAG programmer to the SmartScope, and hit the *Configure Target Device* command at the bottom-left of ISE.

Alternatively, the bitstream can also be uploaded through the Maker Kit app. Browse to the location of your ISE project and find the *smartscopemakerkit.bin* file (make sure you have the .bin file and not the .bit file!). Copy it to the working directory of your Maker Kit app, rename it as *SmartScopeCustom.bin*, run the Maker Kit app and as soon as a SmartScope is detected it will be flashed with your customized firmware!

Note: for PCs the working directory refers to the directory where the executable resides. On Android, this is the directory /sdcard/LabNation/MakerSpecial/.

Customize the Maker Kit app register names

By default, the Maker Kit app shows a list of 255 registers, inspiringly named Reg0-255. You can change this easily by adjusting the contents of the *register_map.yml* file in the working directory of the Maker Kit app. For our example, let’s change the contents of that file accordingly:

```
- I2CAddress: 22
  Name: userBank
  Registers:
  - Red
  - Green
  - Blue
```

As a result, when you restart the Maker Kit app, you’ll see only 3 registers with their correct Red, Blue and Green labels.

Summary

Congratulations! You just created a custom VHDL module with very basic functionality, but completely controllable from virtually any USB PC, tablet or even phone!

As a next step, you might be interested in examples which include more complicated VHDL functionality. At [1] you can find links to the following examples:

- ADC to LED
- USB data link

Happy programming with your SmartScope FPGA development system!

(160166)

Web Links

[1] www.lab-nation.com/makerkit

[2] www.7-zip.org/

[3] https://github.com/labnation/fpga

[4] http://wiki.lab-nation.com/index.php/Sources

Tiny Radio

www.elektormagazine.com/labs/tiny-radio-150587

CORRECTION. The Arduino software developed for this project is designed for operation using the ATmega328's internal 8 MHz clock and not for operation with an external 16 MHz crystal. The schematic shows no external crystal.

In order for the software to function correctly IC1 must first be loaded with an 8 MHz bootloader. Check out www.arduino.cc/en/Tutorial/ArduinoToBreadboard and scroll down to 'Minimal Circuit (Eliminating the External Clock)' and just follow the instructions.

We also have a compiled hex code file which can be programmed directly to IC1 without the need for a bootloader.

Err-lectronics

Corrections, Updates and Feedback to published articles

A GPS Synchronized Clock with Seconds Display

Elektor 5/2016, p. 56 (150189)

UPDATE. Some readers living in areas with marginal reception have asked how they can increase the received GPS signal strength. We haven't tested this suggested modification so consider it just food for thought. It should be possible to mount the GPS receiver remotely from the rest of the clock and link its output data to the clock with a length of cable.

To make the mod you will need to unsolder the GPS module from the main board and mount it on a small piece of perf board. Connect all the ground pins and fit through-hole versions of R10, R20 and C2 to the correct pins and fit an additional 10 µF capacitor in parallel with C2 to act as an energy reservoir.

Once you have made all the connections to the module as shown in the schematic connect the power, ground, Rx and Tx data lines between the module and clock using a standard CAT5 network cable or a length of screened telephone cable (this can be up to 10 m in length). It's advisable to clip ferrite filters around the outside of each end of the cable (these are normally seen fitted to USB and VGA cables) to help suppress any RF interference. The data signals between the module and main board should be carried using twisted-pair cables. The cable ground shield is connected to the GND of the clock board. To neaten it all up you could connect the cable using a round plug and socket connector fitted to the clock case.

The GPS sensor can now be fitted into a weatherproof enclosure and mounted outside in a position with a good view of the sky such as around the edge of a roof skylight.

N.B. This suggested solution hasn't been tested in our lab, it may produce unexpected results. If you do try this out we would love to hear how you got on.

Ralf Schmiedel
Elektor Tech Support

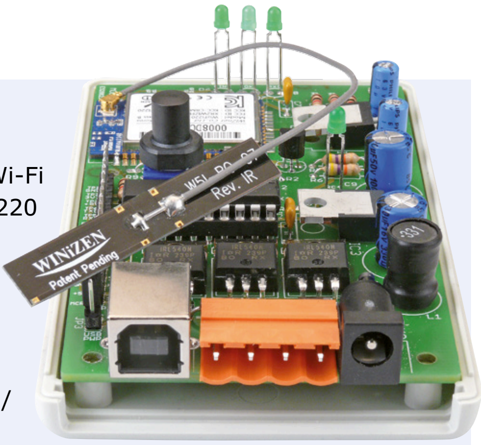
Wi-Fi Controller Board

Elektor June 2013, p. 14 (120718)

FEEDBACK. The Wi-Fi controller board will disconnect itself from the Wi-Fi network after a short period of time. The solution is to disable the WizFi220 module's power-save mode. The AT-command required to turn off power-save is 'AT+WRXPS=0'.

Hans

The original comment may be found at: www.elektormagazine.com/labs/elektor-wifi-rgb-ledstrip-120718

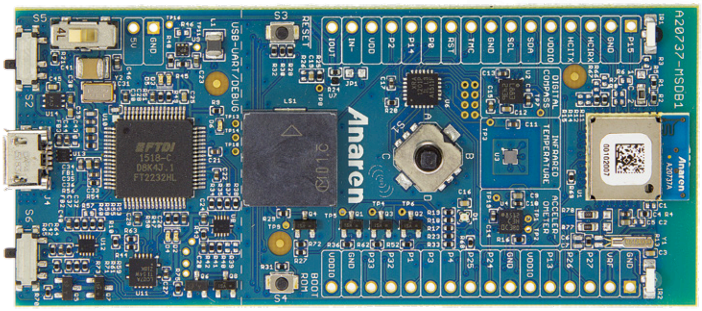


Anaren Bluetooth Smart Development Kit

By Jan Buiting

Anaren
BROADCOM
AIR for WICED
Bluetooth® Smart Development Kit
Featuring the ATMOSPHERE online development tool
Congratulations!
You just opened up a world of possibilities in mobile control of embedded applications.
ATMOSPHERE
Disclaimer and Regulatory Information
Getting Started with the Bluetooth® Smart Development Kit
A20737A-MSDK1

If you find it hard or costly to get your hands on one of those fancy development kits on offer these days from just about any company active in the embedded and microcontroller industry, get a train ticket or hitch a ride to the Nuremberg Embedded World show. At the show, Negative Logic applies, meaning you will have a hard time to NOT get development kits. Also, you have to assume a Logic Low state all the time to ward off hoards of under-25 frontdeskies eager (and paid to) push a dev kit or two in your hands as you pass their booth on your way to the nearest toilet or catering facility.



