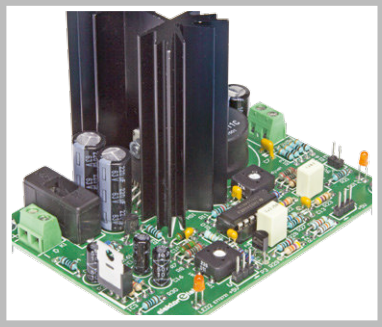
Edition 5/2016 | September & October 2016
www.elektormagazine.com

elektor

LEARN DESIGN SHARE



UniLab Mk. 2

0-30V, 3A compact switch-mode lab supply



Home Automation

10 platforms examined in 5 ways



SmartScope Meets LabVIEW

With a dedicated set of VI blocks

Swiss Pi

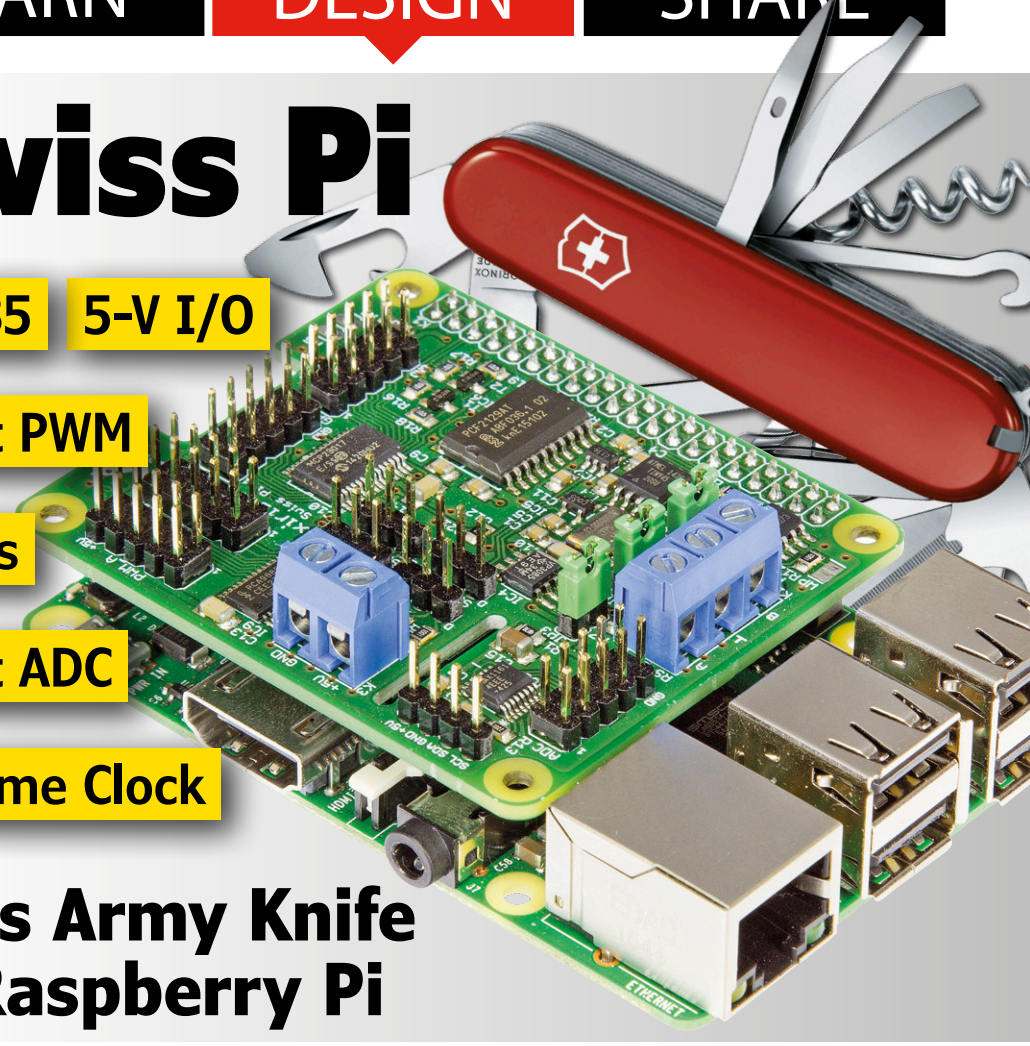
RS-485 5-V I/O

12-bit PWM

Servos

12-bit ADC

Realtime Clock



Swiss Army Knife for Raspberry Pi

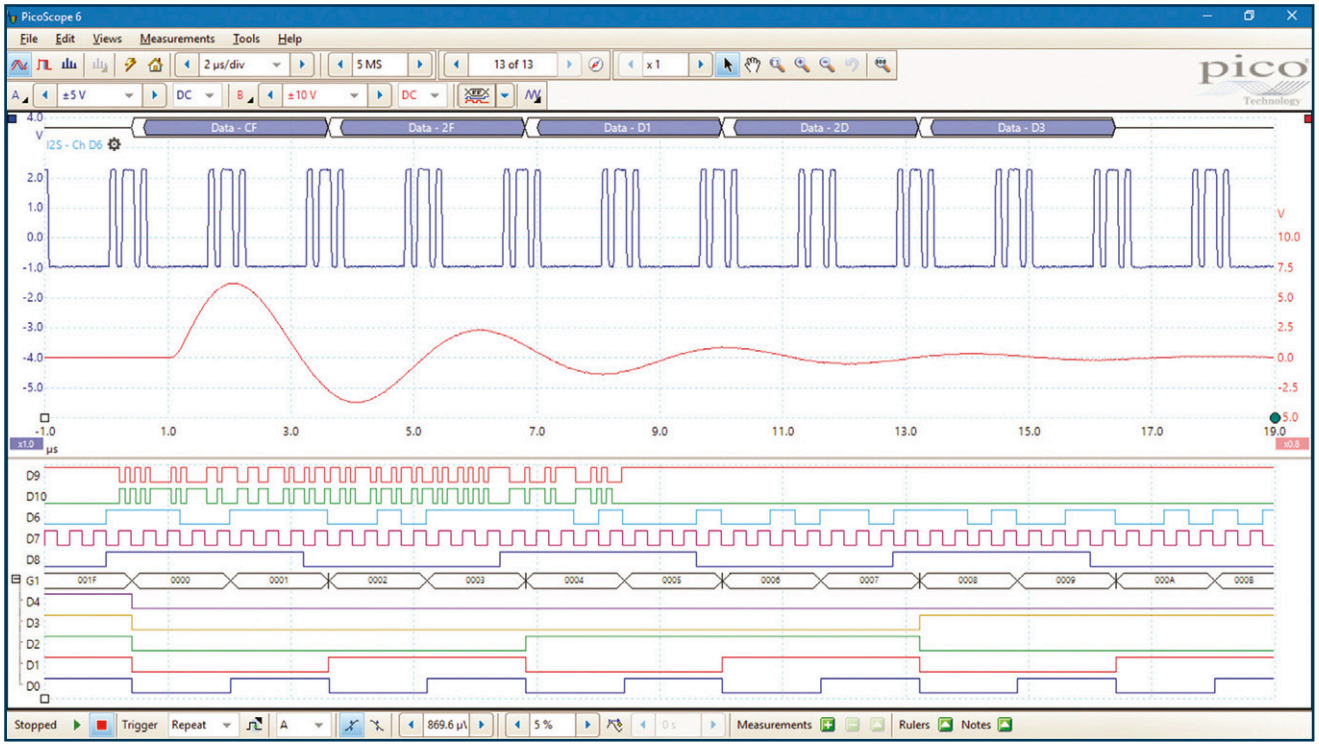
Baroduino • CircuitMaker Tips & Tricks • **Compact and Self-contained WLAN** • **EAGLE Tips & Tricks** • electronica Fast Forward Award 2016 • **Elektor Labs Pipeline** • Elektor SDR Reloaded • **Filtering on the Red Pitaya** • **Fragile LED Filaments** • **Hexadoku** • **Home Automation** • i-Baxandall • **LED Control Tabletized** • LEDitron • **Linked: T-Board 28 and Serial e-BoB** • **A Low-Battery Indicator** • **Magic Eye Indicators** • Master/Slave LED Striplight Dimmer Unit • **Mercury Relays** • **Mobile Communication Instrumentation** • **MOS Clock 5314 (1974)** • Opamp Experimenting Kit for MyDAQ • **Picowatt Transmitter** • **Poor Man's Capacitive Sensing Pad** • Runtime measurements • **Simple Current Sensor Probe** • **SmartScope Meets LabVIEW** • **Swiss Pi** • **THE BIG RED** • Timeshift Radio • **UniLab Mk. 2** • UV Exposure Unit with PIC-based Timer • **& More**

WAVEFORM ENGINEERING ON YOUR DESKTOP

pico® Technology

With bandwidths from 10 MHz to 100 MHz, 8 kS to 128 MS capture memory, 100 MS/s to 1 GS/s sampling rate and a built-in signal generator, there's a PicoScope® 2000 to capture, display and analyze all the waveforms in your world.

2-channel, 4-channel and 2+16 channel mixed-signal (MSO) models have the functionality of an oscilloscope plus a logic analyzer (on MSOs), spectrum analyzer, function generator, arbitrary waveform generator and serial bus analyzer with support for 15 protocols included as standard. All models are connected and powered by USB.





PicoScope 2200A models

from €109

2-channel waveform viewing. 10 or 25 MHz, or 25 MHz MSO.



PicoScope 2200B models

from €319

2-channel waveform analysis. 50, 70 or 100 MHz. Deep memory.



PicoScope 2400B models

from €559

Analyze complex designs. 4 channels. 50, 70 or 100 MHz with deep memory.



PicoScope 2200B MSO models

from €559

Mixed-signal analysis. 2 + 16 channel MSO. 50, 70 or 100 MHz with deep memory.

For more information please visit www.picotech.com/PS470

Prices are correct at the time of publication. Sales taxes not included. Please contact Pico Technology for the latest prices before ordering. Email: sales@picotech.com. Errors and omissions excepted.

Elektor Magazine
Edition 5/2016
Volume 42, No. 477 & 478
September & October 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.

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

© Elektor International Media b.v. 2016
www.eimworld.com
Printed in the Netherlands



Copy that?



Assembling a magazine like the one you are about to start reading is like masonry: you keep your eyes firmly on the structure in front of you, with absolute confidence of a steady supply of clean bricks and fresh mortar being upheld behind your back, literally, as you keep your balance at a perilous height. The bricks are the approved article pages as they exit from our layout department, and the mortar, the advertisements and other contributed content. The final, net size of the structure you hope to erect in time equals 132 units. 132.00 actually, meaning the number is not open to discussion, or adjusting by a brick and a half in a dark corner and hoping no one will notice. Worse, the complete wall is very much in public view and each element is subject to deep scrutiny down to the grain of sand level by you, my reading audience.

If you stand too close to an object it happens that you don't see the larger picture, and you need to step back or come down the ladder every now and then. Some of you know that I am a licensed radio amateur; still others say I talk more to my soldering iron and rigs than to my fellow hams on the air. Whatever, while doing the pagination of this magazine as part of my daytime job I noticed my internal S-Meter starting to deflect. It reached a good S7 or so when I discovered that "radio" got more than average coverage thanks to just the right bricks for me to grab and drop in place. Rohde and Schwarz on page 36 cater for the top professional contribution. On page 60 Burkhard Kainka puts the S in his SDR. The ESP8266 goes on air on 2.4 GHz on page 84, with no microcontroller in sight. On page 102 the FETron hails from the distant past, cleverly disguised as a DG-MOSFET. More happy memories of radio days with Magic Eyes on page 21. Can Red Pitaya on page 55 improve SSB reception with its IIR functionality? Or the Unilab Mk.2 on page 50 power your mobile rig and not go haywire from the strong RF field? Sure, copy that.

73, Jan Buiting, PE1CSI, 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. 477 & 478 September & October 2016

6 **The Elektor Community**

34 **ElektorBusiness: News & New Products**

36 **ElektorBusiness: Mobile Communication Instrumentation**

40 **THE BIG RED**
Call for Papers for Elektor's 2017 Yearbook.

110 **Elektor Store**

126 **Elektor World News Special Edition: electronica Fast Forward Award 2016**

128 **Elektor World News**

130 **Play & Win: Hexadoku**

LEARN | DESIGN | SHARE

8 **Welcome to the LEARN section**

9 **Tips & Tricks from readers for readers**
Runtime measurements.

10 **Home Automation**
Here we compare ten platforms with a promise of building a smart home.

21 **Peculiar Parts, the series: Magic Eye Indicators**
The iconic tuning indicator on tube radios.

22 **CircuitMaker Tips & Tricks (3)**
We make a board and generate some Gerber files.

24 **PIC Assembler Crash Course 2.2**
Indirect addressing of the PIC registers.

28 **LED Control Tabletized**
Turn a strip of LEDs on and off via your tablet or PC, with the help of Flowcode 6.

31 **Peculiar Parts, the series: Mercury Relays**
A remarkable application of that dangerous silvery liquid.

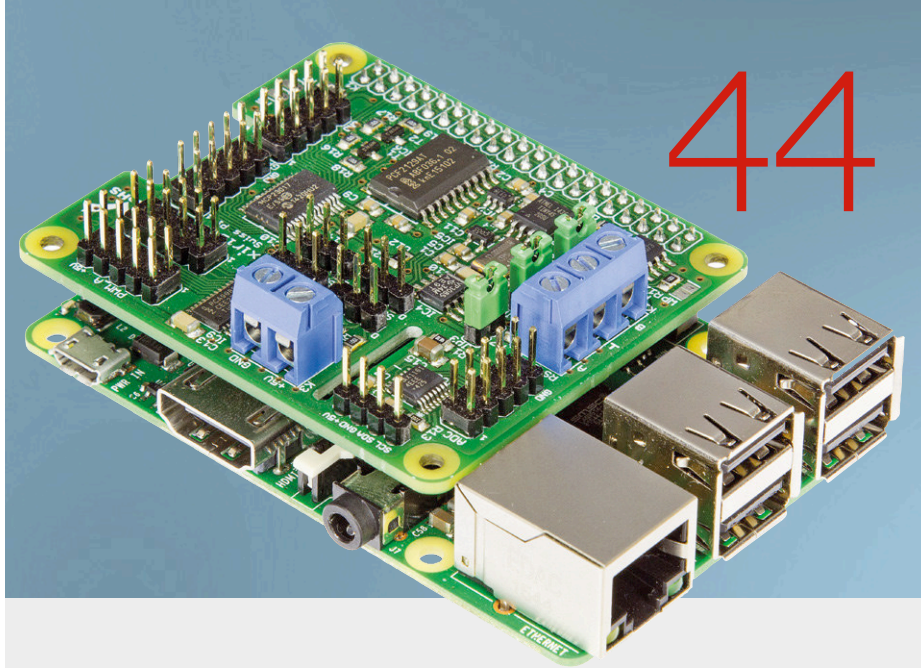
32 **EAGLE Tips & Tricks (5)**
Closing the series we discuss and use EAGLE's XML file format.

LEARN | DESIGN | SHARE

42 **Welcome to the DESIGN section**

43 **Linked: T-Board 28 and Serial e-BoB**

44 **Swiss Pi**
From the Labs, here's a veritable Swiss Army Knife for the Raspberry Pi.



44

50 **UNILAB Mk.2**

The design presented here can be regarded as a successor to the original UniLab from 2010. The maximum output voltage is again 30 V, and the maximum output current is a bit more than 3 A (up to 5 A is possible with some restrictions). For the beefed up design we opted for a different step-down converter IC, in this case the LM2677-ADJ.



50 **UniLab Mk. 2**
The 2010 version of UniLab gets a workover.

55 **Filtering on the Red Pitaya (2)**
Infinite Impulse Response (IIR).

60 **Elektor SDR Reloaded (2)**
Choosing software for our SDR Shield.

64 **LEDitron**
7-segment display made from LED filaments.

70 **UV Exposure Unit with PIC-based Timer**
Simplicity, downward exposure, and consistency of timing.

72 **i-Baxandall**
Tone control on audio amplifiers.

80 **Timeshift Radio**

84 **Compact and Self-contained WLAN**
ESP8266 without a microcontroller.

90 **Master/Slave LED Striplight Dimmer Unit**
How about controlling a couple of LED strings at large distances like 300 feet away?

21



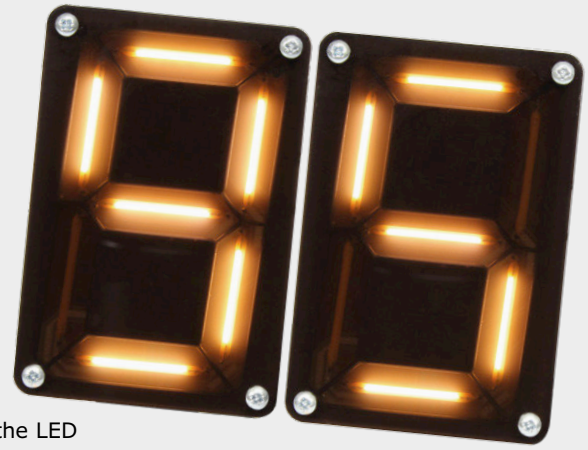
Swiss Pi

A Swiss army knife for the Raspberry Pi

One thing that has certainly contributed to the popularity of the Raspberry Pi is that the 40-pin GPIO connector makes it very easy to extend the basic board with custom modules. We felt challenged and developed a multi-purpose add on board.



64 LEDitron



Conventional low-energy lamps are a dying breed as we have entered the LED era. LEDitron is a seven-segment display module made from LED filaments, similar to a vintage Numitron tube but with better energy efficiency. Also read the LED Filaments article in the SHARE section.

MAGIC EYE INDICATORS



92 Baroduino
An Arduino-based 48-hour recording barograph featuring Storm Alert.

94 A simple Current Sensor Probe
Now you can do floating current measurements with ground-references oscilloscopes.

97 A Low-Battery Indicator
Cheerful and simple, for 9-volt batteries.

98 Poor Man's Capacitive Sensing Pad
Is this the simplest ever capacitive touch pad for use with micros or FPGAs?

102 Picowatt Transmitter
Admit, admit! You have never heard of the incredible Fetron!

104 Opamp Experimenting Kit for MyDAQ
This versatile kit from National Instruments allows you to tinker with 10 different configurations.

elektor magazine

LEARN | DESIGN | SHARE

114 Welcome to the SHARE section

115 Elektor Labs Pipeline
A constant stream of projects and activities in the Labs.

116 SmartScope Meets LabVIEW
A smart mix of VI blocks and custom made software result in a fantastic user interface.

120 From the Labs: Fragile LED Filaments
Let's put many LEDs on a glass carrier, shall we?

122 Retronics: MOS Clock 5314 (1974)
A golden oldie from Elektor's past, now with The Customs Story — and printed in color!

NEXT EDITION

My Tiny Radio

One of Seed Studio's 'Grove modules is a complete VHF receiver controlled fully via its I2C interface. The receive frequency bandwidth extends from 50 MHz to 150 MHz including support for RDS/RBDS and with a stereo jack socket for direct headphone connection. The module uses so little power that battery operation is feasible.