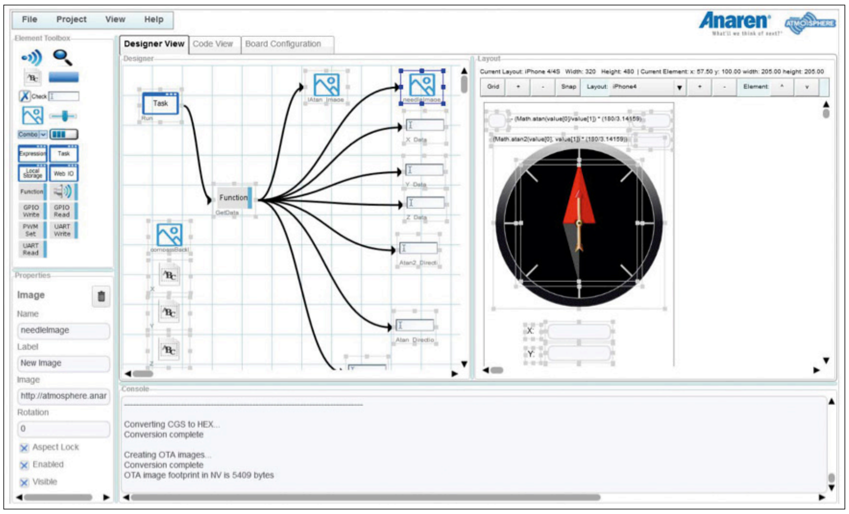
1

Broadcom

To be honest, my interest in a dev kit from Anaren was the mention of "Broadcom" plus a fantastic demo given by Mark Bowyer. Broadcom is the company that got tons of credit from the Programmer Generation for "sponsoring" a million or so fat controller ICs for landing on Eben Upton's Raspberry Pi boards. Broadcom — the same company hitherto associated with cheap plastic Internet routers, boring server and IT hardware, and products for mass production, i.e. OEMs and stuff. Thumbs up to Broadcom, and welcome to the maker world!

Small box, big contents

Anaren's dev kit is based on Broadcom's BCM20732 WICED Smart Bluetooth Low Energy (BLE) module. The BCM20732 is a SoC'ed ARM Cortex M3 so here's a wicked opportunity for me, an 8-bit guy, to learn some ARM programming.



Current Layout: iPhone 4/4S Width: 320 Height: 480 | Current Element: x: 57.50 y: 100.00 width: 205.00 height: 205.00

Layout: iPhone4

(Math.atan(value[0]/value[1]) * (180/3.14159)

(Math.atan2(value[0], value[1]) * (180/3.14159))

Properties

Image

Name: needleImage

Label: New Image

Image: http://atmosphere.anar

Rotation: 0

☒ Aspect Lock

☒ Enabled

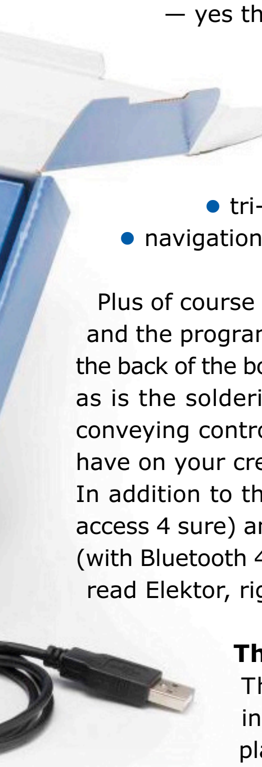
☒ Visible

```
Converting CGS to HEX...
Conversion complete

Creating OTA images...
Conversion complete
OTA image footprint in NV is 5409 bytes
```

2

But it gets better still: WICED SMART enables a wireless cable replacement. Through Bluetooth! My favorite short-range radio system.

So there I had a kit on my desk with a promise of a steep learning curve for ARM and Bluetooth LE in one, all for under $40 from Digikey — and less from Elektor soon (watch our .com site). In terms of hardware, the small board in the kit (**1**) has a heavy and happy accent on sensors and actuators — yes the real world, so how about:



- accelerometer;
- magnetometer;
- infrared temperature sensor;
- buzzer;
- tri-color (RGB) LED;
- navigation switch (smurf's joystick).

Plus of course supporting components like the BLE module, and the programming hardware. Filling the battery holder at the back of the board with a coin cell is optional (data retention), as is the soldering of two 16-pin single-row pinheaders for conveying control signals to any external circuitry you might have on your creative mind.

In addition to the kit you need a PC (with USB and Internet access 4 sure) and an iPhone or Android smartphone or tablet (with Bluetooth 4 sure). Optionally, a solder iron, but then you read Elektor, right?

The right atmosphere

The getting-started process was a breeze including registration to Anaren's 'Atmosphere' platform, where communal spirit is encouraged through a forum with backup from a remarkably good Wiki and FAQ. I followed the demo to put the LED on the board under control of three color sliders on my Samsung smartphone, through Bluetooth. That's right, the kit enables you to build an app complete with Bluetooth comms as you drag and drop elements onto a worksheet shown on your PC screen. The Atmosphere tool talks to the A20737 module and helped me by flagging my most obvious mistakes.

Once you feel confident about your "drawing" on the PC screen (**2**), you can safely proceed to programming the Anaren module with the "firmware" (**3**). You use your PC to sketch, debug and preview your "app", and download the associated firmware to the dev board via an FTDI USB bridge on the board. A bit of magic happens with the smartphone, which receives the code for "your app" wirelessly. You can use the app straight away. Missus or teenage daughters don't like the colors? Go back to Atmosphere, adapt, maybe add a buzzer sound, save, upload, check response again.