Analog Random Numbers

It's easy enough these days to generate random numbers digitally using a computer program. But there is also an analog approach: processing the noise produced by a transistor junction using an Arduino. The random numbers so generated can be used in various interesting experiments and will even lead us a little into the world of mysticism.

DoorSpy using Raspberry Pi

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.

Elektor Magazine edition 6 / 2016 covering November & December is published on October 17, 2016.

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

- ✘ 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

GOLD
€7.58 per month
£5.50 / US $8.42

- ✔ 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/gold

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

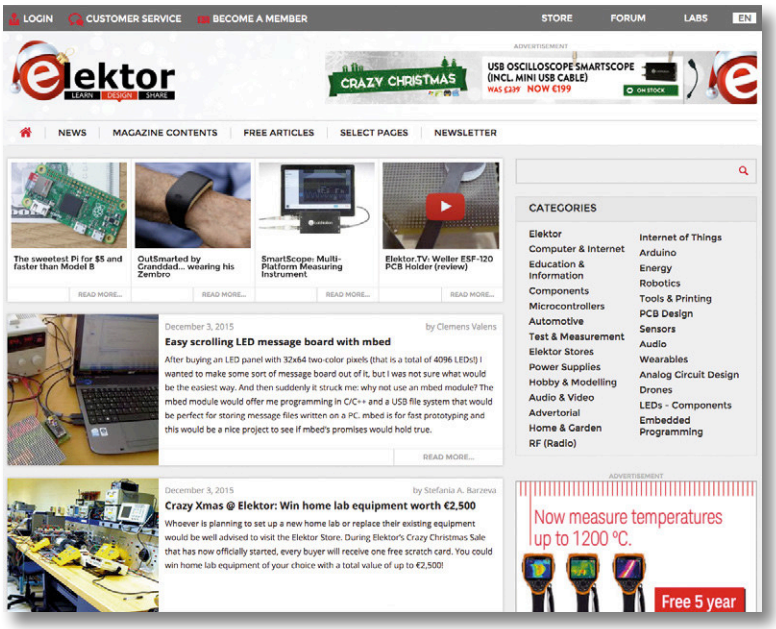
www.elektor.com/newsletter



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.

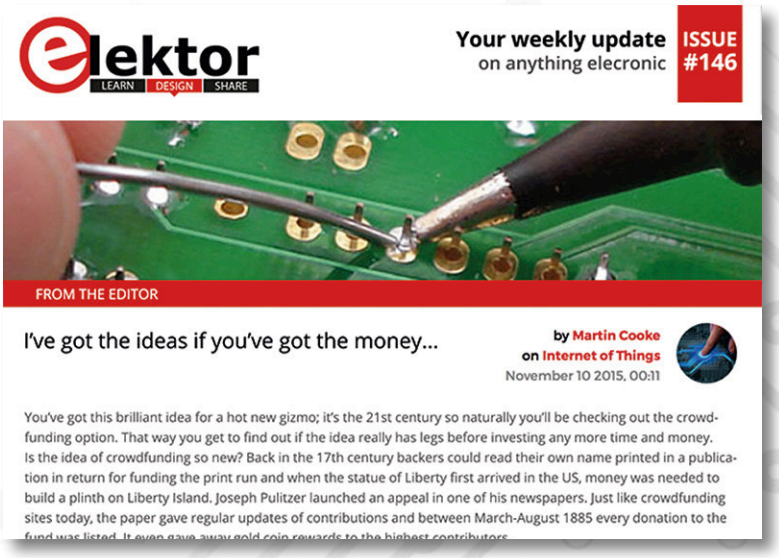


elektor Weekly & Paperless

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



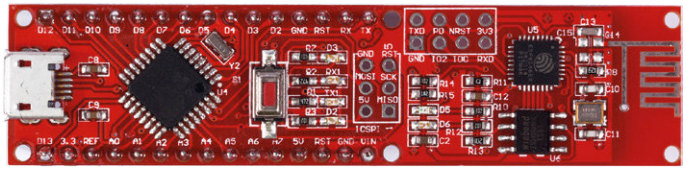
Welcome to the LEARN section



By **Jens Nickel**

First steps with the Pretzel Board

Many of you will probably remember from a previous LEARN editorial that my intention was always to try out Fabian Kainka's 'Pretzel Board'. This is a very interesting little development board which has all the ingredients to make it especially attractive to the maker community. First off it includes an ESP8266 WLAN module controllable via a serial interface using AT commands and an ATmega328 preprogrammed with an Arduino bootloader. All you need to start developing is the free Arduino IDE running on your computer and a micro USB cable, and so it was on one wet and windy day during my holiday when I set out to explore this little board. I'm sure, like me, if you've ever hooked up devices to wireless networks you can expect a certain degree of hacking will be necessary. Not so with the Pretzel Board; I was pleasantly surprised how quickly I was able to get the board talking to my home network.



Using simple commands which I entered using a terminal program I was able to act as a browser and request HTML code from a web server on the big wide Internet. Then using my own TCP server application the Pretzel board was able to send characters to my PC. Follow the series *My Journey into the Cloud* in the news section on www.elektormagazine.com.

Sensors for Dummies

This morning I put in a call to Fabians father, otherwise known as Burkhard Kainka. Berkhard lives in Essen and I always look forward to our chats. We often come up with new ideas when chewing over the latest trends in electronics. I am thinking that in the future more and more projects are likely to be made up of an assembly of off-the-shelf electronic modules; how the finished system behaves will all be down to software. The detailed operation of each individual component is of little interest to the overall system function. That doesn't mean that such information is superfluous however. Coming soon from Burkhard will be a short series of articles demonstrating how the 35 sensors in the Arduino sensor kit (www.elektor.com/arduino-sensor-kit) can be used in a number of interesting measurement applications. We decided he would not only be covering the software in detail but also describing the circuitry used on each sensor board. Burkhard's easy and informative style will make this series one not to miss. Look out for them in upcoming editions of LEARN.

Home Automation – it's a jungle out there

The subject of home automation is one that's dear to the heart of many of our readers. Anyone with enough spare time and expertise can build their own system from the ground up. In fact one of our readers in the Netherlands built his own system and has a thriving community who contribute with their own system improvements and additions. (see www.elektormagazine.com/labs/internet-of-things-gateway-and-end-nodes, look out for a review of the system which we shall be running soon).

Readers looking to automate their own home will most likely be using off-the-shelf modules. There is a huge selection of (not necessarily compatible) modules available from different manufacturers. Elektor freelance contributor, Rolf Gerstendorf has taken up the challenge to carve out a path through this jungle. On the following pages he takes a look at ten different home automation systems. Amongst other important system features he considers how easy it is to expand and develop the systems to meet the needs of individual requirements. ⏮

(160070)

Tips and Tricks

From readers for readers



Here is another neat solution from one of our readers, sure to make life a little easier for engineers and electronics tinkerers alike.



Runtime measurements

by Michael Gaus

In time-critical applications it is often necessary to get an accurate measurement of the time it takes to execute a section of code. In many cases you can use a spare I/O pin as a debug output, setting it high at the start of the critical code section and low at the end. An oscilloscope can then be used to measure the pulse produced on the pin.

Derivatives of the Cortex-M3/M4/M7 processor family offer a much simpler method. They have a dedicated DWT (Data Watchpoint and Trace Unit) which contains a counter clocked from the processor clock (the STM32 family such as the STM32F1, STM32F2, STM32F4 and STM32F7 are examples). This gives you the ability to measure the run time of a section of code without using any additional resources. The counter can be read via the DWT->CYCCNT register. The timing resolution is equal to a processor clock cycle so with a clock of 168 MHz it will be 1/168 µs, i.e. 6 ns. The register is 32 bits wide so at 168 MHz it will be able to measure periods of around 25 seconds before it overflows and starts from 0 again.

If it's possible for another interrupt to occur during the measurement period it will upset the measured time. In this case, to get an accurate measurement make sure to disable interrupts during this critical timing period (the same thing will happen if you're using the method which toggles a port pin).

Advantages:

- *It doesn't take up a spare port pin.*
- *No need for a scope or logic analyzer.*
- *It uses the dedicated DWT-Counter so that all other counters are free for use.*
- *During debugging the measured code run time is true even when break points are used.*
- *Nested run time measurements are possible when separate variables are used.* ⏮

(150533)

Code example:

```
#define PROC_FREQ_MHZ  168  // Processor Clock Frequency in MHz
#define CoreDebug_DEMCR_TRCENA_Msk 0x01000000
#define CYCCNTENA 0x00000001

void enable_DWT(void)
{
   CoreDebug->DEMCR |= CoreDebug_DEMCR_TRCENA_Msk; // Activate DWT access
   DWT_CTRL |= CYCCNTENA; // enable CPU Cycle Counter
}


void testFunc(void)
{
   volatile uint32_t timestamp1, timestamp2, runtime;

   enable_DWT(); // activate DWT

   ... // you can mask interrupts here

   timestamp1 = DWT->CYCCNT; // save first time stamp

   ... // Insert here the section of code you want to time

   timestamp2 = DWT->CYCCNT; // save second time stamp

   ... // re-enable interrupts here.

   runtime = (timestamp2 - timestamp1) / PROC_FREQ_MHZ; // The time given in µs
}
```



Got a neat solution for a tricky problem? Using components or tools in ways they were never intended to be used? Think your idea to solve a problem is better than the usual method? Have you discovered a work-around that you want to share with us and fellow makers? Don't hang around; write to us now, for every tip we publish you'll earn 40 Euros!



Home Automation

Ten platforms examined



By **Rolf Gerstendorf** (Germany)

There are several companies competing in the world of home automation, offering various communication standards and frameworks. Here we compare some of the possibilities for building a 'smart home'.

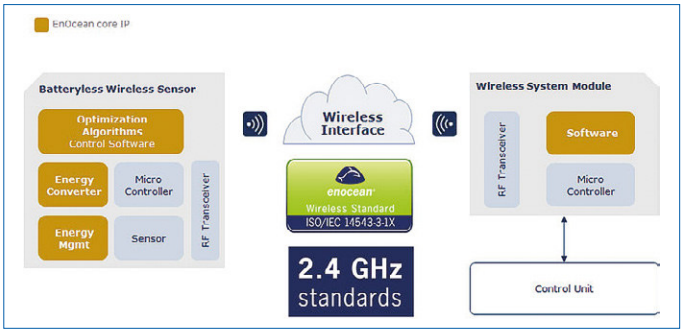
The time is long gone when all that home automation meant to the man in the street was a geek's refrigerator automatically ordering more beer on the Internet when supplies were running low. Serious applications, such as security and monitoring, control of heating and air conditioning, lighting management and multimedia streaming have all become feasible with rapid advances in technology. The 'smart home' is no longer confined to the realm of science fiction but features in the plans of architects, builders and DIY renovators.

Many companies have recognized this market opportunity and have developed home automation systems and devices. In a short space of time a large number of standards have come to the fore, which, needless to say, are mutually incompatible. There is a wide choice of communications media, methods of operation, safety standards, concepts, platforms and systems. Some of these are proprietary and some allow, or even encourage, the development of homebrew software and hardware components. It's a jungle out there, through which the would-be smart home builder must beat a path.


Welcome home
Orange 3G
Wifi
5:45pm
Mon 23.08.2010
London
22°C
Lighting
Climate
Scenes
Equipment
Rooms
Multimedia
Services
Security
Unassigned

EnOcean

EnOcean develops radio devices that do not have batteries. The company produces and sells self-powered radio modules for use in buildings, in home automation and in IoT applications. EnOcean's products are based on miniaturized energy-harvesting devices, low-power electronics and their own radio standard operating below 1 GHz as well as existing standards in the 2.4 GHz band (ZigBee, and, in the future, Bluetooth). The key feature of the EnOcean technology is that the energy to power the radio module in the device is obtained at the point where it is required using energy harvesting. The source of the energy might be motion, light or a difference in temperature. The amounts of energy involved are tiny and so, for this to work successfully, the power consumption of the device must also be tiny. This involves not only low-power hardware, but also a well-thought-out approach to energy management and power-efficient communication.
The EnOcean Alliance is a group of companies that offers an ever-growing range of EnOcean-based batteryless remote switches and sensors operating in the sub-1 GHz band. The interoperability of products based on the EnOcean radio standard is guaranteed; the standard has a strong emphasis on lighting, heating and air conditioning control applications.



Hub: Various manufacturers who subscribe to the EnOcean radio standard produce gateways designed either for DIN rail mounting or for stand-alone use. These devices offer a TCP/IP connection to a router and thence to the Internet, and also allow connection to bus systems using other protocols such as KNX, DALI and Modbus.

Devices: EnOcean develops and markets radio modules that do not require a battery, allowing manufacturers to produce reliable and maintenance-free radio sensor systems, including, for example, switches with or without encryption, magnetic contact switches, motion sensors and temperature sensors. For hobby development the most interesting components are the various energy harvesters and radio modules. There are some 150 manufacturers who make devices to the EnOcean standard for the European market. The website of the EnOcean alliance shows the enormous range of products available, from radio sensors for temperature, position, motion, light, humidity, air quality and leak detection, to actuators such as dimmers, relays, servo motors and space heating controllers.

Apps: The manufacturers of EnOcean gateways bundle software that allow the control of an installation using a local PC or remotely using a smart device.

Protocol and security: The protocol is designed to minimize power consumption and is relatively slow. It is therefore not ideal for data-intensive applications. Devices operate in the SRD (short range devices) 868 MHz band, normally unencrypted; for security-sensitive applications AES-128 encryption is used.

Development: EnOcean offers two development kits. The ESK 300 starter kit demonstrates energy harvesting and ultra-low-power radio technology. It includes a switch module, a temperature sensor, a device to harvest energy from motion, a radio module, a USB transceiver/gateway, and accompanying Windows software called DolphinView Basic which allows the radio messages to be viewed.



The EDK 350 kit gives a deeper insight into the power of the Dolphin platform. The kit allows one to develop one's own energy-harvesting applications. The kit covers the complete product range from energy harvesting and radio modules to complete product designs. DolphinStudio, DolphinView and DolphinAPI are available for download to help develop your own applications; firmware can be developed using the Keil uVision integrated development environment.
As well as the two development kits there is much information to be found on the website including system and hardware specifications, tutorials, application notes and various software and firmware packages, including free trial versions.

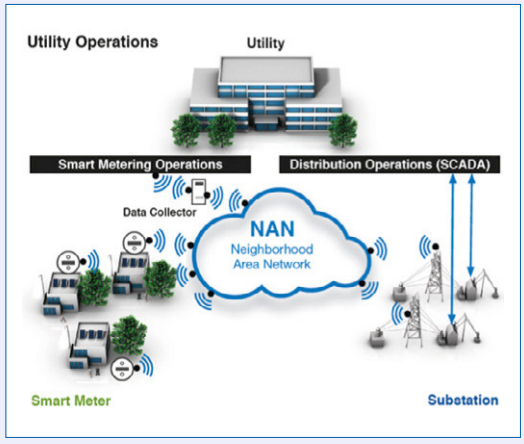
Pros and cons: Maintenance-free, well-thought-out and modern radio system for use in building and home automation; comprehensive website and thorough product documentation; home development is well supported.
Strong emphasis on automation of lighting and on heating, ventilation and air conditioning (HVAC) applications.

Web: For consumers: www.enocean-alliance.org/en/products/ and / www.enocean-alliance.org/en/building_professionals/
For developers: www.enocean.com/en/support/

ZigBee

ZigBee is a particularly simple and flexible radio standard designed for home automation as well as for other applications. It is highly energy-efficient, which makes it ideal for use in battery-operated devices. The problems of incompatibility arising from the large number of different standards and specifications, often a cause for complaint, are now a thing of the past with the introduction of profiles for home and building automation, for lighting control, for health care and for other applications. This means that all participants in a network that fulfill the requirements of a particular profile will be able to communicate with one another. The most relevant profiles to us are the ones called Home Automation, Light Link, Smart Energy, and perhaps Health Care.

Hub: The hub in a ZigBee network is called the 'Coordinator' (ZC), and it simultaneously acts as the root of the network and provides the connection to other networks such as the Internet. The Coordinator stores information relating to the network, including security information. Possible network topologies include star, tree and mesh: in the last of these arrangements ZigBee Routers (ZRs) are used to extend communications at the network level. A router forms a node in the network that can forward data packets between the Coordinator and a leaf device. The leaf devices themselves do not perform any routing functions: they can only send and receive data.





Devices: Some 240 certified 'Home Automation' devices are available, including gateways, hubs and leaf devices. The list on the ZigBee website can only serve as a starting point to discover more about these certified products and their manufacturers, as the site only describes them briefly. Further information is lacking, and sometimes there is not even a link to the manufacturer. However, a little research will be rewarded: there is a vast range of devices of all kinds to be discovered.

Protocol and security: The ZigBee system uses the 2.4 GHz band for communication, which is available worldwide. Communications are encrypted using AES-128 in CCM mode according to IEEE 802.15.4.

Development: The ZigBee standard is completely open and designed for independent development. There are countless compliant platforms, controllers and boards available for every conceivable application: see the 'Products/Compliant Platforms' section of the ZigBee website. Many platform manufacturers offer protocol stacks and software development kits for download from their websites.
All the application standards and network specifications, as well as white papers on the ZigBee gateway standard, are available from the ZigBee website developer pages.

Pros and cons: An open standard with many options for tools and development, giving (depending on the developer's experience with microcontroller programming and applications) an infinite range of opportunities.
Cumbersome and complex search for compatible devices. Although the 2.4 GHz band is available worldwide, it is shared with WLAN, Bluetooth, baby monitors and other devices.

Web: For consumers: www.zigbee.org
For developers: www.zigbee.org/zigbee-for-developers

Z-Wave

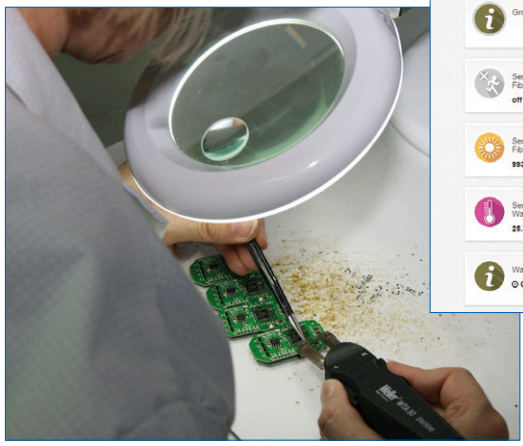
Although the Z-Wave radio standard was originally developed in Denmark, the Z-Wave Alliance is based in the USA. This consortium consists of about forty member companies, offering a range of over 1500 different devices and controllers. The Z-Wave system is ideal for DIY installation (which is not an insignificant point as in many countries there are no registered Z-Wave installers!). A wide range of manuals and tutorials is available in various languages to help with the planning and installation of a Z-Wave home automation system.

Hub: There is a range of cloud-based 'smart hubs', hubs, gateways and home controllers, in some cases involving recurring subscription costs.