Very powerful

Anaren in their kit documents print a strong and sincerely expressed encouragement to start developing what you want and believe is feasible, using the kit "as a mere get-you-going tool", which I found overmodest on their part. I found the kit top class because as a newbie I am not ready for the nooks and crannies of BLE, ARM, or app building, and all the hassle I feared from reading about these new technologies is happily absent. With the kit, BLE is transparent, almost invisible, in terms of programming (a far cry from Elektor's own BL600), while ARM programming is "limited" to dragging elements from the library onto the worksheet and seeing what happens (a far cry from Elektor's ARM programming course and most hardware). That's fully respecting anyone's ability to delve into these subjects, explore, understand and appreciate as he or she wishes.

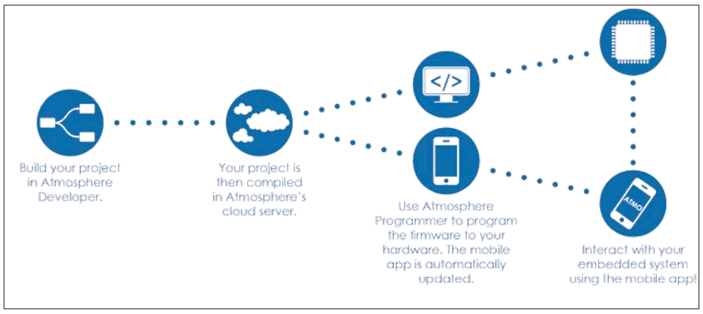
Wider atmosphere please

As a minor criticism I would suggest that the library of control elements in Atmosphere for your app to use can do with expansion, and that the worksheet is not optimal in terms of graphics (**4**). Also, I have a slight concern about the use in the future of other ARM processors, although for now I am still heavily challenged by the one lurking in the Broadcom chip.

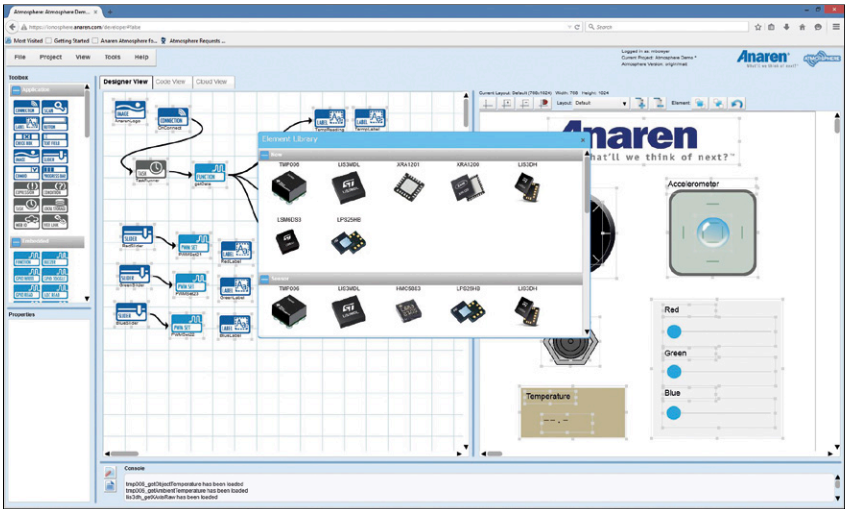
Conclusion: Just under $40 spent at Digikey, or less at the Elektor Store brings extremely flexible ARM and Bluetooth Low Energy hardware, and for "softerware" you get embedded programming, app building skills, and ease of mind. That kit is WICKED.

(160195)

Anaren Bluetooth Smart Development Kit demo video:
https://youtu.be/sjDn-6S0NKw


Build your project in Atmosphere Developer.
Your project is then compiled in Atmosphere's cloud server.
</>
Use Atmosphere Programmer to program the firmware to your hardware. The mobile app is automatically updated.
Interact with your embedded system using the mobile app!

3



tmp006_getObjectTemperature has been loaded
tmp006_getAmbientTemperature has been loaded
lis3dh_getXAxisRaw has been loaded