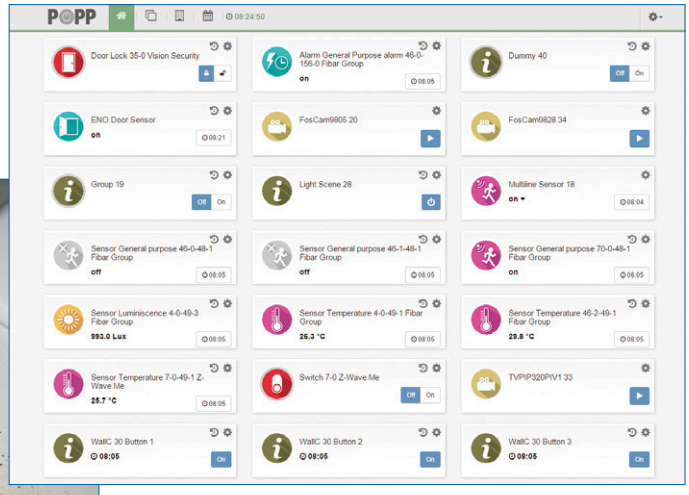
When making a selection it is important to verify that the device in question complies with relevant regulations applicable in your country. Of particular interest are the three hubs made by HomeSeer in the USA: out of the enclosure of the tiny HomeTroller S2 poke the USB sockets of a Raspberry Pi 2! The controllers available differ in what drivers and plug-ins are available for various functions, as well, of course, as in price. Many functions can also be purchased as add-ons. Special USB sticks are also available to allow 'foreign' controllers to be integrated into a Z-Wave network.

Devices: The same warning applies to devices as to controllers: not

everything that is on sale is suitable for use in every country, as the frequency bands available for short-range devices vary. Thus seeking out and choosing something from the great variety of devices on offer is not straightforward. On the international website it is possible to sort devices by function, but not according to the frequency band used, while on some of the localized sites it is the other way around.



Various categories of battery-operated devices are available, including lighting and dimmers, switches and buttons, security devices, sensors, heating thermostats, and controllers for blinds. Other interesting and more unusual offerings include energy meters, remote controllers, flooding alarms, water flow control valves, light fittings with built-in radio controllers, and much more.



Chips: A dedicated Z-Wave SoC is used as the main controller. This integrates a radio transceiver, an 8051 (!) -compatible microcontroller and a wide range of peripheral interfaces.

Apps: Most of the controllers, hubs and gateways come with smartphone apps and PC-based software for control, monitoring and configuration.

Protocol and security: Z-Wave uses frequency-shift keying (FSK) with a 20 kHz deviation and data rates of up to 100 kbit/s depending on the frequency. The radio transmit power is limited to a few milliwatts, except in the 868 MHz short-range data band where up to 25 mW is permitted. Nevertheless, range within buildings can be up to 40 m. Z-Wave uses a standard coding system for communication; for security-related applications such as door locks or skylight openers a separate transaction authentication number (TAN)-based system is used.

Development: The developers' website offers a development kit for Z-Wave to IP gateways, which consists of a USB bridge controller and a BeagleBone single-board computer, C libraries, an API plug-in and Z-Wave apps, and a kit for developing the hardware and software for end devices, which includes a hardware kit, software development tools and a region-specific kit. Individual Z-Wave hardware modules are also available.

Pros and cons: The Z-Wave system offers a wide choice of devices and thanks to the large amount of support available it is easy to install yourself. It is also very cost-effective. Developing your own controllers and end devices is well supported.
The descriptions of devices and hubs on the Z-Wave site are rather vague; more precise information, such as a datasheet, has to be obtained from the manufacturer in question. It is inconvenient to have one's choice of devices limited by which carrier frequencies are permitted in the country of installation.

Web: For consumers: www.z-wave.com
For developers: z-wave.sigmadesigns.com

Loxone

Loxone is an Austrian company that produces its own complete system for home automation. The wired system is designed for new electrical installations, but it can also be extended wirelessly using radio technology. It is also possible to convert an existing electrical installation into a 'smart' installation using the Loxone system.

Hub: Loxone offers two central severs, one for retro-fitting to existing electrical installations and a mini-server for new installations. A proprietary operating system is used that includes a web server. The server can be connected either directly to a computer or via a router, and then the Loxone Config program used to set up the system.
The mini-server, which comes in a DIN-rail enclosure, is similar in construction to a PLC controller. It has eight relay outputs rated at 250 V/5 A, eight 24 V digital inputs, and four analog inputs and four analog outputs, all operating from 0 V to 10 V. Since all devices must be connected via cabling, a direct retro-fit of this system into an existing electrical installation is likely to be difficult. Instead, retro-fits are done using separate radio products. The server also includes a LAN socket for connection to a router and a KNX/EIB interface for connecting to KNX sensors.

Extensions: 'Extensions' can be added to the server to expand its capabilities. Again, these are DIN-rail-mounted. There are dedicated extensions to handle DALI, DMX and dimmer control, external interfaces including Modbus for energy meters, RS232, RS485, 1-Wire bus and infrared, and universal extensions offering more relays and/or digital inputs and outputs. Two extensions are available that add radio connections to the server, Loxone Air for up to 128 devices, or EnOcean for up

to 128 sensors and 128 actuators. These are particularly useful when upgrading or adding to an existing installation. The 'Tree' extension allows groups of devices to be divided into separate branches, which considerably reduces the amount of cabling required.
The server and extensions, along with their power supplies, take up a considerable amount of space in the consumer unit; normally a dedicated extra cabinet is not used for the Loxone system.
The 'Go' mini-server is equipped with its own radio module (operating on 868 MHz) and so is better suited to retro-fit applications. The server can control Loxone Air devices wirelessly. It is also possible to make wired connections to extensions (and thereby control all types

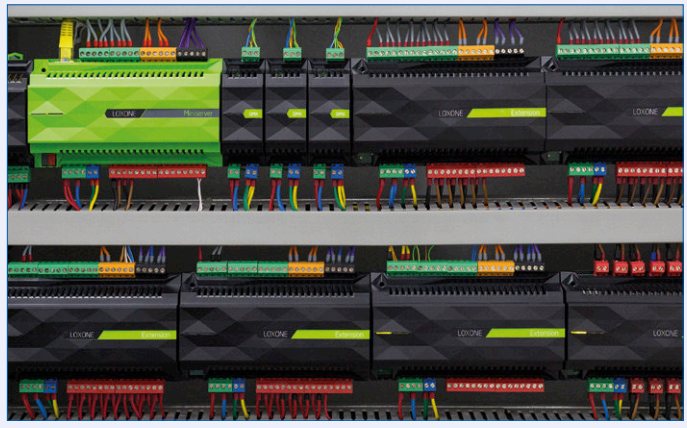


of Loxone device), but this makes little sense if the router, which must be connected using a wired connection, cannot conveniently be placed near to the consumer unit where the extensions are located.

Devices: Loxone offers devices covering a wide range of application requirements: lighting, heating, entry control, alarms, energy monitoring, and sauna and pool control. These devices are available in both wired and wireless ('Loxone Air') versions. The only devices currently supporting the new 'Tree' technology are a button, a motion detector, and a servo valve for heating control.
Sensors available include a motion detector, a water sensor, door and window contacts and a glass break detector, a button, a pressure sensor and various sensors for indoor and outdoor air condition and temperature, an electronic lock (using the Maxim iButton) and a smoke detector.
Actuators available include servo valves for heating control, relays, intelligent sockets and an infrared transmitter and receiver.

A special feature integrated into the Loxone home automation system is its music server. It allows the playing back of music and can also make system status announcements.

Apps: The Windows program Loxone Config can be found in the download area of the Loxone website. This allows the server to be configured, and comes with a large number of examples and videos, along with a general-purpose application for controlling devices from a smartphone (for iPhones from iOS 7, Android KitKat or later). There are also several planning aids and calculators to help architects and builders design a Loxone home automation system.



Protocol and security: Both wired connections and Loxone Air wireless connections (on 868 MHz) use the IPsec security protocol suite.

Development: A comprehensive API is available to help you develop your own visualization software, and indeed this API is used in the Loxone app itself. For lower-level programming the PicoC scripting language is available.

Pros and cons: Secure wireless data transfer, easy expandability using short-range radio technology, and APIs to help in the development of your own software and hardware.
Configuration program only runs under Windows.

Web: For consumers: www.loxone.com/enen/start.html

HomeMatic (eQ-3)

The German eQ-3 group offers a range of automation products for controlling various functions in a house or apartment under its 'HomeMatic' brand. There are more than eighty different components in the HomeMatic system, and they are linked by short-range radio to a hub from where they are controlled, monitored and configured.

Hub: The HomeMatic CCU2 hub is the central component, carrying out a wide range of tasks relating to the control, monitoring and configuration of all the devices in an installation. The hub communicates with the devices by short-range radio on 868 MHz.



The hub in turn is controlled from a PC using a web browser running the HomeMatic WebUI user interface, which offers a comprehensive set of functions. It can configure and control the HomeMatic devices, poll their status, create direct links between devices which remain active even when the hub is not, and even define complex control commands and individual logic functions.
If the hub is connected to a server or router, the user interface can be reached over the network. Otherwise, it is also possible to access it directly via a USB interface.
The hub includes a WLAN gateway to allow connection to the Internet



and a status display that operates independently of the PC. It is also possible to configure the components without using the hub over the LAN or using a wireless adapter.

Devices: More than eighty different HomeMatic devices are available, covering the areas of heating and air conditioning (radiator and wall thermostats, indoor and outdoor temperature), lighting (switches, buttons and dimmers), blinds and awnings (motor control), security and monitoring (smoke detectors, motion sensors, water detectors, alarms, tilt sensors, shock sensors), doors and windows (contact switches, locks and automatic window openers) and remote controls.

Apps: HomeMatic is programmed and controlled via its web user interface. A range of (paid-for) apps from third parties is available under the 'Software and Apps' menu: these allow control and monitoring from a smart device. Android, iOS and Windows are supported.

Protocol and security: The platform uses short-range data (SRD) on 868 MHz with AES-128 encryption.

Development: The Windows program 'homeputer CL Studio' makes it easier to program complex control algorithms into a HomeMatic system. The software, which is entirely in German (even the macro programming language uses German), allows you to build special web pages for PCs and smartphones as well as iOS apps.
Hidden away in the download area of the eQ-3 website there are some scripts and specifications of the API and the dedicated HomeMatic programming language.

Pros and cons: HomeMatic systems are readily available, well documented and low-cost. Many devices are available, but there are relatively few special-purpose solutions, such as the ability to control Hue lights. There are hidden costs for the apps and for the programming software.

Special features: The new HomeMatic IP system is being introduced alongside the HomeMatic system described above. The new system has no hub; instead, devices connect to a cloud service via an access point and Internet router in order to be configured. Control and monitoring is done using apps (for iOS and Android). At present only devices are only available in the areas of heating and air conditioning and for security and monitoring. Products for controlling lighting and blinds should follow soon.

Web: For consumers: www.homematic.com/ http://www.eq-3.de/
For developers: www.eq-3.de/service/downloads.html (German only)

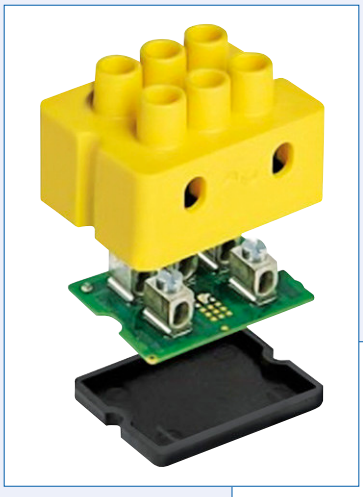
digitalSTROM

The Swiss smart home manufacturer digitalSTROM uses the regular power cables for data communication between one network component and another, and with the hub. To this end each device to be controlled is fitted with a small brightly-colored intelligent connection block called a digitalSTROM switch, similar in appearance to a Lego brick or a terminal block found in a junction box, which contains a high-voltage chip connected to the power lines. No additional cabling is required. Intelligent IP devices are connected over WLAN; the control units, which include the server and a digital power meter, come in DIN rail mounted enclosures and can be wired directly into the consumer unit after the main fuse.

Hub: A digitalSTROM meter is connected in each circuit, acting as a bus master for communication with digitalSTROM devices and digitalSTROM switches. To allow communications from one power circuit to another the digitalSTROM meters can be connected to one another over a two-wire RS-485 bus.
The server is available in DIN-rail-mounted and external variants, and provides a platform for digitalSTROM apps. It is an optional component, but in practice is indispensable. It provides a connection to the Internet (allowing remote access from a computer or smartphone) and allows local access over the home network to configure the digitalSTROM applications. The server is connected to the meter using an RS-485 interface.
A mains filter ensures that communication over the power lines is not affected by interference.

Devices: The digitalSTROM switches allow various types of device to be

connected and controlled. There are ready-configured standard behaviors to enable the fully-functional networking of switches, buttons, sensors, lights, blinds, alarms and electrically-driven heating control valves. The functions of the switches (which include turning on and off, dimming, communicating, executing simple programs, and measuring current) are configurable, and the switches can be linked to other devices as desired to implement a wide range of possible scenarios.



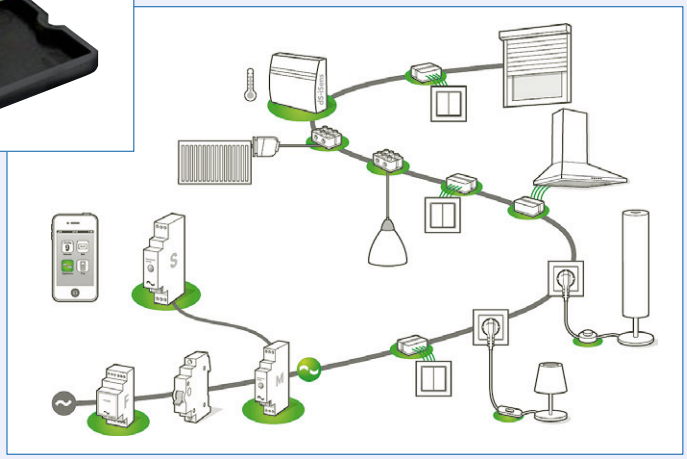
Besides the individual switches there are also pull-switches and dimmers, relay and lighting control modules, a servo driver and other modules with a built-in digitalSTROM chip. Various other manufacturers also produce lamps with digitalSTROM chips incorporated into them, which can be independently integrated 'out-of-the-box' into a digitalSTROM system.

Apps: The digitalSTROM website currently offers twelve apps and services for free download, encompassing a wide range of functions including time switches and occupancy simulation, and to enable the system to send notifications. They require no specialist knowledge to install. An NFC transponder can also be added to a digitalSTROM system, which lets an NFC-equipped smartphone with a suitable app initiate preprogrammed actions.

Protocol and security: Being a wired system it is more secure against attack. Communication between devices and the meter is over the house's power lines; RS-485 is used between the meter and the server. WLAN communications are encrypted.

Development: The digitalSTROM developer website contains a wide array of documents on how to build a system and the interfaces (APIs) used. Links are provided to ancillary material on the development of the open digitalSTROM server software, and some code examples are also given. A 'reference platform' is available to assist with the development of your own hardware and software.

Pros and cons: Secure data communication over the power line network, few modifications required to an existing installation, practical DIN rail mounting for the control hub. Very expensive.



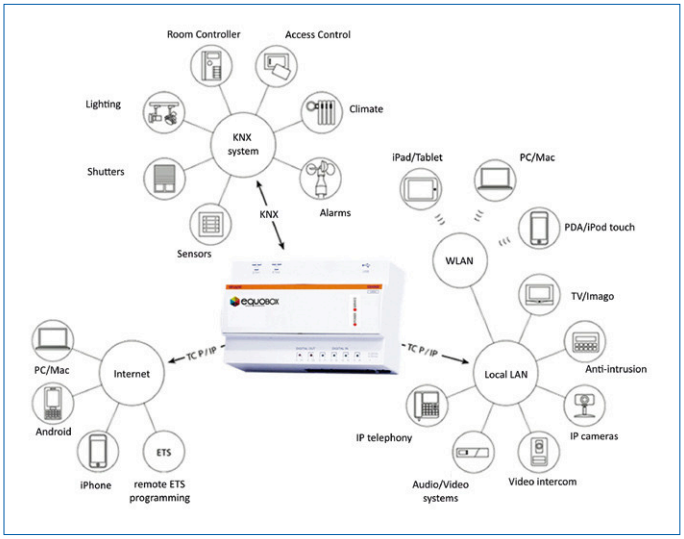
Web: For consumers: www.digitalstrom.com/
For developers: www.digitalstrom.org/en/alliance/developer/

KNX

The KNX Association is a group formed on the basis of three bus systems developed for building automation (EIB, EHS and BatiBUS). KNX is the leading open standard for home and building automation technology. The KNX organization itself does not offer any products, but instead defines the standard for devices, ranging from controllers for lighting and blinds to security and energy management systems, controllers for heating, cooling, ventilation and water, various monitoring devices, meters and alarms. The scope of the standard even extends to domestic appliances and audio and video systems.

Worldwide more than 370 firms are members of the KNX Association, developing and selling devices and software that employ the KNX standard. KNX is not limited to any one particular communication medium: possibilities include twisted-pair cables, radio, power line signaling and IP over Ethernet. Each bus device is either a sensor or an actuator, whether it be for lighting, control of blinds and louvers, security, energy management, maintenance and monitoring of a building, remote control, gathering meter readings, audio and video control, domestic appliance control or any other purpose. All these functions are controlled using a common system, secured and monitored by alarms, without the need for the additional control hubs required by many other home automation systems.

Configuration: The KNX standard has two different configuration modes. System mode ('S-mode') is aimed at experienced KNX installers, and allows the full power of the system to be realized. An S-mode installation can be planned using the ETS Professional software on the basis of





product database entries. ETS is then also used to connect the products together and configure them.
Easy mode ('E-mode') is intended for installers with only a basic grounding in the KNX system. E-mode devices have limited functionality and come ready-programmed with default settings for their parameters. A simple configuration tool allows some degree of customization of each device, mainly in terms of adjusting parameters and connections.

Protocol and security: KNX can use a variety of different communications media. The twisted-pair bus (running at a bit rate of 9600 bits/s) originated as EIB; power line communication (using the PL110 FSK and PL110 OFDM standards) allows a bit rate of 1200 bits/s. KNX devices that use short-range radio communications (at 868 MHz, maximum power 25 mW) offer a maximum bit rate of 16384 bits/s. Data transfer can be unidirectional or bidirectional. KNX radio devices can be made using off-the-shelf components: the devices have a low power consumption and are suitable for small and medium-sized installations. Repeaters are rarely required. KNX messages can also be embedded in IP packets for transmission over Ethernet or WLAN on a local network or on the Internet.

Development: A MyKNX account is required before you can download the development software. Unfortunately the software is priced for professionals. To create a home automation system installation the ETS software is needed: there is a free demonstration version, but it is limited to just five devices.
An alternative for developers is the free Falcon software development kit. This is a library that enables you to build your own applications compatible with the KNX bus. The Falcon API provides device management services (essentially reading and writing the memory in a device), network management services such as scanning for and changing physical addresses, group data services such as turning a light on or off, and bus services such as sending or receiving a message over the KNX bus. Among the supported programming languages are C++ and C#.
If you wish to bring a KNX device to market it is necessary to test it using the EITT KNX interworking test tool and certify it using the KNX manufacturer tool. It can then be entered in the KNX product database, which is used by the ETS software. Purchasing these software tools is, however, only of relevance to commercial developers.

Pros and cons: Practically unrivaled selection of devices, sensors and actuators from many different manufacturers, choice of communications media for new and existing installations.
Planning, installation and development of new devices only practical on a commercial basis because of the costly software.

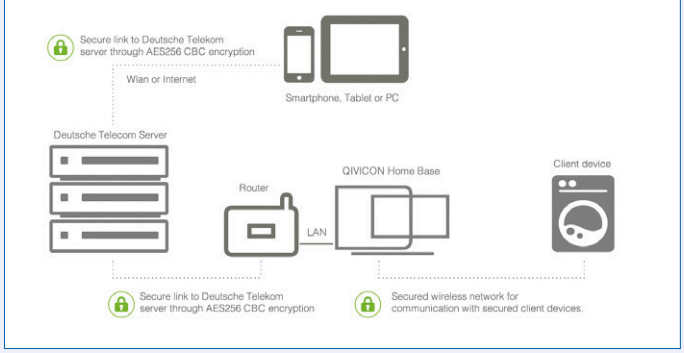
Web: www.knx.org/knx-en/knx/association/introduction/index.php

Qivicon

Qivicon is a group of companies under the wing of Deutsche Telekom, the German telecommunications company. Qivicon member companies include energy utilities, home appliance manufacturers, and security technology and telecommunications companies.

Hub: The hub of a Qivicon home network is called the 'HomeBase', which uses a secure radio connection to talk to devices and to a PC, tablet or smartphone. At the moment the HomeMatic protocol is used, but for radio connections to components using other protocols such as ZigBee a special USB adapter is required. The newest version of the HomeBase, which is due to appear this year, will have ZigBee and DECT protocols built in.
The HomeBase is also connected by LAN to the home's router. This allows devices to be controlled remotely over an Internet connection. After registration, the base station can be configured using a browser over the Internet or over the local network.





Devices: A wide range of smart home compatible devices is available, including buttons and remote controls, sensors, switched sockets, surface-mounted and concealed units for switching, dimming and data collection, thermostats and heating controllers, cameras (from D-Link), home appliances (from Miele), lighting products (from Osram and Philips) and products from Bitron and the eQ-3 ranges. An overview of the available devices can be found on the Qivicon consumer website. A special feature of the system is the bundles offered by individual device manufacturers and energy and telecommunications utilities, that provide a range of different basic hardware configurations for smart home concepts, for example focusing on security, energy or heating.

Apps: General-purpose apps are available, as well as apps specific to particular devices or manufacturers. Both iOS and Android are supported. The smart home apps from Deutsche Telekom, the eww group and ENTEGA allow any device given in the compatibility list to be controlled.

Protocol and security: The central HomeBase unit communicates with HomeMatic system devices using the BidCoS protocol developed by eQ-3. The radio link is bidirectional, requiring an acknowledgement of a command from the device, and is therefore very reliable. The system uses 868 MHz radio modules and so the radio link hardware is inexpensive.

Development: The Eclipse IDE is freely available to help with the development of compatible hardware and applications software. Eclipse, which is now an open-source IDE, was originally used for Java development. This set-up, Eclipse IDE for Java Developers, includes a runtime environment and a smart home designer tool. The Eclipse IDE is available for Windows, Mac OS and Linux (both 32-bit and 64-bit versions).

Web: For consumers: www.qivicon.com/en/
For developers: https://developer.qivicon.com/qivicon-developer, www.eclipse.org/smarthome/

Bluetooth Smart

Bluetooth is known for point-to-point communications between computers and devices, such as between a tablet and headphones, between a PC and a wireless keyboard, or between a smartphone and a car radio. The new Bluetooth Low Energy standard (BLE, also known as Bluetooth Smart) allows for extremely low power consumption in battery-powered devices, which can operate for up to four years without change of battery. The standard is therefore also suitable for use in a home automation network.

Hub: Programming and configuration is carried out using any smartphone or a tablet (such as the Archos Smart Home tablet) with a Bluetooth LE chip. A hub with LAN bridge is not required, unless it is desired to connect the network to the Internet. Although from a technical point of view there were no obstacles (the hardware is already available) a specification for a Bluetooth gateway has only appeared very recently.

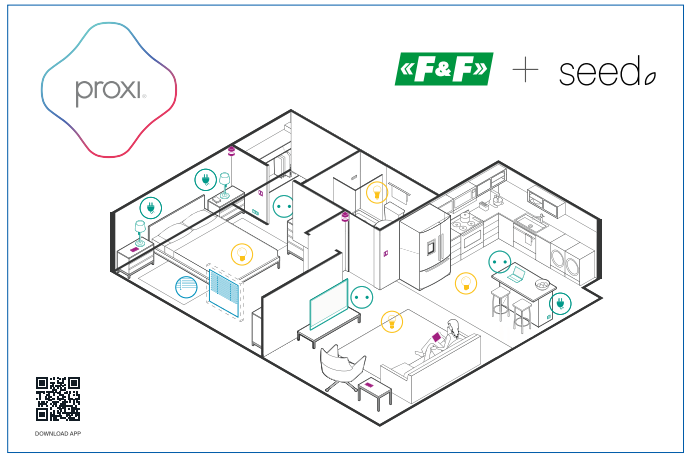
Devices: Only a relatively small number of devices are currently on the market, from a small number of companies, including Archos, Elgato, Euronic, Megaman, Schage and Seed Labs. Devices available include radiator thermostats, motion sensors, cameras, intelligent power switches, sensors for indoor and outdoor conditions, door and window sensors and smart LED lamps. Many further developments have been announced (including those for the compatible Apple HomeKit), but are not yet on the market.

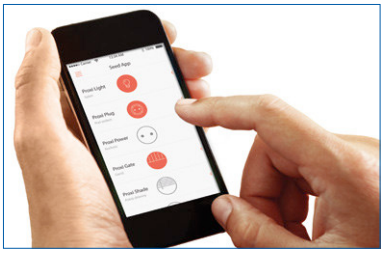
Apps: Most manufacturers of devices provide apps for iOS and Android; Elgato supports only iOS. Unfortunately there is no universal app yet.

Protocol and security: Bluetooth BLE, like standard Bluetooth, transmits in the 2.4 GHz band and uses 128-bit AES encryption.

Development: A knowledge of the Bluetooth 4.2 specification is a prerequisite for developing devices and software for home automation: see the website under 'Specifications'. In the developers' area, after registration, you will be able to find various free resources, software starter kits, testing tools and a link to the developer community. Also, several companies offer low-cost hardware platforms, modules and development kits.







Pros and cons: A low-cost and easy-to-expand system, good for open development. Operates worldwide using a familiar radio standard.
The range is too low for a large star-topology home network, but the introduction of a mesh networking standard promises an improvement in this regard. So far only a few devices are available, and there are no Internet gateways. Bluetooth Smart Home is still finding its feet, but with the participation of Apple that could change very quickly.

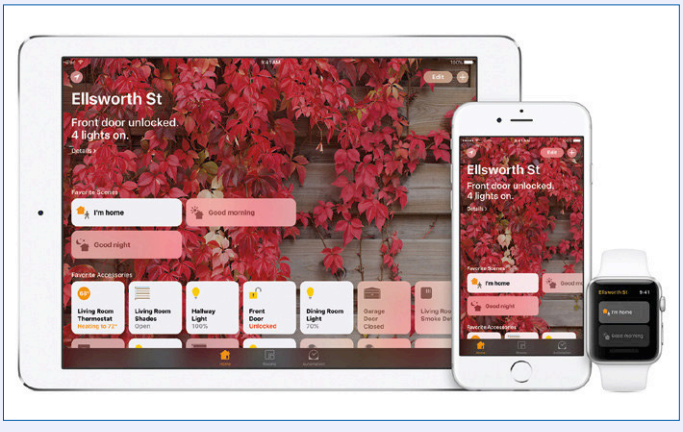
Web:
www.bluetooth.com/marketing-branding/markets/home-automation

Apple HomeKit

Apple has come rather late to the home automation party. Apple uses the Bluetooth LE standard and WLAN for communication. A highlight unique among home automation system suppliers is voice control using Siri.

Chips: Apple-certified chips are available from Texas Instruments, Marvell and Broadcom.

Devices: Apple has not developed any products in-house for HomeKit. Until recently the sale of HomeKit-compatible devices was limited almost exclusively to the North American market, but availability to European consumers has started to grow rapidly over the last few months.
Apple shops currently sell devices from Philips (the Hue lighting system), a multi-room air condition sensor and a water sensor from the German manufacturer Elgato, and a room thermostat from Netatmo. Further products, including door and window contacts, intelligent sockets and heating valves) are available directly from Elgato.



Hub: HomeKit is an interface for controlling compatible devices using iOS apps. This means that an iPhone, iPad or iPod is also required to control a system. Device manufacturers supply their own control apps, but all certified HomeKit devices can be connected to Apple's HomeKit app, which offers Siri integration. Depending on which app is used, groups of functions can be defined, which can then be controlled using a single (Siri) command.
Remote control of HomeKit devices over the Internet is possible using an Apple TV (from third generation onwards) with an Internet connection as the hub.
Apps: The 'Home' app must be downloaded from the App Store to the iOS device and then linked to the HomeKit-compatible device using an installation code. The app is expected to be built into iOS 10. The device can then be operated manually or by Siri voice control. If Internet access via Apple TV is enabled then the device list can be stored in the iCloud, from where it can be synchronized to other Apple devices that have suitable permissions.

Protocol and security: WLAN and Bluetooth Low Energy (operating on 2.4 GHz) using Apple HomeKit Accessory Protocol under iOS from version 8.1.
Development: Developing your own apps is supported using the iOS developer library. The (for the moment free of charge) XCode IDE is required, which is of course restricted to running on Mac computers. The programming language used is called Swift, largely a development of Objective C.
After a few registration procedures the (also free of charge) HomeKit Accessory Simulator can also be downloaded. This can be used later to thoroughly test a completed app. The developer site gives a comprehensive description of how to create a Home Layout, add Accessories (as HomeKit-compatible devices are called), define services and actions and much more besides.



Pros and cons: Limited to iOS; so far only a few devices are available; an Apple TV is required as a hub. The limited range of Bluetooth LE is also a disadvantage in this application. One solution might be a mesh network, which may become possible with the introduction of a new version of the Apple TV.

Web: For consumers: www.apple.com/ios/homekit/
For developers: https://developer.apple.com/homekit/

Magic Eye Indicators

Peculiar Parts, the series





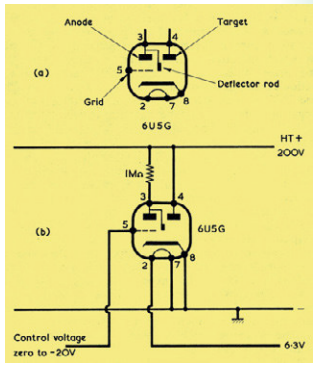


Figure 1. Type 6U5G Magic Eye from various manufacturers, and its connection diagram [1].

By **Neil Gruending** (Canada)

I bet that you don't think twice about what happens every time you tune into a broadcast station using an AM/FM radio. Modern digital radios can tune into a station with great accuracy but analog radios have always depended on their users to tune into a station, which can be difficult to get just right. The usual pattern is to tune analog radios for the loudest signal through the speakers but that doesn't work very well if the radio has some sort of automatic gain control (AGC) or volume control (AVC).

One solution is to use an indicator that helps the user tune the radio like the meters used in ham radios. Analog meters are commonly used for this because they are versatile and can be used to show the signal level (AM and FM), or the fine tuning as a null meter (FM discriminator). But back in the 1930's meters weren't a viable solution (and FM broadcasting nonexistent) because they were too expensive for consumer radios and they were also difficult to make with the required sensitivity. A better solution was needed and the magic eye vacuum tube was created. The original magic eye tube was the Type 6U5G shown in **Figure 1**. It's a triode tube where the anode is connected to the deflector rod electrode, and the cathode is connected to the target electrode. The target is the round top metal piece in the tube, and the deflector rod is a metal rod that goes inside of the target to control the flow of electrons. When the grid voltage is 0 volts no current will flow through the anode so the target and deflector rod will be at the same potential. A negative grid voltage allows an anode current to flow which increases the potential between the target and deflector. The maximum current flow will happen with a grid voltage of about −20 V.

The magic eye effect so many young people marvel at, happens when the target glows or fluoresces from the high voltage applied to it and is controlled by the deflector rod like in **Figure 2**. When the deflector is at its most negative potential relative to the target like in **Figure 2a** the shadow is very narrow because the deflector isn't repelling any electrons. However, as you increase the grid voltage the differential between the target and deflector decreases which causes the deflector to repel more electrons. This results in a larger shadow effect with the largest shadow at a grid voltage of 0V like **Figure 2c**.

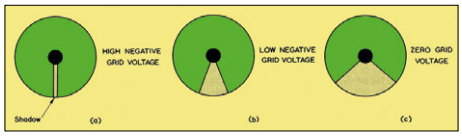


Figure 2. Magic eye examples [1].

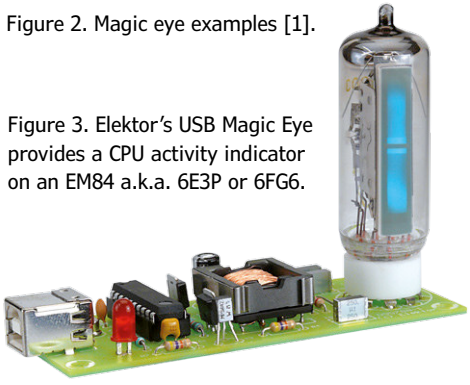


Figure 3. Elektor's USB Magic Eye provides a CPU activity indicator on an EM84 a.k.a. 6E3P or 6FG6.

Magic eyes also came in different styles. For example, the Philips Type EM34 used two deflectors that created two shadows instead of one. Another interesting one is the DM70 which has a straight line shape with a dot at the end instead of circles. Unfortunately, barring Russian imitation tubes which "burn up" rapidly, they are hard to find these days but magic eyes are definitely an interesting indicator.

Finally, Elektor star author Martin Ossmann cheerfully bridged 40 years of electronics technology from vacuum tube to AVR micro with his *USB Magic Eye* project published in the January 2010 edition (**Figure 3**) [2]. And there are people breaking the original glass envelopes of extinguished EM34s and EM4s to install *ersatz* circuitry like LEDs and PICs. In most cases the result is slightly lacking in magic.

(150803)

Web Link

[1] www.r-type.org/articles/art-137.htm

[2] www.elektormagazine.com/090788

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

CIRCUITMAKER Tips & Tricks (3)

By **Neil Gruending** (Canada)

Let's make a PCB!

So far we've seen how to start a project and added a schematic to it. Now let's make a board and generate some Gerber files.

Creating a PCB

The first step to add a PCB to your project by using the Home → Add new PCB command. You will be prompted for a file name and then a default PCB file will be created for you with a 100 mm × 100 mm square PCB board like in **Figure 1**.
I doubt that will be the size you need so you can change the board size directly using the Home → Board → Board Shape → Redefine Board Shape or Edit Board Shape. But the recommended method is to actually draw the board shape on the Keepout layer using the line and arc tools. This way the keepout lines will also apply to the component placement and copper routing rules. Once you've finished drawing the board shape you can select all of the items on the keepout layer using Home → Clipboard → Select All on Layer and then going into the board shape menu and selecting Define From Selected Objects.

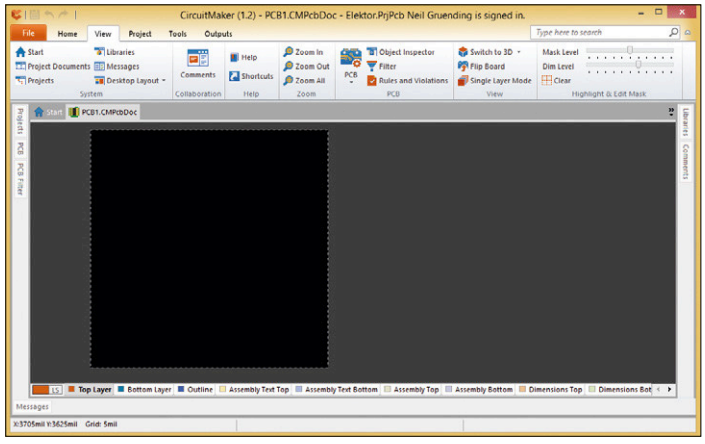


Figure 1. Default 100 mm × 100 mm PCB.

Once the board outline is finished it's time to transfer the components from the schematic to the PCB by going into Home → Project → Update PCB Document. CircuitMaker will then automatically compile all of the schematic documents to resolve the design hierarchy and to generate all of the connectivity information. Then the Comparator tool runs to identify all of the differences between the schematic and PCB documents. Those differences are then used to generate an engineering change order (ECO) as in **Figure 2** which you can validate and apply (execute) the listed changes to the PCB. You can also prevent changes from being applied by clearing the checkmark in the Enable column.
Afterwards all of the schematic components and the netlist will be updated in the PCB. As you place the components on the board don't forget about the keyboard shortcuts mentioned last time because they can be a real time saver. For example the easiest way to rotate a component is to use the space bar instead of using the mouse. Once everything has been placed then it's time to start routing the board.

Routing a board

CircuitMaker has several interactive routing tools available for routing a board. The Interactive Routing tool lets you route a single trace while CircuitMaker makes sure that the trace avoids other traces and obstacles. The Differential Pair tool is used to route both sides of a differential pair at the same time and the Interactive Multi-Routing tool is used to route a group of traces like a data bus all together.

As you route the trace you can cycle through several options using the Shift+R keyboard shortcut:

- Ignore: Allows you to place traces anywhere on the board.
- Stop at first obstacle: This will stop the trace when an obstacle is encountered.
- Push: Tries to push traces out of the way of the trace currently being routed like in **Figure 3**.
- Autoroute on current Layer: Adds a simple autorouter to the interactive routing tool which chooses the best routing option when an obstacle is discovered.

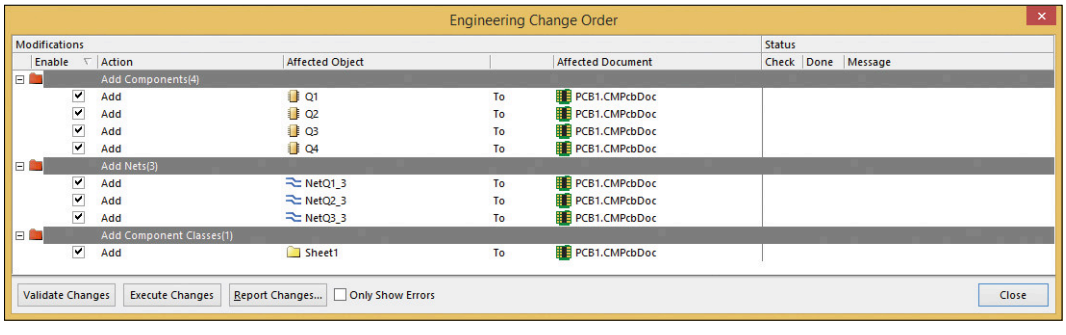


Figure 2. Engineering change order window.