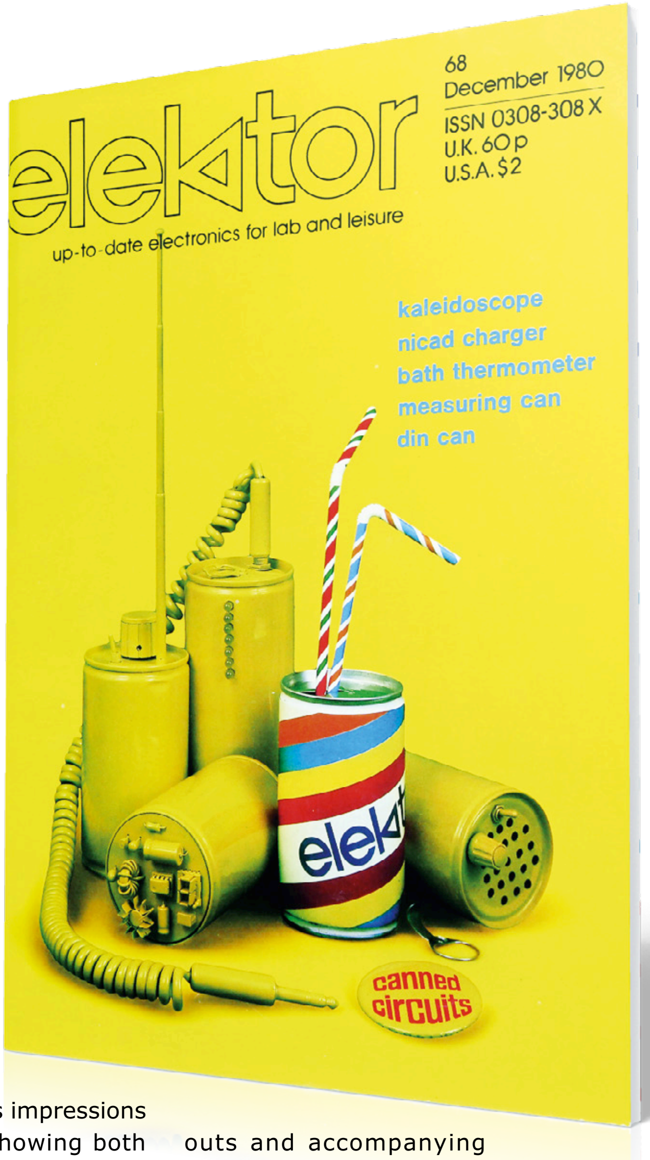
4

Elektor 35 Years Ago

December 1980: Canned Circuits

By Jan Buiting

Back in 1980 Elektor challenged its readership to come up with circuits and projects housed in a beer, soft drink or sardine can. Conceivably because of the mention of beer, the result was an impressive number of entries, some exhibiting skill and ingenuity, others plain daftness. It was Cannery Row in an electronics magazine.



Among the other qualifications used by the anonymous Jury in their December 1980 little foreword were "no good", "very good", "hilarious", "made in Germany", and "impossible". It's hardly surprising with the competition challenge formulated as "send in your impression of what electronics can do in a can" and not specifying the can size. Apparently no entries were received showing the use of a 10-gallon oil drum, so the Jury had a lucky escape there.

The December 1980 edition of Elektor was mainly composed of 23 of the best entries — in fact the prize winners. See the **inset** for the full list of article titles, and please do reflect for a moment how much of the fun in electronics has been lost since the great wave of commercialization and CEO-ing that swept the hobby.

Of the final selection back in 1980, seven got published complete with a *printed wiring board* (PWB). Reportedly the lucky winners received a *Casio LCD credit card* (-sized, *Ed.*) *calculator* that can also be used to compose and play back one's own tunes. I am lead to believe the calculator is featured at Vintage Calculators [1]. The remaining 16 prize winning entries the Jury said "were again original ideas but either not quite so practical or needed a little extra in the way of construction."

These 'honorary mentions' received a stylish all-plastic AM/FM radio Radio (*sic*). One reader sent in 'Locomotive' and I cannot resist showing it to you here.

Among the entries classified as undoable were a Budgie Entertainer and a complete oscilloscope built in a number of cans. With hindsight, surprisingly few readers sent in radio projects exploring the excellent RF shielding capacity of a beer can, or was it the aluminum you can't solder?

Highly improbable

Excepting the December 1980 magazine cover, no photos appear of the winning projects with their intended appearance, i.e. housed in a can. Instead, there are wonderful artist's impressions of the projects in cans, showing both Laurent Martin's craftsmanship with pen, ink and possibly some cigar smoke.

Sadly the drawings also suggest that none of this stuff was ever tested to the extent of someone in the lab actually building the circuit, canning it, and offering it up for photography. Although the drawings add much to the speculative character of the projects, you might counter that some Elektor-style PCB layouts and accompanying parts lists suggest that it CAN be done.

When looking at some of the winning entries on these Retronics pages, please bear in mind that the images, schematics and drawings were photographed from printed pages out of my personal collection of Elektor magazines. Sadly no original artwork has remained of the Canned Circuits competition. ⏮

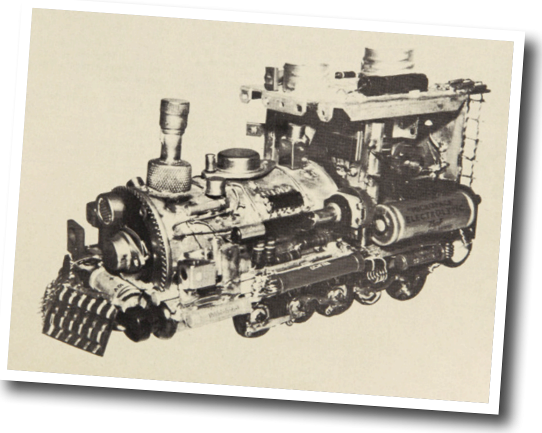
(150542)

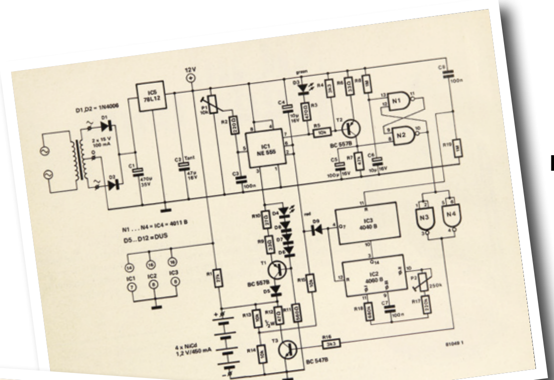
ESTD 2004
www.elektor.tv

Retronics is a monthly section covering vintage electronics including legendary Elektor designs.
Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

Locomotive

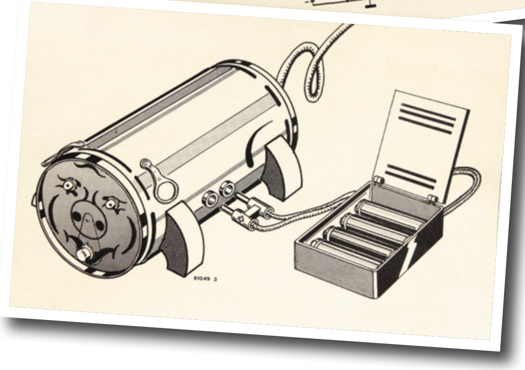
This entry from H. Thiele is utterly remarkable not just for not being a circuit but also as unexpectedly triggering another competition: “which parts are used in the construction of the locomotive?” That should have been hard to answer even in 1980 as the print quality of monochrome photos in the magazine at the time was, ermm, incomparable to today, but still way better than Elektor’s rivals EPE and EW. The Can? Oh yes, that’s the boiler. I can see two Toko coils, two TO-3 transistors, one TO-5, one U/S transducer, two FM band coils (?), one ESR-suspect electrolytic, and… sorry, have to stop here.