CircuitMaker's interactive router constantly checks the trace that your routing against the design rules to make sure that you didn't accidentally violate them. Every new project gets a default set of design rules that you can edit with the Design Rules Editor (Home → Design Rules → Define Design Rules) which is shown in **Figure 4** editing the default clearance rule. For example, you can tell CircuitMaker to normally use a 0.25-mm trace clearance and to use a clearance of 0.15 mm near certain components. The rules system also lets you group components and nets into classes that can also be used in the design rules.

Generating Gerber and STEP outputs

Once the board is routed it's time to generate the files needed for manufacturing. It depends on the board manufacturer but generally the Gerber file format is the most widely supported. Gerber files have a pretty simple format where one file is generated for each layer on the PCB which contains all of the photoplot and apertures (tools) used.

But before we generate any Gerbers it's always a good idea to run a design rule check using Outputs → Reports → Design Rule Check command. You want to do this because the online (real time) design rule check only checks a subset of all of the design rules to minimize processing time whereas Batch mode, i.e. the mode that runs when you click on the Run Design Rule Check button is much more thorough. Once the check is finished, CircuitMaker puts all of the errors into a HTML report file and in the Messages pane. Clicking on the errors will take you to that point in the PCB while giving a description for the error.

Once you've corrected any design errors it's time to generate the Gerber files using the Output → Fabrication → Gerber command. The Gerber Setup window will appear where you can specify the units, precision and the layers that should be converted. You can also configure the drill drawing and the apertures used. Note that by default CircuitMaker will generate RS274X Gerbers that incorporate the apertures in each Gerber file instead of having them in a separate file. Clicking the Ok button will generate all of the Gerber files which CircuitMaker puts into a ZIP file that you can save on your computer. Then all that's left is to double check the Gerbers using a Gerber viewer.

You can also generate a STEP model for your board by using the Outputs → Export → Step command. The generated model will also include all of the relevant component STEP models from the CircuitMaker library which can then be used in a MCAD program to work on the mechanical aspects of a design like the enclosure design. It's also possible to view your board in 3D using the integrated 3D viewer like in **Figure 5** by using the View → View → Switch to 3D command.

CU on CircuitMaker

This article series has just barely scratched the surface on what CircuitMaker can do. I encourage you to look at the online documentation [1], especially the PCB design walkthrough [2] which goes into the PCB design process in a lot more detail. Have fun! ⏮

(160017)

Web Links

[1] http://documentation.circuitmaker.com/

[2] http://documentation.circuitmaker.com/display/CMAK/From+Idea+to+Manufacture+-+Driving+a+PCB+Design+through+CircuitMaker

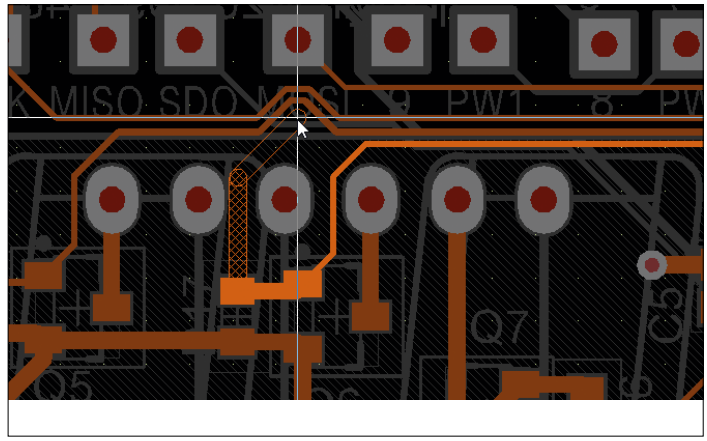


Figure 3. Interactive routing tool.

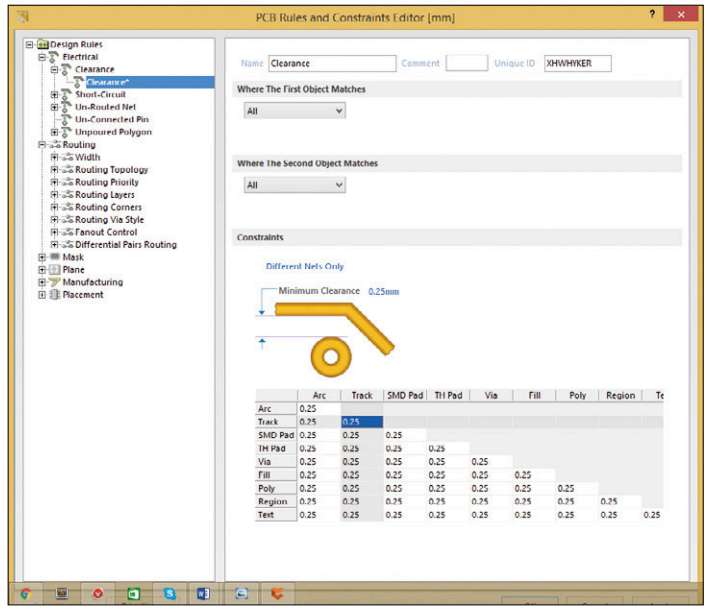


Figure 4. PCB design rules.

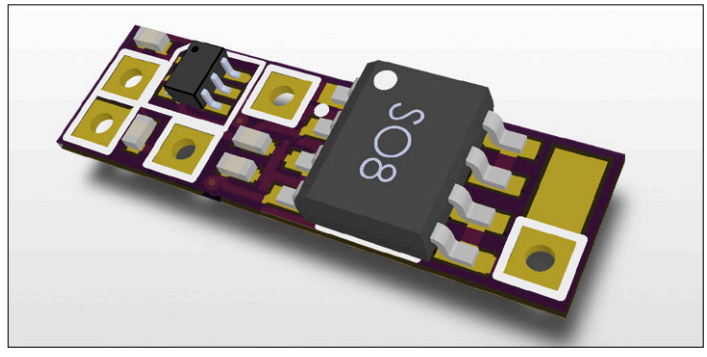
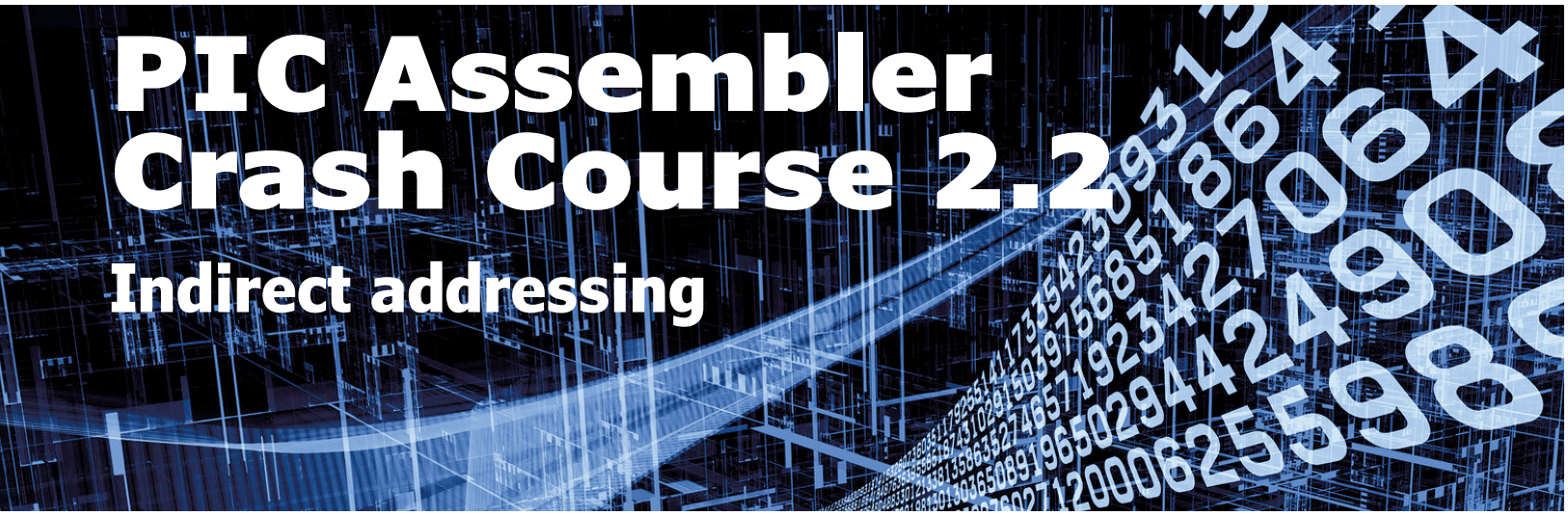


Figure 5. Integrated 3D board view.

PIC Assembler Crash Course 2.2

Indirect addressing



In the previous installment of our second assembler crash course series we looked at how to drive a multiplexed LED display. The advanced addressing modes of a modern microcontroller make the programming of this kind of task much easier. In this second part we will find out about indirect addressing.

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

The modern PIC1xF18xx family of PIC microcontrollers offers a very handy feature in the form of indirect addressing. The big advantage of indirect addressing is that it allows linear access to the full address space and so avoids the need to deal with paged memory areas, which can be inconvenient. The result is that the microcontroller is as easy to use as the 'genuine' eight-bit microprocessors from the 1980s such as the Z80.

The INDFx registers

In the previous installment we described direct addressing with the assistance of the BSR register. All microcontrollers in the PIC1xF18xx series contain two further special registers, called INDF0 and INDF1, which allow linear access to RAM. It gets better: these registers allow linear access to the program memory as well. The whole memory area, RAM and flash (but not the data EEPROM) are made to appear to the microcontroller as a unified 64 kbyte memory block.

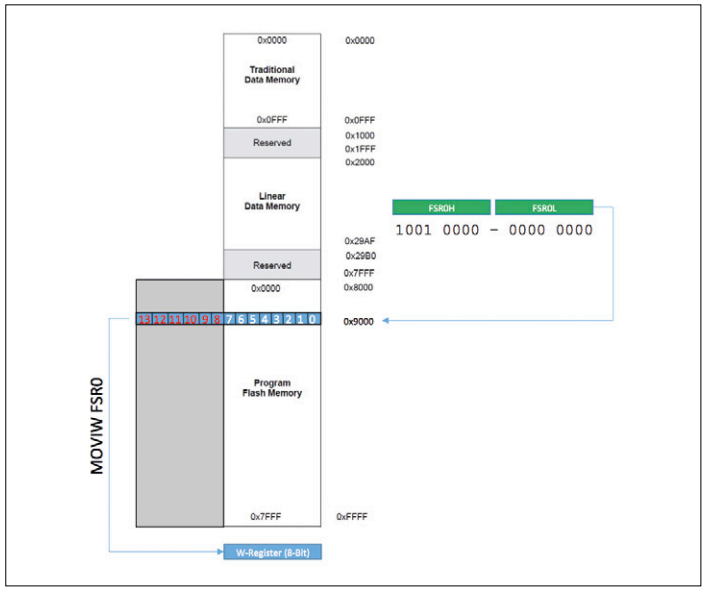


Figure 1. Memory map of the PIC12F18xx series of microcontrollers.

To address 64 kbytes we need addresses 16 bits long. These addresses are held in the register pairs FSR0 (composed of FSR0H and FSR0L) and FSR1 (FSR1H and FSR1L). We have already seen these registers as part of the core register set. What we now need for indirect addressing are a couple of new assembler instructions: in describing these we will see what purposes the INDF0 and INDF1 registers can be put to.

The RAM appears twice in the 64 kbyte address space. First, in the region from 0000h to 0FFFh, it appears as ordinary working memory, and access to these addresses via FSRx works in much the same way as direct addressing using BSR. The RAM then reappears in the region from 2000h to 29AFh, but in this case the RAM is contiguous. This is advantageous in that, for example, it allows the simple implementation of a large array or table. But why does the address range end at 29AFh and not at 2FFFh? The reason is that this memory area contains copies of the individual 80 byte blocks in each RAM bank. Thus bank 0, which corresponds to addresses 020h to 06Fh, is mirrored from 2000h to 204Fh, bank 1 is mirrored from address 2050h to 209Fh, and so on. The last bank that is mirrored in this region (which is bank 30) therefore ends at address 29AFh.

Access to the flash program memory is done using addresses from 8000h to FFFFh. Clearly not all the address space in this region is actually occupied: for example, the 8 kwords of flash in the PIC16F1829 occupy the range from 8000h to 9FFFh, whereas the 2 kwords of flash in the PIC12F1822 run from 8000h only as far as 87FFh. The rest of the region is unused: it can be accessed, but there is little point in doing so.

As is typical in PIC microcontrollers, the RAM array is eight bits wide whereas the program memory is organized into 14-bit words. You might wonder how these 14-bit values are packed into the linear address space. The answer is remarkably simple: only the byte comprising the eight least significant bits (LSBs)

of the word is directly accessible. Before we look at how the MOVIW instruction is used to access memory, it is worth taking a quick look at the memory map in **Figure 1**. Here we can see that although the program memory is 14 bits wide, we do not have access to the most significant six bits (shown shaded in gray). The MOVIW instruction copies the contents of the memory location whose address is in the FSR0 register pair to the W register. Only the eight least significant bits are copied.

MOVIW

This instruction transfers the contents of the INDFn register to the W register (MOVe Indfn to W). Here INDFn can be either INDF0 or INDF1. But things are not quite what they seem: the INDFn registers do not exist. An access to INDF0 instead accesses memory at the address specified in FSR0, and likewise INDF1 accesses memory at the address specified in FSR1. There are two variants of the instruction: in one variant the pointer is updated (incremented or decremented) and in the other variant it is not. Here we look at the first variant, which is often used to access arrays.

The first variant has in turn four subtypes as listed in **Table 1**. These differ in whether the pointer is updated before or after the data access, and in whether the pointer is incremented or decremented. Also, the value of 'n' can be '0' or '1', specifying whether FSR0 or FSR1 is to be used.

As well as this MOVIW instruction we also have its counterpart, the MOVWI instruction. This instruction writes the contents of the W register to the memory location with the specified address. Of course, this only works in a region of the address space where there is RAM. Other techniques are needed if you wish to alter the contents of the flash memory. Care should be taken when doing this, as there are pitfalls associated with writing self-modifying code.

Displaying program memory

From the previous installment we already have hardware we can use for showing information in the form of an LED display. We will now use this to display the contents of the flash memory starting at address 1000h. To do this we simply need to make some changes to the main loop in the microcontroller's software; we will re-use the display driver code from the previous installment but that too needs a little modification. In the main routine we replace the counter with code to read data from the program memory.

The idea is that inside a loop we read data from the program memory and then pass it to the display. The code will read data byte by byte starting at address 1000h. Each byte that is read is displayed for a short period before we proceed to the next in sequence. The loop finishes when we read a value of FFh from the memory, and then we start again from address 1000h.

Since we are looking to display byte values, we would like the display to be capable of displaying hexadecimal values rather than just decimal digits: that way we can display a byte in hex format using just two characters. Here we will use the two leftmost digits of the display, LD0 and LD1. LD2 will be left blank and in the fourth position LD3 will permanently show an 'h'. The decimal value will then be displayed as shown in **Figure 2** as '7A h'.

Table 1.

Syntax	Meaning
`moviw ++FSRn`	**Preincrement** Increment before read access
`moviw --FSRn`	**Predecrement** Decrement before read access
`moviw FSRn++`	**Postincrement** Increment after read access
`moviw FSRn--`	**Postdecrement** Decrement after read access

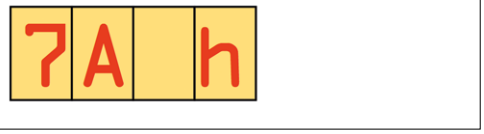


Figure 2. Showing a byte in hexadecimal form on the LED display from the first installment of this series.

Modifying the display driver

As mentioned above, we want to make the display able to show hexadecimal characters. The code in the first installment of this series could display only the digits from '0' to '9' and the '-' character. To those we need to add the letters 'A' to 'F', the blank character and the lower-case 'h', which requires modifying the display driver. We will then be able to display any value from 0 to 255 neatly in hexadecimal.

The character generation in the routine wm_char_set therefore needs to be expanded a little. The differences between the old decimal version in **Listing 1** and the new hexadecimal version in **Listing 2** have been kept within reasonable limits. This example shows how easy it is to extend the display driver code when required. In this case the addition is just an extra line of assembler to define each new character.

The main code

We can now look at the main part of the program, which consists of the following three parts.

- Reading data from the flash memory using the subroutine `f_get_data`.
- Converting the byte into two hexadecimal digits using the subroutine calc_hex_disp.
- The overall infinite loop structure.

Listing 1.

```
;-------------------------
;old decimal character set
;-------------------------
wm_char_set brw
    retlw B'00111111' ;00h -> 0
    retlw B'00000110' ;01h -> 1
    retlw B'01011011' ;02h -> 2
    retlw B'01001111' ;03h -> 3
    retlw B'01100110' ;04h -> 4
    retlw B'01101101' ;05h -> 5
    retlw B'01111101' ;06h -> 6
    retlw B'00100111' ;07h -> 7
    retlw B'01111111' ;08h -> 8
    retlw B'01101111' ;09h -> 9
    retlw B'01000000' ;0Ah -> minus
```

Reading the flash memory

This part of the program demonstrates the use of indirect addressing. The complete set of constant declarations for the data in the flash memory and the accompanying code to read it is very straightforward, as **Listing 3** shows.

The data to be read must be placed in program memory starting from address 1000h. This is achieved using the directive 'ORG 1000' (see step 3 in the listing). We also label the start of the data area as 'c_data1'. In principle we could also fill other areas of the flash memory, with two restrictions: first, the memory area must physically exist within the flash memory provided by the particular microcontroller device being used; and second, the declaration must not 'accidentally' tread on the toes of the actual program code. The microcontroller we are using here, the PIC16F1829, has 8 kwords of program memory, and so the highest available flash memory address is 1FFFh. If by mistake we attempt to use address 2000h, address 0000h will be accessed instead, probably with unfortunate results.

If we declare the storage area for the data using the ORG directive at too low an address, an error will be reported when we try to assemble the code. The message will depend on where in program memory our declaration falls. For each erroneously overwritten byte we receive exactly two error messages, as in the following example.

```
Error[118]   C:\...\05_TEST.ASM 339 :
   Overwriting previous address contents (0051)
Error[118]   C:\...\05_TEST.ASM 339 :
   Overwriting previous address contents (0051)
```

Why exactly two messages per byte? This is because the program memory words are 14 bits long, and hence a declaration H'AB' is treated identically to a declaration H'00AB'. The latter type of declaration overwrites two bytes, and we receive one error message for each. Note that the top half of the word can only take on values from 00h to 3Fh; so, since using indirect addressing we cannot access the top six bits of the program memory word, we can take the opportunity to use the short form of the declaration.