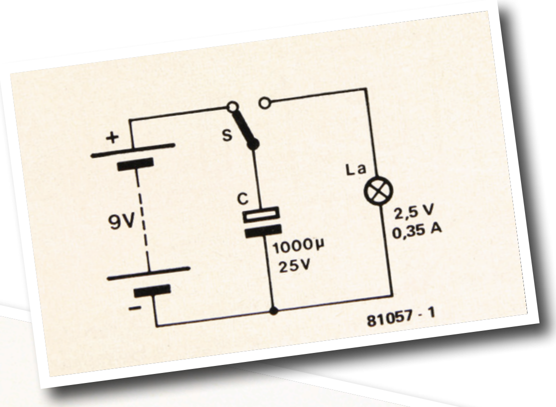
NiCad Piggy Bank

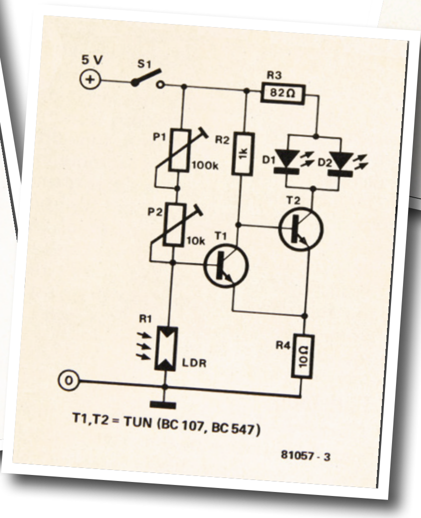
Notice that spelling in the title, ‘NiCd’ has dominated since then, and is also seen in the schematic. Which has all the favorites of the late 1970s scene: the LED, CMOS ICs 4011, 4040 and 4060, the 7812, and of course the BC55x and the NE555. These and other parts from the buy-by-the-dozen class spawned hundreds of circuits in Elektor and even today are seen in project proposals we get by airmail letter from places like Kathmandu and South Novosibirsk. In the drawing of mother pig nursing her four NiCad piglets, the curly tail going off the page edge suggests the AC line power cord, which is careless if not extremely dangerous in terms of electrical safety. Also notice two (?) pull tabs turned into ears. The current source with its chain of LEDs and DUSs we would judge today as doubtful in respect of thermal response.

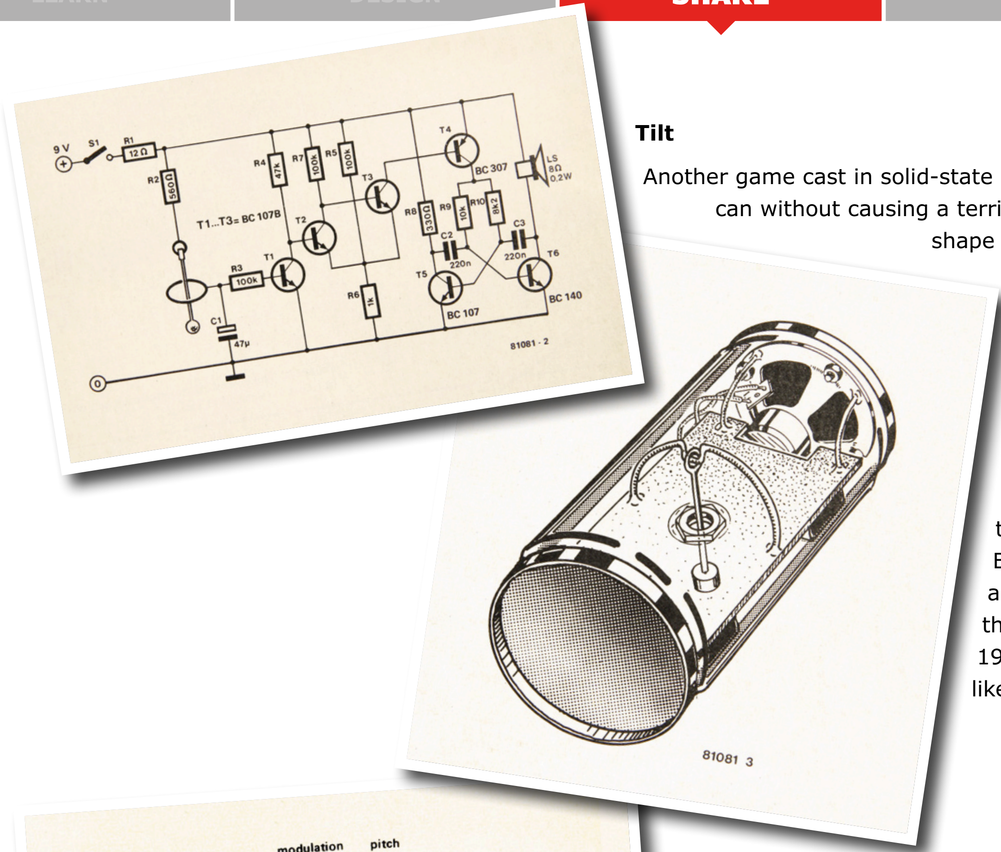


A Flash in the Can

In this project, only the target is housed in a can. The shooting device is a light emitting pistol which would get you arrested today in some areas, and the publisher grilled by lawyers. It works by hard-discharging a 1000-µF electrolytic capacitor charged to 9 volts across a 2.5-volts flashlight bulb. The light ‘shot’ fired by the marksman is supposed to hit an LDR fitted in the canned target, which responds by lighting two LEDs. As with the NiCad Piggy Bank, curiously there are two pull tabs for ears. Whence the face of the target and its horns (?) is uncertain. If this is actually a one-shot we do not know either. Don’t try this at home.

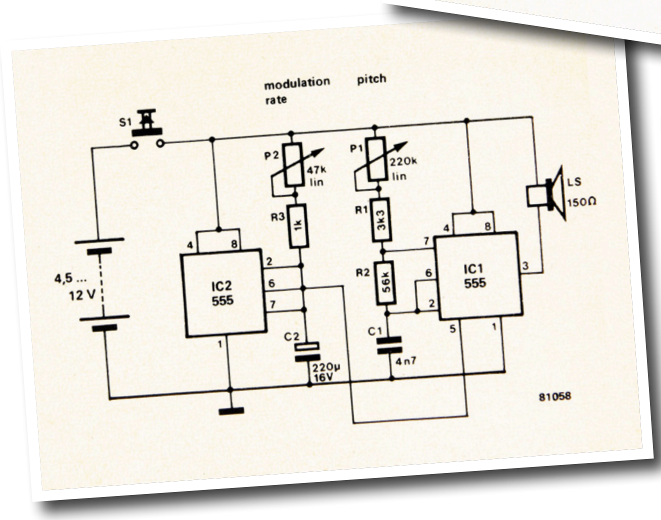






Tilt

Another game cast in solid-state electronics. The challenge was to roll a can without causing a terrifying noise. We read that the cylindrical shape of the can has been used to its fullest advantage, the most important component being a pendulum which reacts to any movement of the can by swinging and thereby effectively closing a switch. The switch triggers a monostable multivibrator T2-T3 which in turn activates an astable multivibrator (T5-T6). Interestingly the transistors used are the metal can BC107s, which indicates the relative age of P. Groger's component stock as the plastic BC547 was already around in 1980. Fair enough, even older trannies like the AC187K appear in other circuits.



DinCan

It's really doubtful how this one from one L. van Ginderen could have made it to the winners. So frugal, it must come from the Netherlands. Apparently not even Laurent could be bothered to make a can-ish drawing for it. Or someone from labs Dutchifying the BOM further by using one 556 instead of two 555s. The squeezed-in-a-corner article says that the two timer ICs produce a Kojak-style siren sound which by employing the resonance of the can, "makes your ears tingle for quite a while". A suggested place to use is on a skate board, spelled *skateboard* today.

The winning entries

Image: Vintagecalculators.com

Boldface indicates project with PCB design. Bath Thermometer**,** Xylophone**; Can Shy; Midnight Raid Detector;** NiCad Piggy Bank**; A Flash in the Can; Candid Comments;** Drinks Round Indicator**; Kaleidoscope; Panoramascope; Locomotive; Dice Shaker; Canometer; Tilt; Marimba Can;** Multi-Game Can**; Mobile Can; DinCan;** Genie in the Can**;** Piping Can**; Simple Remote Control; Water Can; Canine Defence; Garden Gnome.**

Web Link: [1] Casio Mini Card LC-78: www.vintagecalculators.com/html/casio_mini_card_lc-78.html

The complete December 1980 edition is available on the Elektor 1980-1989 DVD, get it from our STORE.

Elektor Labs Pipeline

The winter season is nigh — time to plan your end-of-year holiday pursuits. Here are a few suggestions to keep you busy.

Build your own LoRaWAN with gateway & nodes

Did you know it's really easy to set up your own LoRaWAN? All you need is a bunch of commercial modules plugged together and away you go. A gateway is constructed with a Raspberry Pi and a LoRa hat; nodes are built from Arduino boards and serial-port-to-LoRaWAN modules. The gateway connects to The Things Network which in turn connects to other gateways in the world. Join the crowdsourced LoRa network and enjoy IoT services for free.

http://po.st/LoRaWAN-gateway

Build a very long range remote telemetry system using a LoRa repeater

Coal based thermal power plants are among the greatest polluters today. In India, the National Green Tribunal (NGT) is assigned to keep a watchful eye. They not only check on polluters but also take measurements at the source of the pollution and monitor it. Since polluters occur in (very) remote locations telemetry is often needed. This project uses a LoRa repeater to achieve the required range.

http://po.st/LoRa-repeater-telemetry

Build an analog joystick-to-USB interface

The purpose of this project is to provide a USB interface between a vintage (analog) joystick with a game port connector (DA-15, better, but incorrectly, known as DB15) and a PC. The analog joystick uses two variable resistors for X and Y positioning. A constant current through these resistors results in a voltage which a Microchip PIC18F2550 converts into USB data. The hard- and software (written in mikroPascal) can be easily adapted to joysticks of similar design.

http://po.st/joystick-to-USB

Build a solar-powered USB plug

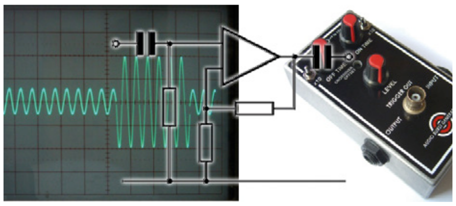
When you want to use your smartphone for an extended stretch of time in an area where electricity is very scarce, you either bring with you enough batteries to get your phone through the trip, or you bring a charger that can harvest energy from the environment. Here is a project for a solar mobile phone charger for top mounting on your backpack or at another convenient location. It works — the designer tried it out in the Gobi desert and survived to tell us about it.

http://po.st/solar-USB-charger

Build an audio burst generator

While examining overload recovery in AC-coupled amplifiers or testing the attack/release behavior of audio compressors, well-defined tone bursts come in really handy. This simple circuit accepts a steady tone from an ordinary audio oscillator at its input, and responds by producing bursts at its output. The duration of the burst as well as their repetition rate are variable from less than a millisecond to several seconds. As a matter of course the amplitude and switching point are adjustable too. ◀

(160184)



http://po.st/audioburst

Compiled by **Robert van der Zwan**

The Electroniad is Nigh

On Friday October 14, an ‘electronics Olympiad’ called Electroniad is held in Mechelen, Belgium. The event is staged to celebrate the 10th anniversary of Caeleste, one of the world players in the design and realization of high-end image sensors. If you happen to read this before October 14 and would like to participate with your own team of four members, contact the Electroniad team at www.electroniad.org and dash off to Mechelen. Winners in one of the disciplines defined by Electroniad are awarded a free *Elektor Magazine* subscription.