It would also be possible to dispense with the 'ORG 1000' declaration entirely. In that case the data bytes would be placed in flash memory immediately after the end of the subroutine f_get_data. Sometimes, however, it is important to know exactly where the data are placed, and without the 'ORG' directive the position of the data will move when the code changes.

The routine f_get_data (see step 2 in the listing) reads one byte from the flash memory: to do this requires just four lines of code. As we described above the MOVIW instruction copies the value into the W register; we then store the value in the temporary register v_value for later use, and the job is done. We use the post-increment version of the instruction here, which means that the pointer will automatically be incremented after the read operation. On the next call to the subroutine we will therefore automatically read the next byte.

Now a word on initializing the pointer, which in this case is the register pair FSR1. The registers need to be set to the correct address so that we start reading from the right location. This is done in the subroutine f_get_add (see step 1 in the listing). The 'LOW' and 'HIGH' directives are used to split the address into two parts so that they can be written to the correct halves of the register pair. The line

```
movlw   LOW c_data1
```

first loads the less-significant byte of the address c_data1 into the W register. In this case this is the value 00h (the second half of 1000h). Similarly, one might expect the line

```
movlw   HIGH c_data1
```

Listing 2.

```
;------------------------------
;new hexadecimal character set
;------------------------------
wm_char_set brw
   retlw B'00111111' ;00h -> 0
   retlw B'00000110' ;01h -> 1
   retlw B'01011011' ;02h -> 2
   retlw B'01001111' ;03h -> 3
   retlw B'01100110' ;04h -> 4
   retlw B'01101101' ;05h -> 5
   retlw B'01111101' ;06h -> 6
   retlw B'00100111' ;07h -> 7
   retlw B'01111111' ;08h -> 8
   retlw B'01101111' ;09h -> 9
   retlw B'01110111' ;0Ah -> A
   retlw B'01111100' ;0Bh -> b
   retlw B'00111001' ;0Ch -> C
   retlw B'01011110' ;0Dh -> d
   retlw B'01111001' ;0Eh -> E
   retlw B'01110001' ;0Fh -> F
   retlw B'01000000' ;10h -> minus
   retlw B'00000000' ;11h -> space
   retlw B'01110100' ;12h -> h
```

Listing 3.

```
;---------------------
;declare and read data
;---------------------
;step 1
f_get_add   clrf  BSR
   movlw LOW c_data1
   movwf FSR1L
   movlw HIGH c_data1
   movwf FSR1H
   return
;
;step 2
f_get_data  clrf  BSR
   moviw FSR1++
   movwf v_value
   return
;
;step 3
   ORG 1000
c_data1  da H'01', H'02', H'03', H'04'
   da H'AB', H'CD', H'EF'
   da H'FF'
```

to load the more-significant byte 10h (the first half of 1000h) into the W register. Now this is not quite correct, as when using indirect addressing the base address of the flash memory is 8000h. Flash memory location 0000h is at 8000h in the FSR address space, and so the data at c_data1 starts at 9000h. For this reason the assembler will ensure that this line of code will load the value 90h into the W register rather than 10h.

One byte in two nibbles

The final routine has nothing to do with indirect addressing, but is needed in order to show a byte in hexadecimal on the display. To understand this code we first need to look at the LSRF (logical shift right) instruction which we have not previously seen. The instruction is one of the 'byte-oriented file register operations' and simply shifts all the bits in a register to the right. The syntax is

```
lsrf  f
```

where 'f' is the register to be shifted. The bit at the far right of the register (the LSB) is shuffled off into the carry flag in the status register; the previous value of the carry flag is overwritten. Into the left-hand end of the register comes a zero bit as the new MSB (most significant bit). The process is illustrated in **Figure 3**. If the register initially contains the value 7Ah then the result of executing the LSRF instruction will be to leave 3Dh in the register (see **Figure 4**). For completeness we can also mention the LSLF instruction, which is similar but which shifts in the opposite direction.

The code to divide a byte into its more-significant and less-significant nibbles (1 nibble = 4 bits = half a byte) is shown in **Listing 4**. First we load the byte that we have read into the W register (step 1) and then the AND instruction with an argument of 00001111b clears the four MSBs to zero. The result of this is written to v_lsb. The original byte is then loaded into the W register again (step 2) and a series of four LSRF instructions is used to remove the four LSBs. The four MSBs of the original value are now in the four LSBs of the result. **Figure 5** shows how this code fragment works. The variables v_lsb and v_msb end up holding the values that we require, ready to show the byte as a hexadecimal value on the display.

Coming up

Now that we have succeeded in reading some bytes and converting them to hexadecimal values for the display we come to the end of the second part of this second assembler crash course. All the code is available for download from the web pages accompanying this article [1]. In the next installment we will look at serial communication using the I²C and SPI buses, which are particularly important when using peripheral chips in conjunction with microcontrollers. Then we will show how simple it is to use a serial interface to control a port expander, and demonstrate using the LED display to make a clock or voltage and current meter, which will require the use of an external chip and a little code.

If you have any questions on the code in these examples please do not hesitate to contact the author at miroslav.cina@t-online.de.

(160018)



Figure 3. The effect of the LSRF bit shifting instruction.

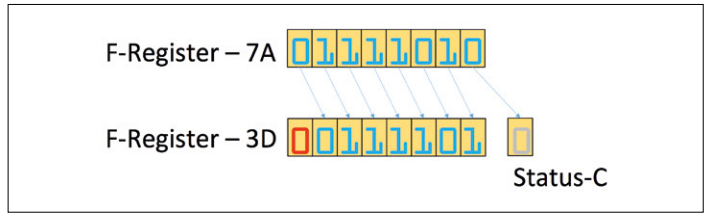


Figure 4. The LSRF instruction shifts the contents of a register to the right. The old LSB ends up in the status register and the register is filled with a zero from the left.

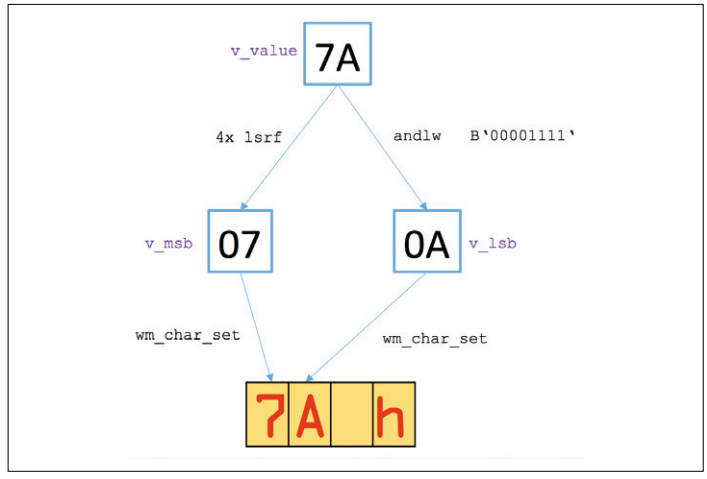


Figure 5. Splitting a byte into two nibbles so that it can be displayed in hexadecimal.

Web Link

[1] Download: www.elektormagazine.com/160018

Listing 4.

```
;-----------------
;byte in 2 nibbles
;-----------------
;step 1
calc_hex_disp
   movf  v_value,0
   andlw B'00001111'
   movwf v_lsb

;step 2
   movf  v_value,0
   movwf v_msb
   lsrf  v_msb,1
   lsrf  v_msb,1
   lsrf  v_msb,1
   lsrf  v_msb,1
   return
```

LED Control Tabletized

using Flowcode and E-blocks

By **Bert van Dam** (Netherlands)

In this article we'll show you how to turn a strip of LEDs on and off via your tablet (or PC) with the help of Flowcode 6 and several E-blocks modules. We will also display the state of a switch on the tablet.

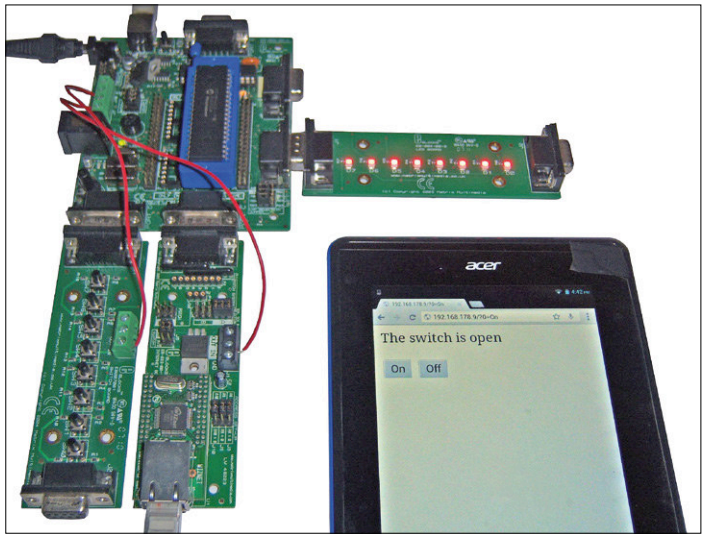


Figure 1. Controlling the LEDs.

E-blocks are highly useful electronic modules, which can be used to put a project together quickly and easily. In combination with Flowcode you can program these graphically. You will need the following to carry out the experiments in this article: Flowcode V6 for PIC, a PIC Multi-Programmer (EB006) with a 18F4620 PIC microcontroller, an LED Board (EB004), a Switch Board (EB007), an Internet Board (EB023) and an Ethernet cable. With the help of these E-blocks we're going to create a local Internet— an intranet—in our home.

Setting up the intranet

We start by connecting the EB023 Internet Board to your router using an Ethernet cable. Most devices automatically request an IP address from the router, but for the EB023 we have to do this manually. This is called a static IP address, since it remains fixed. The simplest method is to log on to your router to see which static addresses are available for you to grab. You must always keep the first three numbers of the address the same (in my case these are 192.168.178) and you can select something for the last number. The minimum is usually 2 (since 1 is used by the router itself), and the maximum depends on the router. In my case, the maximum is configurable and has been set to 9. It is advisable to make a list when you start using several static IP addresses, so you don't accidentally allocate the same address twice. An example of this is shown in **Table 1**.

You don't need to do anything special with the tablet, since this will connect to the router in the usual way via Wi-Fi. If you don't have a tablet, you could also use a smartphone or a PC.

The build

The hardware used for this project is shown in **Figure 1**. Connecting it all up is a very quick and simple task, since we're using E-blocks.

- **Programmer**
  EB006. Switches on XTAL and Fast, LVP jumper on I/O port, J29 on PSU, J12-14 on USB, use an external power supply. Note that we're using an 18F4620 PIC instead of the standard PIC, since this doesn't have enough memory.
- **Port B**
  EB004 LED board.
- **Port C**
  EB023 Internet board. Jumpers on A, +5 V, address at 1-1-1, network cable to the router. Power supply cable to the programmer.
- **Port D**
  EB007 switch board. Power supply cable to the programmer.

Operation of the software

The software is written (drawn, really) in Flowcode 6. The EB023 Internet Board is connected to the microcontroller via an I^2C bus. In the program we use a webserver component where all the settings for the EB023 are configured (**Figure 3**). The I^2C address jumper setting of 1-1-1 corresponds to High-High-High. The IP settings have to be set for the same address as configured in the router (see **Table 2**). For the MAC address you should use the default one that is already in the component.

Table 1. Static IP addresses in the router of the author.

IP address	Device
192.168.178.9	Flowcode EB023
192.168.178.8	ARM mbed
192.168.178.7	Raspberry Pi on Wi-Fi

Table 2. IP addresses in the webserver component.

IP address	Item
192.168.178.1	gateway (the router)
255.255.255.0	subnet mask
192.168.178.9	EB023

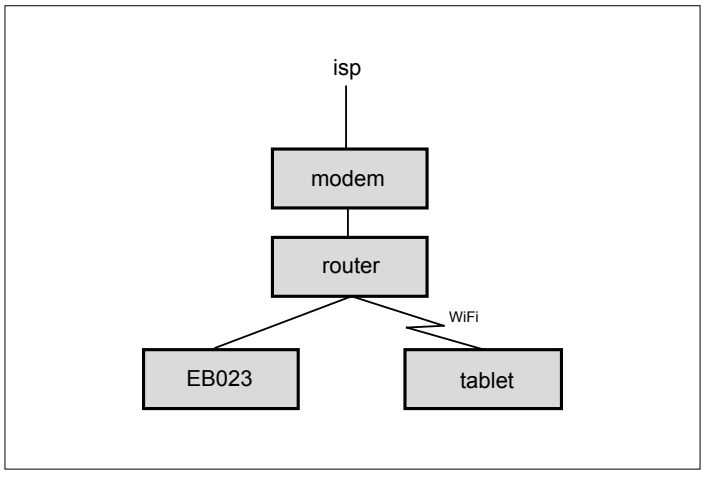


Figure 2. The EB023 module and the tablet both communicate with the router, one via a cable, the other wirelessly.

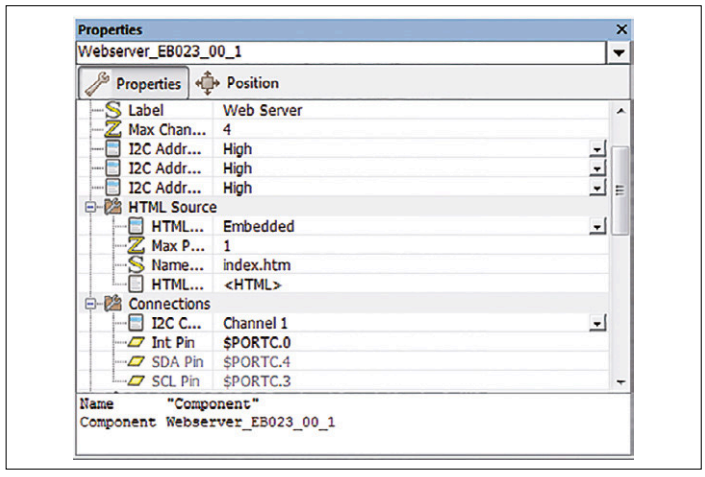


Figure 3. The I²C bus settings of the webserver component.

For the next step, you have to enter the page that appears on the tablet when it has made a connection to the EB023. We want to be able to turn the LEDs on and off using two buttons shown on the tablet screen. We also want to display the state of a switch on the screen.

When the user browses to the IP address of the EB023, the webserver component sends a page containing two buttons. When we use one of these buttons to make the page call itself (using "_self"), a suffix with the name and value of that button is added to the IP address. The first button has the name "0" and a value of "On". The suffix will then be "?0=On". The full address sent to the webserver component will therefore be "192.168.178.9/?0=On". When we ask the webserver-component for GetInValue(0) in the Flowcode program, the result will be the value (as a string) of the variable at index zero on the address line, which in this case is "On". This value is then used to turn on the LEDs.

Since we're not doing anything else with the buttons on the

Advertisement

USB Add USB to your next project. It's easier than you might think!

DLP-USB1232H: USB 2.0 UART/FIFO

HIGH-SPEED 480Mb/s



- Multipurpose: 7 interfaces
- Royalty-free, robust USB drivers
- No in-depth knowledge of USB required
- Standard 18-pin DIP interface; 0.6x1.26-inch footprint

DLP-IO8-G
8-Channel Data Acquisition



Only $29.95!

- 8 I/Os: Digital I/O
  Analog In
  Temperature
- USB Port Powered
- Single-Byte Commands

DLP-IOR4
4-Channel Relay Cable

DLP-TH1b
Temp/Humidity Cable

DLP-RFID1
HF RFID Reader/Writer

DLP-FPGA
USB-to-Xilinx FPGA Module

DLP Design

www.dlpdesign.com

The Easiest Way to Design Custom
Front Panels & Enclosures

Free Front Panel Designer



You design it
to your specifications using our FREE CAD software, Front Panel Designer

We machine it
and ship to you a professionally finished product, no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

FRONT PANEL EXPRESS

FrontPanelExpress.com

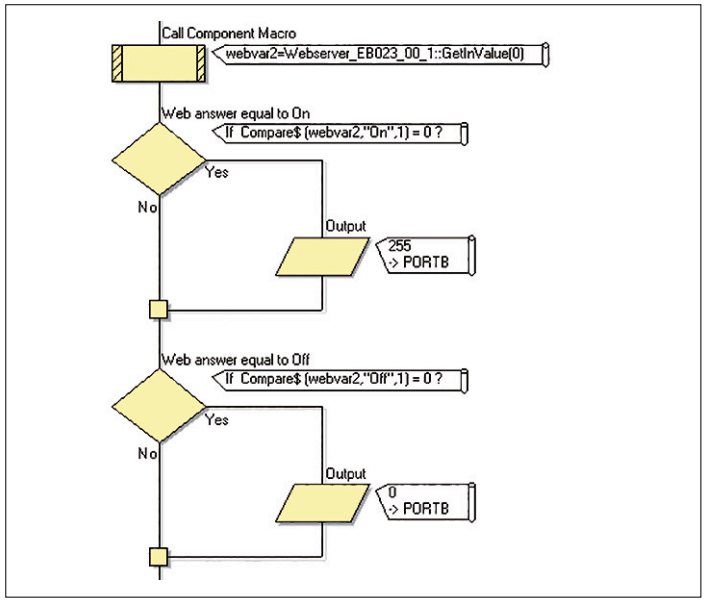


Figure 4. Flowcode is used to deal with the instructions from the tablet.

web page, we can also give the name “0” to the second button, but this time with the value “Off”, which causes the LEDs to be turned off.
Since text is usually very small on a tablet, we’ve increased the size of the buttons (using “width:100px;height:50px”) and selected a larger text size (using “font-size: 18pt;”).
When we want to send some information to the tablet (which in this case is the state of the switch), we put an index at that position in the HTML code, preceded by a percent sign. In this case we want to show the position of the switch, so we use the text “ The switch is %1”, where the %1 will be replaced with the required value later on by a macro.

```
<HTML>
<FORM>
<P></P>
<P STYLE=”font-size: 18pt;”>The switch is %1</P>
<P><INPUT NAME=”0” TYPE=”submit” STYLE=”width:1
   00px;height:50px;font-size: 18pt” VALUE=”On”
   onClick=”window.open(‘’,’_self’)”>
<INPUT NAME=”0” TYPE=”submit” STYLE=”width:100
   px;height:50px;font-size: 18pt” VALUE=”Off”
   onClick=”window.open(‘’,’_self’)”>
</P>
</FORM>
</HTML>
```

You enter this code into HTML source in the section for HTML (Figure 3). To save space, we have ignored some of the standard HTML conventions (we don’t have a header or body), but modern browsers can cope with this.

The operation of the system is very simple. At the start of the program a *socket* is opened. You could compare a socket with a telephone. Both parties need a telephone to communicate with each other, and those phones have to be connected together. The same applies to sockets. The EB023 opens a socket and listens. When the tablet makes a connection, the socket of the tablet is connected to the socket of the EB023 and communications can take place. The macro CheckSocketActivity in the main loop of the program then checks if something has happened on this socket. This is then dealt with automatically. The only things left to do are to use the GetInValue(0) macro to check if new information has been received from the tablet (with an index of 0) and then use SetOutValue(1,state) to insert the actual state of the switch instead of the %1 sign.