Fast Forward Indeed

Jury members from electronica 2016 as well as Elektor are busy deliberating who deserve to be the finalists for the *electronica Fast Forward Start-up Award*. Rest assured this is a tough one! Who will be the winners on November 11 at the electronica show in Munich? Frankly it’s a guess as the last minute entries are still coming in.

Latest mid-position: 60, and counting…

READ ONLY MEMORY

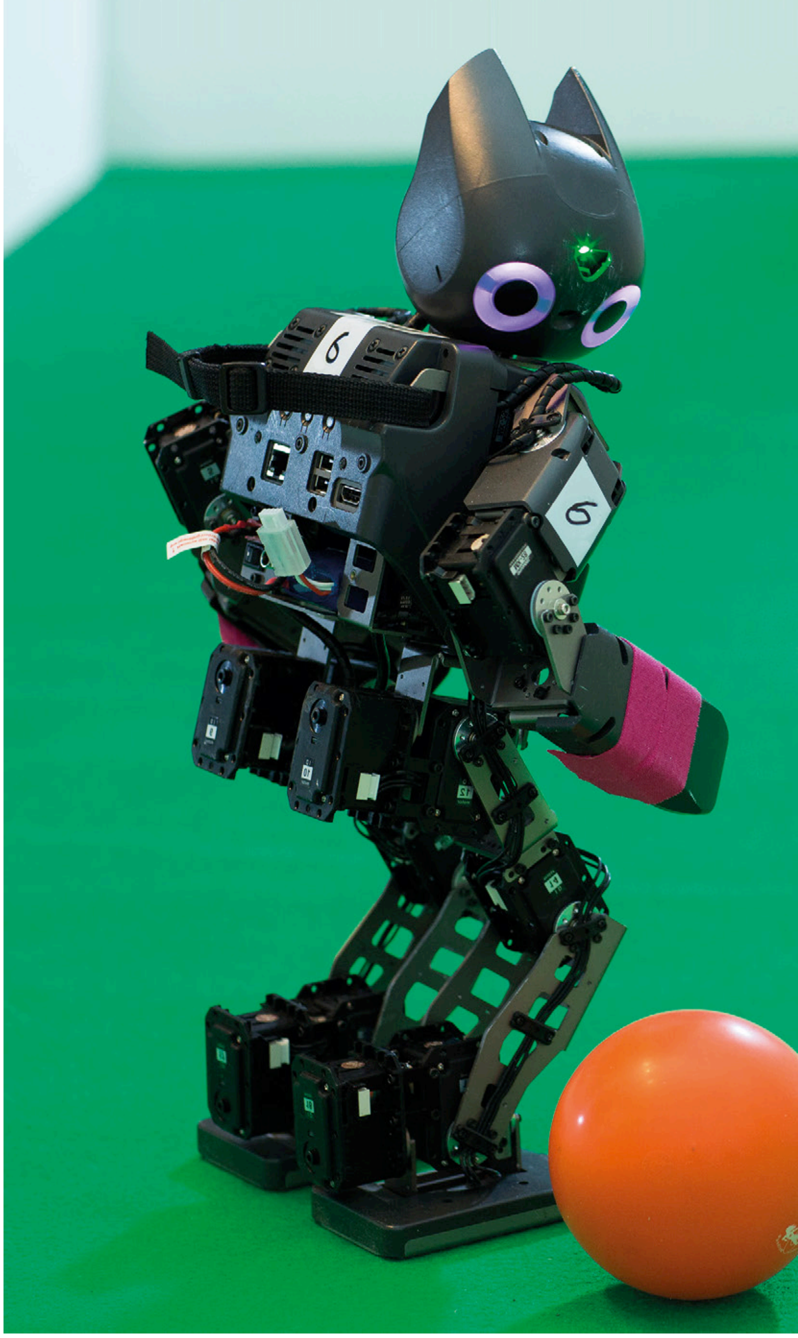
Elektor magazine and its parent publishing company boast a long and rich history. In this space we picture a gem from the past.

GBDSO — a cracker!

Do you recall the Nintendo GameBoy? This game console had been around for 10-odd years when in 2000 one Steve Willis came up with a splendid idea. Why not use the GB as a sampling oscilloscope instead of playing kiddo games? With the circuit of course housed in a slightly adapted GB ROM cartridge casing. The Elektor GBDSO was born. Only the ‘scope probes were conventional. Steve’s GBDSO cartridge described in the October and November 2000 issues of *Elektor Electronics* magazine went sky rocket. It was Elektor’s first SMA-preassembled board and thousands were sold.

RoboCup Soccer: Will They Ever Win?

RoboCup is a worldwide organization aiming to develop a robot soccer team able to beat the human soccer champion in 2050. One of Elektor Holland’s translators, Evelien Snel, is a member of a RoboCup team that participates in the Mid Sized League. What is her take on RoboCup soccer?



PEOPLE NEWS • Conrad Djedjebi is originally from Benin, West Africa. Currently he’s an intern working

Ecole supérieure d’électronique de l’Ouest in Angers, France, he is working temporarily at Elektor to beef

• Udo Bormann is playing a pivotal role in polishing the newsrooms Elektor is providing for its business

force. The newsrooms combine partner specific news with Elektor’s complete online news database … …

Evelien started in 2000 in a team called CFT: Philips Cyber Football Team. Nowadays this team works under the name VDL Robot Sports and it is staffed by techies from VDL ETG and other companies in the Eindhoven, Netherlands, region. Apart from a growing number of young enthusiasts, the team boasts many old hands at mechatronics who learned the basics of electronics in the 1970's from *Elektuur* magazine, now also dubbed *Elektor*.

"2050 may seem far away," says Evelien, "but there is still a lot of work to do. Right now all four leagues concentrate on a part of the problem. The Humanoid League is busy with kicking and standing on real legs — producing lots of endearing footage on YouTube. The Small Sized League is very good at strategic play, but it is too quick and too small to attract a large audience. The Simulation League is the only one that handles 11 players in a team, but it happens in a virtual environment. The Mid Sized League is the only one that can play against real humans. An important match in this league can easily attract an audience of more than 1000 live spectators."