Operating the system

Wait for the program to start and give the EB023 and your router a few seconds to establish a connection. Then use a browser on your tablet to navigate to the address of the EB023, which in my case is 192.168.178.9. The Internet page will now be loaded. Press one of the buttons to turn the LEDS on or off.
The Internet page also displays the state of switch SW1 on the EB007 switch board. Press the switch and touch one of the buttons on your tablet to refresh the page, so the new state of the switch will be shown.
You can also configure the router so that the EB023 can be accessed via the Internet. In that case you’ll be able to control the LEDs from anywhere in the world.
When you connect a relay instead of the LED Board, you can operate all kinds of devices via your tablet. Keep in mind that the E-blocks relay board is not suitable for use with AC line voltage.

Hardware/software

The required hardware and the source code for the Flowcode program can be found on the Elektor Store [1] and Elektor Magazine [2] websites respectively.
Other similar projects can be found in the book “Flowcode V6 — 30 projects for PIC microcontrollers”. ⏮

(140351)

Web Links

[1] www.elektor.com/development/eblocks and www.elektor.com/development/flowcode-1-user

[2] www.elektor-magazine.com/140351

The Author

Bert van Dam is a freelance author of books, course materials and articles about PIC and ARM microcontrollers, Arduino, Raspberry Pi, PC, Artificial Intelligence and programming languages JAL, C, assembler, Python and Flowcode.



Mercury Relays

Peculiar Parts, the series

By **David Ashton** (Australia)

In days gone by, mercury was used as the contact medium in some relays. Most old-timers like myself will remember the mercury-wetted relays which were common in telegraphy equipment. Fairly normal relay contacts were coated in a thin film of liquid mercury which served a few purposes: it reduced the contact resistance by making the effective contact area larger, and it also eliminated contact bounce to a large extent because once the initial contact was made, the surface tension of the mercury would maintain the connection as long as the contacts did not bounce too far apart, and it meant that the relays could be small and fast — teleprinters worked at up to 100 Baud (= Hz in this case). (You would periodically take the relays and shake them to recoat the contacts with the mercury, and these relays had to be mounted a certain way to prevent the excess mercury causing short circuits. These relays were usually labeled accordingly (like the one in **Figure 1**) and often had strict instructions not to open them due to the toxic nature of the mercury. Our esteemed Editor Jan has a 1959 Tektronix oscilloscope plug-in test unit Type P which uses a mercury wetted relay to generate pulses with 4-ns rise times...but I won't steal his thunder, you can read about that in a forthcoming *Retronics* article.

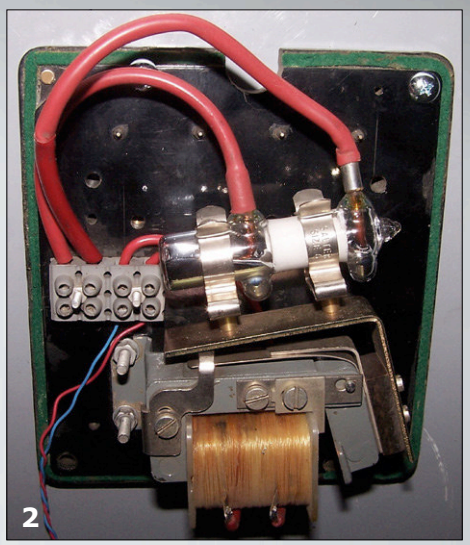
I recently found this old beauty pictured in **Figure 2**. It consists of a fairly normal relay mechanism, with a 240-VAC coil, but instead of operating contacts, the armature of the relay tilts a mercury switch. The switch itself consists of a shaped glass tube containing the mercury and a couple of contacts, which are about 2 mm in diameter and 6 mm long. When the relay is operated, it is tilted and the mercury flows to make a connection between the two contacts (**Figure 3**). When the relay is released, the mercury flows back into the reservoir on the left, and there is no longer enough to cover both contacts and make a connection. The switch and the relay coil are connected to terminals with flexible multi-strand wires. The neck of the glass tube appears to have a smaller tube of what looks like ceramic material bonded to it inside to reduce its diameter to ensure the mercury only makes a connection when the relay is operated. There is a fair bit of mercury in this relay — I would estimate 1 cc or more.
I have read about other mercury relays which worked by magnetically attracting a slug of iron either into or out of a pool of mercury to make the level of the mercury rise or fall, and hence make or break the contacts.

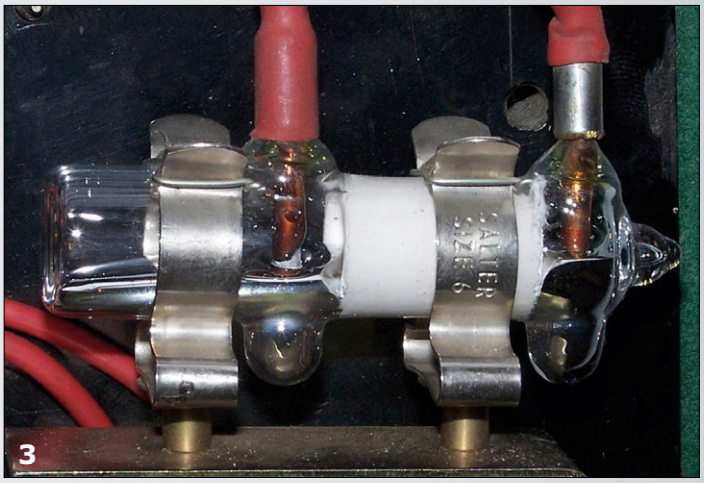
Mercury relays and switches are not often seen these days. I remember as a kid playing with the mercury from a broken thermometer, and todays CFL Lamps still have mercury in them, but that's another story… Mercury was also used widely in tilt switches (which open of close a contact if they are tilted away from their normal position) and some of these are still made today, well-sealed. My mercury relay above reminded me what a fascinating substance this silvery liquid metal is, for all the bad press it seems to get.

(150672)



1


2


3

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

Eagle Tips & Tricks (5)

The XML file format



By Neil Gruending (Canada)

In this installment,
let's explore Eagle's XML file format

A unique Eagle feature is that it has been using an XML format for all of your design files since version 6. The nice thing about XML files is that they are human readable which means that you can view and edit them should you need to.

XML file structure

Extensible markup language (XML) files are text files that follow the formatting rules that are defined by the World Wide Web Consortium (W3C) in the XML 1.0 specification [1]. It's very detailed but the basic rules are straightforward and is a superset of HTML if you've ever worked with that.
For our purposes we are interested with how the tags and attributes in the file are used to create elements to store information. Tags always start with a `<` which is then followed by the tag name. Then comes any attributes which are variables that get set in the form `name="value"`. The end of the tag is always a `>`. Tags can have one of the following forms:

- Start tags like `<tag>`
- End tags like `</tag>`
- Empty tags like `<tag />`

An element is all of the information between matching start and end tags or contained within an empty tag. Any attributes in the start tag or empty tag are also part of the element. Start and end tags are used when you want to nest elements but if you only need to set the attributes for an element then an empty tag is appropriate.
This makes more sense if you look at an example like **Listing 1** which is a layer stack definition from an Eagle board file. The `<layers>` start tag and the `</layers>` end tag group all of the layer elements together. Each layer then uses an empty tag to set all of the attributes for that layer.
XML files also allow comments which start with `<!--` and end with a `-->`. They aren't part of Eagle's saved files but are common in other XML files as we will see later.

Eagle XML format

The Eagle XML file format is defined in a file called eagle.dtd in the Eagle documentation directory. You can open it in a text editor from the Eagle Control Panel but I would suggest using an editor that will color code the file for you. An XML DTD or Document Type Definition file is a way to define an XML file format using XML and can also be used to verify whether a XML file is formatted correctly.
DTD files can be very complex but fortunately for us the Eagle team has done a really good job at making everything understandable by humans by just reading the file. The first part of the file is the Entity section where all of the allowable data types and their values are defined. **Listing 2** shows a portion of the entity section.
The first line defines an entity called Int which be assigned character data or CDATA just like the Real and String types. A

Listing 1. Eagle XML example

```
<layers>
<layer number="1" name="Top" color="4" fill="1" visible="no" active="no"/>
<layer number="16" name="Bottom" color="1" fill="1" visible="no" active="no"/>
<layer number="97" name="Info" color="7" fill="1" visible="yes" active="yes"/>
<layer number="98" name="Guide" color="6" fill="1" visible="yes" active="yes"/>
</layers>
```

Listing 2. Entity section.

```
<!ENTITY % Int         "CDATA"> <!-- an integer number -->
<!ENTITY % Real        "CDATA"> <!-- a real number -->
<!ENTITY % String      "CDATA"> <!-- a character string -->
<!ENTITY % Bool        "(no | yes)">
<!ENTITY % Coord       "%Real;"> <!-- coordinates, given in millimeters -->
<!ENTITY % Dimension   "%Real;"> <!-- dimensions, given in millimeters -->
<!ENTITY % Layer       "%Int;"> <!-- layer number -->
<!ENTITY % GridUnit    "(mic | mm | mil | inch)">
<!ENTITY % GridStyle   "(lines | dots)">
<!ENTITY % WireStyle   "(continuous | longdash | shortdash | dashdot)">
```

Listing 3. Element definitions.

```
<!ELEMENT eagle (compatibility?, drawing, compatibility?)>
<!ATTLIST eagle
          version       %Real;          #REQUIRED
          >
          <!-- version: The EAGLE program version that generated this file, in the form V.RR -->

<!ELEMENT drawing (settings?, grid?, layers, (library | schematic | board))>

<!ELEMENT library (description?, packages?, symbols?, devicesets?)>
<!ATTLIST library
          name          %String;        #REQUIRED
          >
```

Bool can only be “no” or “yes” and a GridStyle can be “lines” or “dots”. These entity data types can then be used throughout the file to refer to the allowable values using the `"%type;"` notation. For example, a Layer is an Int which is actually a CDATA. But don’t worry, Eagle will convert the CDATA to an Int when it loads the file so only number formats are implicitly allowed. After the entity section is the Element section which defines how the XML structure and data will be stored in an XML file. **Listing 3** shows some example element definitions.

Element definitions are a little trickier because they list all of nested elements after the element name. In the above program, the “eagle” element can contain the optional element “compatibility”, the required element “drawing” and then another optional “compatibility” element. The attributes for the eagle element are defined next in the !ATTLIST element along with their data types and whether it is required, implied, a fixed value or has a default value. In the example the eagle element has a required attribute called version which should be a real number. The drawing element can possibly contain a settings element, possibly a grid element, a layers element and then a library, schematic or board element. This is how the nested elements are defined until an element only has attributes and no nested elements.

A real example

Let’s take a look at an extract of a schematic file in **Listing 4** now that we’ve seen a little about how Eagle uses XML. The … characters signify where text was removed from the real file for the example and are not part of the actual file.

```
</library>
</libraries>
...
</schematic>
</drawing>
</eagle>
```

Now chances are that you won’t ever need an Eagle XML file directly but it’s nice to have the option. However, the text based format does mean that you can do many text based operations on them like checking the differences between two files. It also makes it easier to store the files in a version control system if you want to. Own your data! ⏮

(150639)

Web Link

[1] XML 1.0 Specification: www.w3.org/TR/REC-xml/

Listing 4. Example schematic file (extract).

```
<package name="DIL20">
<description>&lt;b&gt;Dual In Line Package&lt;/b&gt;</description>
<wire x1="12.7" y1="2.921" x2="-12.7" y2="2.921" width="0.1524" layer="21"/>
...
<pad name="1" x="-11.43" y="-3.81" drill="0.8128" shape="long" rot="R90"/>
...
<text x="-13.081" y="-3.048" size="1.27" layer="25" rot="R90">&gt;NAME</text>
<text x="-9.779" y="-0.381" size="1.27" layer="27">&gt;VALUE</text>
</package>
...
</packages>
...
</library>
</libraries>
...
</schematic>
</drawing>
</eagle>
```

Coming soon: free STM32 ODE Poster download powered by Mouser Electronics

Mouser Electronics and Elektor have teamed up to bring you a yet another info-poster. The idea is simple: (1) download the poster artwork file (a fat pdf) free of charge, (2) have it printed at your local print shop; (3) roll it up; and (4) tack it on the wall, ceiling or door in your dorm, lab or college classroom!

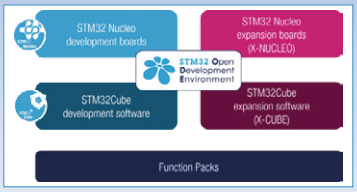
This time the poster has a richly detailed exploded view of the **STMicroelectronics STM32 Open Development Environment (ODE),** a prototyping and development environment that facilitates the design of embedded projects by providing easy-to-use hardware and open-source software functional blocks. With its unique combination of a broad range of stackable boards based on leading-edge integrated circuits and modular software — from driver to application examples — the STM32 Open Development Environment enables engineers and makers to easily and quickly transform their bright ideas into optimized projects.

STM32 Nucleo development boards include a 64-pin STM32 microcontroller based on a 32-bit ARM® Cortex®-M core and are available in a variety of memory sizes and peripherals as well as computing power versions. Thanks to the embedded ST-LINK/V2-1 debugger/programmer tool with mini USB port, you'll have all the tools to start developing and debugging code right away.

With the STM32 ODE, your project can grow at the speed of thought as Nucleo development boards have a standardized morpho connector giving access to all of the STM32 I/Os and enabling connections with STM32 Nucleo expansion boards (X-NUCLEO). Expansion boards provide more than 20 different hardware functions including the Connect, Move-Actuate, Power-Drive, Sense and Translate series and can be stacked like building blocks to let your project grow in a modular way. Moreover, STM32 Nucleo development boards also have an Arduino Uno R3 connector for you to easily integrate your favorite shields in your project.

Tired of spending sleepless nights developing software from scratch again and again? Developers will enjoy peace of mind and appreciate having more time to be creative thanks to the STM32 ODE with the STM32Cube software environment and embedded software libraries, including Hardware Abstraction Layer (HAL) APIs, as well as STM32Cube expansion software (X-CUBE) with pre-defined software projects created for the different expansion boards. And no, you're not dreaming: STM32 Nucleo development and expansion boards are budget-friendly and all the software is available for free!







The free STM32 ODE poster powered by Mouser Electronics is available for downloading for a limited time starting August 31, 2016, from **www.elektor.com/STM32-ODE-Poster**. Got it? Get it!

(160082-1) **www.elektor.com/STM32-ODE-Poster**

LoRa technology on TraceME

Holland based KCS BV has upgraded its successful TraceME module TM-178, targeted for tracing and controlling vehicles and other powered equipment.

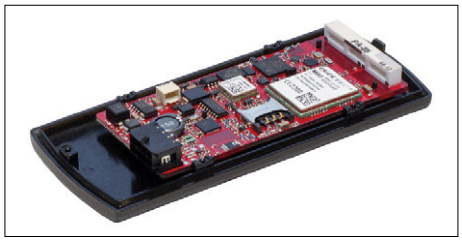
The TM-178/R9H7 module has been upgraded with optional LoRa™, Wi-Fi, Bluetooth Smart (BLE), ANT/ANT+ and proprietary RF, which enables easy integration with existing wireless networks and specific custom mobile App's on smartphones and tablets.

The LoRa™ technology offers a communication range up to 60 km line of sight. The module offers an advanced indoor and outdoor location based positioning solution, which covers a variety of IoT-applications and enables stolen object/vehicle recovery.

The TM-178 is equipped with external power and battery backup connection, basic I/O-connectivity and multiple on-board sensors. The unit contains multiple integrated antennas for GPS/Glonass, GSM (2G/3G) and RF functionality. The functionality of the module can be remotely programmed to fit any job. From basic/general functionality to advanced/low-level application specific detailed functionality.

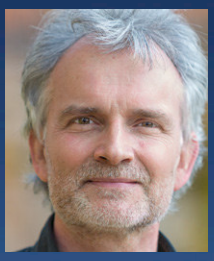
With a compact size of 91 x 40 mm, weighing 30 grams, a battery lifespan of more than 10 years, the module offers endless OEM integration possibilities. Optionally, the module can be ordered in a robust IP67 housing.

www.trace.me **(160082-3)**





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

Startupbootcamp Announces IoT Cohort for 2016 Program

Startupbootcamp IoT | Connected Devices today announced the ten startups that will join the first cohort of its London-based program. A diverse range of startups working on an Internet of Things (IoT) device — from security, to intelligent horse monitoring technology and smart manufacturing — were selected to join the accelerator program.

After receiving, and reviewing, 425 startup applications from 61 countries, 21 finalists were invited to attend Startupbootcamp Selection Days held between July 6-8th. The final ten to join the program were chosen following an intense three-day selection process that included expert workshops, pitches to program partners such as Cisco, DLL and Premier Farnell, and mentor feedback sessions. Here are the 10 selected startups:

CityCrop (Greece), a connected intelligent indoor garden that allows you to produce pesticide-free fruits and vegetables 24/7, all year-round.

Doordeck (UK), a hardware and software product for controlling electronic locks such as magnetic locks used in commercial and industrial spaces, strike locks, garage shutters, gates, etc.

Eskesso (Spain), a mobile app, cloud-based online platform and a "sous vide" smart Wi-Fi compatible cooking appliance to allow people to eat properly.

HomyHub (Spain), a Plug and Play Smart Device for garage doors that lets you control, monitor and share your garage from your smartphone.

Joyride (Canada), building a tracking device for cyclists and bike fleets.



MOONA (France), tracks your sleep and sets the perfect pillow temperature to fall asleep quickly, have a restorative night and wake up beautifully rested!

Thingtrax (UK), building a tracking solution for Injection Molding machines targeting mid-level manufacturing customers.

Trackener (UK), an intelligent horse monitoring technology helping horse owners give the best to their horses and alert them if an abnormal situation is detected.

Woogie (Romania), building a new way of AI interaction with technology especially designed and engineered for kids.
YodelUP (Canada), an "on-glove" wearable device that enables skiers and snowboarders to instantly access their smartphone and gadgets while wearing gloves.

In its first year of running the program, Startupbootcamp IoT | Connected Devices provides each startup with €15,000 in cash, office space in London connected to a full fast prototyping lab, and access to a global network of IoT and hardware experts and investors.

(160082-2)

Advertisement

HAMMOND MANUFACTURING

Housings for Raspberry Pi, Arduino and many other bareboard computers
- Enclosure
- Platform

+ 44 1256 812812
sales@hammondmfg.eu



/1593HAM.htm

The choice is yours.
- Enclosure for all round protection
- Platform for all round access
- Design-specific versions for all popular models
- Visit hammondmfg.com for full details



/1593HAMEGG.htm

Mobile Communications Instrumentation

meets the challenge of Big Data and IoT

By **Meik Kottkamp**, Technology Manager, Rohde & Schwarz (Germany)



Figure 1. RF and protocol testing systems help network operators select suitable wireless device providers. The test solutions ensure that the devices comply with mobile network standards.