Eventually the knowledge and experience of the leagues will have to be merged to reach the ultimate goal. There is a lot of sharing going on already. The annual World Championships also feature a conference: a complete day of technical lectures about the latest developments. Most of the software and hardware development is Open Source, and there is also informal sharing and communication between teams. According to Evelien, this closely resembles the way of working in the maker scene or at Elektor Labs.

Speaking of Elektor: as a first step to become human enough to win from humans, Evelien and her teammates teach their robots to read *Elektuur*, now *Elektor* magazine (see photo). That's classified though — do not tell this to the other teams.

www.robocup.org
www.youtube.com/watch?v=1h5147KLikU

for Elektor. Coming from the

up Elektor's FPGA board with a new controller

partners, giving them a news outlet with due attractive

...

EXPERT PROFILE

Elektor works closely together with more than 1,000 experts and authors for the publication of books, articles, DVDs, webinars and live events. In each installment of Elektor Word News we put one of them in the limelight.

Name: **Bart Dierickx**

Age: **57**

Education: **Msc Electronics Engineering and PhD at Leuven University, Belgium**

Currently busy with: **CTO of Caeleste, involved in many 'beyond state of the art' image sensor developments; also organizer of the Electroniad 2016**

Who is Bart Dierickx?

I started as a CCD designer at KU Leuven, subsequently pioneering CMOS image sensors at IMEC. In 2006, I founded Caeleste, now a company at the forefront of image sensors. I am intrigued by quantum physics as the basic science for image sensing.

WHAT trigger to start developing electronics?

Since early childhood I was fascinated by airplanes, and later by space journeying. I was part of the generation that grew up in the belief that space journeys were just around the corner. Then I quickly found out that a science called 'electronics' was my key missing skill.

Which project are you most proud of and why?

My team and I made many one-offs. Sometimes I boast of a few chips I designed, now resting on the surface of a planet or a comet. I am also proud of having co-founded several good companies and having trained many excellent engineers.

What do you hope to accomplish within the next five years?

I would like to further extend the critical mass of Caeleste in the broad domain of image sensing, while staying in close contact of course with the technological and conceptual developments invariably ahead of us.

Why did your company decide to organize the Electroniad?

It started as a funny idea to celebrate our 10th anniversary. In fact, we initially called our initiative the 'Electronics Olympiad'. One or our collaborators had the brilliant idea to ask permission of the International Olympic Committee to use the term 'Olympiad'. *Of course* the answer was a solid No. Then we rebranded the name to *Electroniad*. A second good reason was that, with image sensing, we are on the crossroads between *almost all* sub areas in the wide field of electronics: analog, mixed, digital, opto-electronics, PCB, protocols, solid state physics, CMOS processing, you name it. But let's not forget: it must be enjoyable. You should not really train for it.

Suppose you get $500 to buy stuff in the Elektor Store: what's it going to be? Why?

For Caeleste's use: books and tutorial DVDs on electronic design, especially those that go in-depth for us to understand how circuits work.

Hexadoku The Original Elektorized Sudoku

Traditionally, the last pages of Elektor Magazine's SHARE section are reserved for non-engineering activities like reading about old equipment and solving a 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 three 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 three Elektor Book Vouchers worth **$70.00 / £40.00 / €50.00 each**, which should encourage all Elektor readers to participate.

Participate!

Ultimately December 1, 2016, 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 installment 5/2016 (September & October) is: **19A4E**.
The €50 / £40 / $70 book vouchers have been awarded to: Ted Black (USA), A. Huisman (Netherlands), and Per Troelsen (Denemark).

Congratulations everyone!

3	6			A			F			E		2			
D		1		C	0	2		F	9	3				4	
A	F	0				1		C	B	4	6				
	8		C	3					1			5			0
E	C					4		0	2	F	9		A		
			8			0		D	6		E		F	7	9
	7	9			A	C			8	B	1	4	6	D	
		F	D			9	6	4		5	7		0	8	
				2	9			6							F
1					B			7	4	9	D		8	E	
	3		4		E	D	C		5					2	
			9				A					0		B	4
				0	C			2		D		7			
0		▨	▨	▨	▨	▨		9	7				5	A	
	D				6				0		A	F	C		8
		C				5					B		2	0	D

2	8	0	7	3	C	D	1	9	A	4	E	5	F	B	6
B	9	1	A	5	7	2	4	F	3	6	D	8	0	C	E
C	3	5	F	E	6	8	9	B	0	2	7	D	1	4	A
D	4	E	6	A	B	F	0	C	1	5	8	2	9	3	7
3	5	6	0	2	8	1	D	4	C	9	B	E	A	7	F
1	2	F	4	6	9	3	E	5	D	7	A	B	C	8	0
E	A	8	C	7	0	B	5	2	6	3	F	9	D	1	4
7	B	9	D	4	A	C	F	E	8	0	1	3	2	6	5
4	1	3	B	8	2	E	7	0	9	A	C	6	5	F	D
5	D	7	E	9	F	0	6	1	2	8	4	C	3	A	B
9	F	C	8	B	3	4	A	D	5	E	6	0	7	2	1
0	6	A	2	D	1	5	C	7	B	F	3	4	8	E	9
8	C	D	5	F	4	A	2	6	E	1	9	7	B	0	3
6	7	4	1	C	D	9	3	8	F	B	0	A	E	5	2
A	0	2	9	1	E	6	B	3	7	C	5	F	4	D	8
F	E	B	3	0	5	7	8	A	4	D	2	1	6	9	C

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.

DO YOU WANT THE BEST ELECTRONICS DESIGN SOFTWARE?

FEATURES

- Schematic Capture
- PCB Layout
- Gridless Autorouting
- 3D Visualization
- M-CAD Integration
- SPICE Simulation
- MCU Co-simulation
- Built in IDE
- Visual Programming

Labcenter Electronics

www.labcenter.com

Tel: +44 (0)1756 753440