The Internet of Things, enhanced mobile broadband and self-driving cars require extreme mobile network performance. The next generation of mobile communications, 5G, is expected to deliver the necessary performance. But with 5G still in its infancy, enhancements such as LTE/LTE-Advanced networks provide an excellent evolution path. Evolving instrumentation is needed to ensure technology and user experience standards are maintained.

It is 25 years since the 900 MHz GSM standard marked the birth of digital wireless communications. The hunger for data in cellular communications remains insatiable and demand for further technological progress persists. A tenfold increase in mobile data traffic is expected over the next six years, and experts predict an exponential increase in the number of things (Internet of Things, IoT) that communicate with each other via mobile networks. Inexorable growth in the both the number of subscribers — and the amount of data they consume — means mobile network operators need to guarantee an excellent quality of experience long into in the future.

2G / 3G / 4G technologies and future improvements

State of the art, commercially available LTE A devices achieve 600 Mbit/s in ideal lab environments. In real networks, propagation conditions and the shared channel principle reduce achievable download speeds due to the fact that available bandwidth is divided among all active subscribers. Nonetheless, LTE/LTE-A technology has significantly enhanced available data rates and network capacity. There are a number of reasons for this.

- A wide system bandwidth of 20 MHz that can be provided to an individual subscriber as well as the ability to bundle up to five of these 20-MHz carrier frequencies for each subscriber (commonly known as carrier aggregation, CA).
- Use of spatial multiplexing (MIMO technology), i.e. using from two to eight/four transmit/receive antennas.
- Fast OFDMA multiplexing enables the frequency and time resource allocation to be changed on a millisecond basis.
- High quality modulation methods, specifically QPSK, 16QAM, 64QAM and 256QAM.

Meeting demands by measuring against standards

Introducing LTE/LTE-A has allowed network operators to meet increasing demands. Several key LTE improvements have resulted from enhancements introduced by the 3GPP standardization body. As an example, LTE's enhanced multimedia broadcast multicast service provides a highly efficient method of addressing mobile TV applications, for example, where many subscribers receive the same data simultaneously. This mode also allows wireless devices to install new software efficiently without requiring individual data connections for each device.

3GPP specifications provide a special mode to use, for example, an email application running in the background via WLAN while transmitting video data via LTE. Alternatively, operating LTE/LTE-A in unlicensed frequency bands allows data rate capability to be increased using the carrier aggregation feature.
In heterogeneous networks such as pedestrian areas of cities, coordinated multipoint transmission and reception (CoMP) makes it possible to transmit a signal to a wireless device at the cell boundary in a coordinated manner. Using MIMO technology and also influencing the baseband signal (precoding) in a coordinated manner allows optimal coverage at cell boundaries.

The introduction of device-to-device (D2D) capabilities is of special importance, as these provide two fundamental new functions. First, the network-supported discovery function makes it possible for two spatially neighboring wireless devices to detect each other. Secondly, these devices and others in their vicinity will be able to exchange data directly, i.e. without going through the base station covering the area. This technology is limited to public safety and security applications or application-related broadcast services. However, it could be extended, for example to support automotive applications including autonomous driving.
Even though LTE/LTE-A networks demonstrate increasing performance, comprehensive 4G coverage will take time. Efficient handover to 2G and 3G technologies remains critical. There are also many cases where 2G / 3G data rates are sufficient and a low-tech approach allows cost-efficient solutions with long battery life. For example, modules with GPRS technology are often used in machine-to-machine (M2M) environments.

The high performance of LTE/LTE-A technology, its seamless cooperation with existing 2G/3G networks and the complementary use of WLAN enable network operators to meet subscribers' continuously growing big data requirements. Broadcast/multicast solutions enhance system flexibility.

LTE/LTE-A has already ushered in several improvements to serve M2M applications. For example, there is a category 0 for LTE user equipment that reduces the effort required to implement this class of devices (lower data rate requirements and no MIMO support). Processes have also been introduced to prevent mobile network overloading when large numbers of M2M devices attempt to access the network simultaneously.

The increasing number of things communicating with each other in the future (IoT) and new demands stemming from vertical branches of industry (automotive industry, health care, robot control, etc.) are expected to dictate further significant enhancements. That is why researchers in the mobile communications industry are already discussing 5G as they look toward 2020 and beyond. Instrument vendors such as Rohde&Schwarz and its subsidiaries SwissQual and ipoque offer a comprehensive portfolio for today's T&M tasks and are actively involved in 5G research and development.



Figure 2. Cost efficient mobile T&M equipment is used to install a base station.



Figure 3. Network operators use T&M equipment to analyze network performance and optimize the end user experience.



Figure 4. Mobile measuring receivers with directional antennas help operators monitor operational networks and identify interference.

How does T&M equipment contribute?

Test and measurement equipment plays a central role in both introducing new technologies and in operating networks. Countless test solutions are required to develop and manufacture mobile devices, components, base stations and switching nodes. Test solutions are also needed when deploying the network and verifying its performance.

Network operators have to select the right infrastructure products to operate their networks. Using instruments such as signal generators and signal and spectrum analyzers, they can qualify infrastructure products to select those that perform best. Many operators specify additional tests based on their specific network requirements. T&M equipment such as the R&S® CMW500 wideband radio communication tester emulates all required network functions; verifying whether a wireless device is behaving properly ensuring hardware is correctly implemented (for example, complying with specified maximum transmit power). When deploying base stations in the field, compact testers are needed to quickly verify whether regulatory requirements are being met. Following deployment, operators need to adjust parameters, such as handover thresholds, and identify coverage gaps in order to optimize their network and ensure the best possible data rates. QualiPoc from SwissQual implements a measurement application on a commercial smartphone. The test solution can be used like a normal app, enabling network operators to evaluate end user experience.

In the operator's core network, where all data streams are processed, it is increasingly important to be able to analyze data traffic down to the packet level. This makes it possible to classify the data traffic and optimally route a service's data packets through the network. IP analysis technology provides this insight. The same functionality is also of great interest for mobile device testing. When implemented on the R&S®CMW500, users can analyze which IP data streams (including protocols used) a smartphone maintains only due to applications running in the background.

There can always be unforeseen interference when operating mobile networks. Stationary network monitoring tools in the mobile network and mobile interference hunting solutions are used to identify and eliminate this interference as quickly as possible. For example, malfunctioning neon signs can create interference in base station receive bands and negatively impact all data traffic in a cell.

Test instruments are also helping to delineate the next generation of standards. Already signal generators and signal analyzers are used to evaluate potential 5G components. Test and measurement are indispensable because of their flexibility in terms of frequency range, bandwidth and transmission technology. They are also essential components of measuring systems used to analyze propagation conditions in new, undefined frequency bands. Multiport network analyzers will play a decisive role in implementing future antenna technologies. Finally, the influence of individual applications on data rates, signaling loads and current drain are under examination. This will be especially important for IoT modules as it must be possible to measure individual applications on the IP layer itself.

In all these areas, instrumentation is evolving to ensure technology and user experience standards are maintained. ⏮

(160068)

Messe München

Connecting Global Competence

Planet e:
Where the future begins.

Embedded solutions of tomorrow. Today.

- Embedded exhibition sector:
  November 8–11, 2016
- electronica Embedded Forum:
  November 8–11, 2016
- Embedded Platforms Conference:
  November 9–10, 2016
- All about electronica Embedded:
  electronica.de/en/embedded



Tickets & Registration: **electronica.de/en/tickets**

World's Leading Trade Fair for Electronic Components, Systems and Applications

Messe München I November 8–11, 2016 I electronica.de

electronica 2016

inside tomorrow

The Big Red passion

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!

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?



is about Big Time

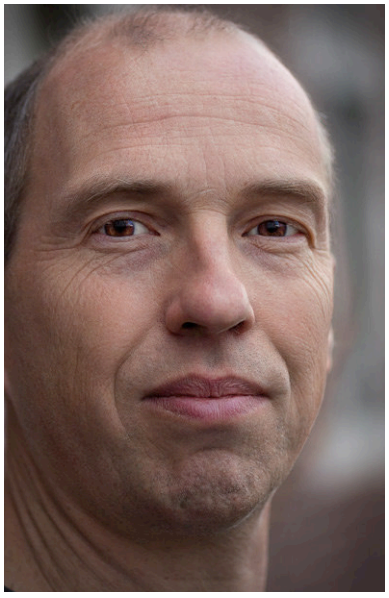


Anyone who wants to join the spirit and enjoy the sparks, should be in The Big Red. Everyone is welcome.

JOIN NOW! **www.elektor.com/the-big-red-2017**

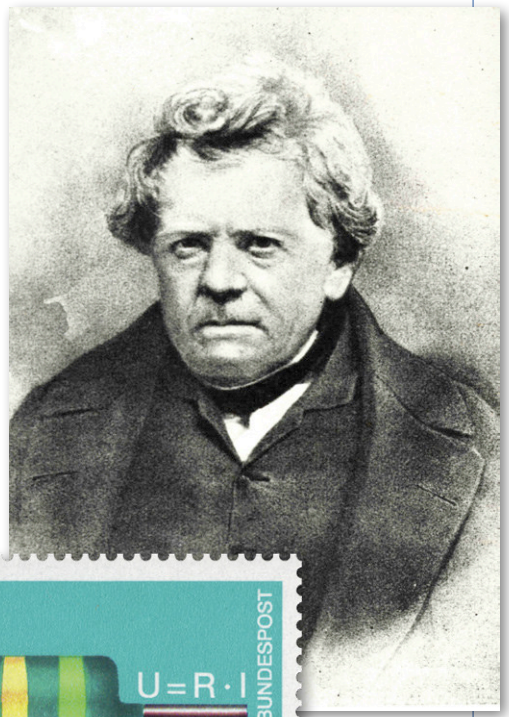
Welcome to the **DESIGN** section

By **Clemens Valens,** Elektor Labs



Georg Simon Ohm (1789 – 1854)

Georg Simon Ohm was one of seven children of which only three made it to adulthood: G.S. (b. 1789), Martin (b. 1792), and Elisabeth Barbara (b. 1794). The two boys were taught mathematics, physics, chemistry, and philosophy by their father Johann Wolfgang Ohm, a locksmith and autodidact who would have loved to be a scientist. G.S. went to university but preferred playing billiards over studying and consequently his father sent him out into the world to make a living on his own. He did so, albeit modestly, roaming around as a teacher and private tutor. Eventually he came back to the University of Erlangen where he obtained his Ph.D. in 1811, the same year as his younger brother Martin. Martin entered a career in mathematics that took him smoothly to a professorate at the University of Berlin, but G.S. was less lucky and in order to survive had to continue teaching at low-prestige schools, notably in Bamberg and Cologne. All this time G.S. dreamed of a real job at a university, like his successful younger brother, but he never managed to secure one.



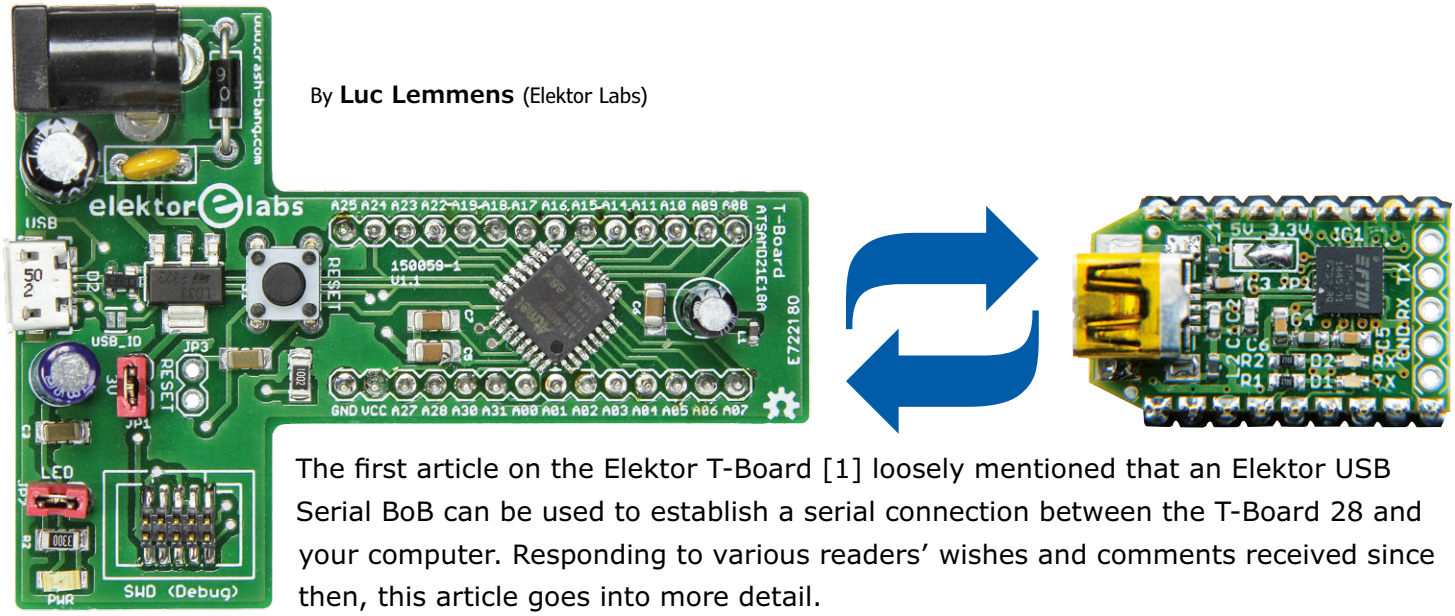
At this point in his lonely, unmarried, unhappy life G.S. concluded that in order to become a scientist he had to act like one and so he started experimenting and publishing his results. Inspired by the Danish physicist Hans Christian Ørsted who in 1820 discovered that a magnetic needle is deflected by a nearby electric current, G.S. concentrated his attention on electricity and magnetism. In 1825, in his first paper, he investigated the relationship between the decrease in the electromagnetic force exerted by a current-carrying wire and the length of the wire. Other scientists like Antoine-César Becquerel (not to be confused with his grandson Henri) and Peter Barlow where active in the same arena trying to formulate the law that described the phenomenon correctly. Barlow came close, but G.S. nailed it by introducing the concept of internal resistance of a battery. This brilliant insight not only resulted in his famous law, but it also allowed him to solve many unanswered questions that had eluded researchers for years. His masterpiece from 1827 (the same year as Ampère's *magnum opus*), "*Die galvanische Kette, mathematisch bearbeitet" (*The Galvanic Circuit Investigated Mathematically), presents G.S.'s theory of electricity and was quickly adopted by the scientific community, except in Germany where physics was non-mathematical. So, in spite of his fame and stature, G.S. still could not get a job at a university. His brother's divergence with the Ministry of Education that had earned him the reputation of a dangerous revolutionary may have had something to do with it too.



The story does have a happy ending, sort of: at the end of his life even the Germans could no longer ignore the importance of G.S.'s work and so, in October 1852, he obtained the Chair of Physics at the University of Munich, less than two years before his death. ⏮

(160073)

Linked: T-Board 28 and Serial e-BOB



By **Luc Lemmens** (Elektor Labs)

The first article on the Elektor T-Board [1] loosely mentioned that an Elektor USB Serial BoB can be used to establish a serial connection between the T-Board 28 and your computer. Responding to various readers' wishes and comments received since then, this article goes into more detail.

Firstly and crucially, be sure to get the I/O voltages right. Both modules have a jumper marked JP1 to select the voltage range of the I/O pins. The selection will depend on the circuit you are constructing with the T-Board, i.e. if you use 3.3 V or 5 V logic in the external circuitry hooked up to the T-board.

For the serial connection itself it is immaterial whether 5 V or 3.3V is selected, as long as "the volts" are identical on both boards. However, if you plan to power your project from the USB BoB's power supply, the maximum current is limited to just 50 mA in case of 3.3 V, while selecting 5 V gives you 500 mA provided the BoB is connected to a USB-2.0 port (or 150 mA for USB-1.x).

The voltage selection on the Serial BoB and T-Board is set on JP1. On the Serial BoB it's a solder bridge, on the T-board a 3-pin pinheader. Leave JP1 on the T-Board open when powering this board via K3! When using an external power supply for the T-Board, leave the +5 V pin on K3 open.

For the rest it's just a matter of connecting the pins of the serial interface and — if you like — the power supply. **Figure 1** shows the digest. Please note that TxD is connected to TxD, and RxD to RxD (i.e. no cross-wiring). The pin labels of T-board's K3 are named after the corresponding pins of an FTDI cable (which can be directly connected to K3).

If you want to use this configuration to make an Arduino-style prototyping system that's Arduino IDE savvy, please don't forget to install the 16-MHz quartz crystal in socket X1 on the T-Board. And lest we forget: IC3 should contain an Arduino compatible serial bootloader, else all jumpering and wire linking is futile!

(160106)

[1] T-Boards 8/14/28, Elektor September 2014, www.elektormagazine.com/130581

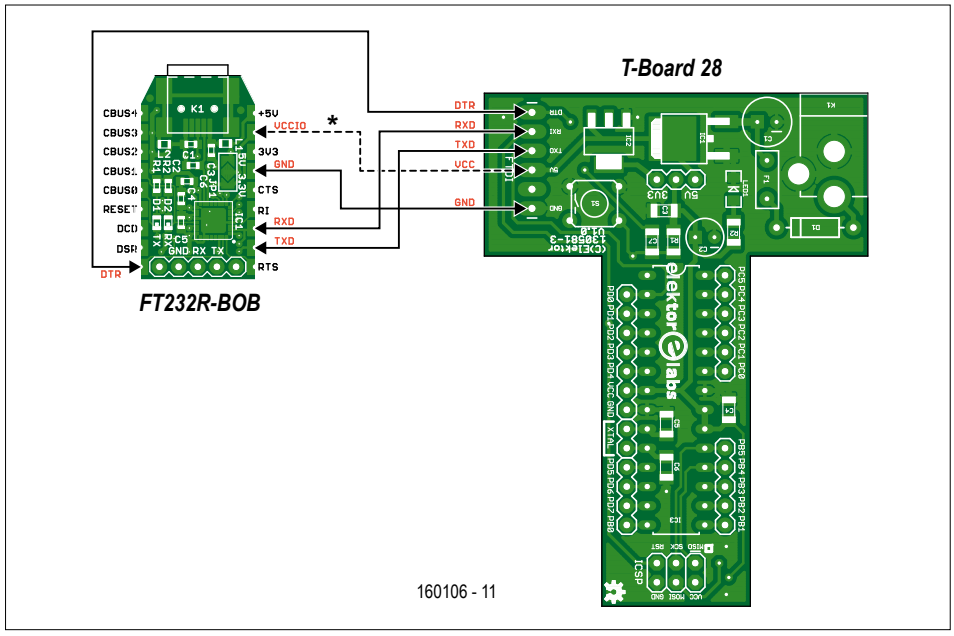


Figure 1. In wires: linking the T-Board 28 to the USB serial RS232 BoB.

Swiss Pi

A Swiss army knife for the Raspberry Pi

By **Ilse Joostens** and **Peter S'heeren** (Belgium)



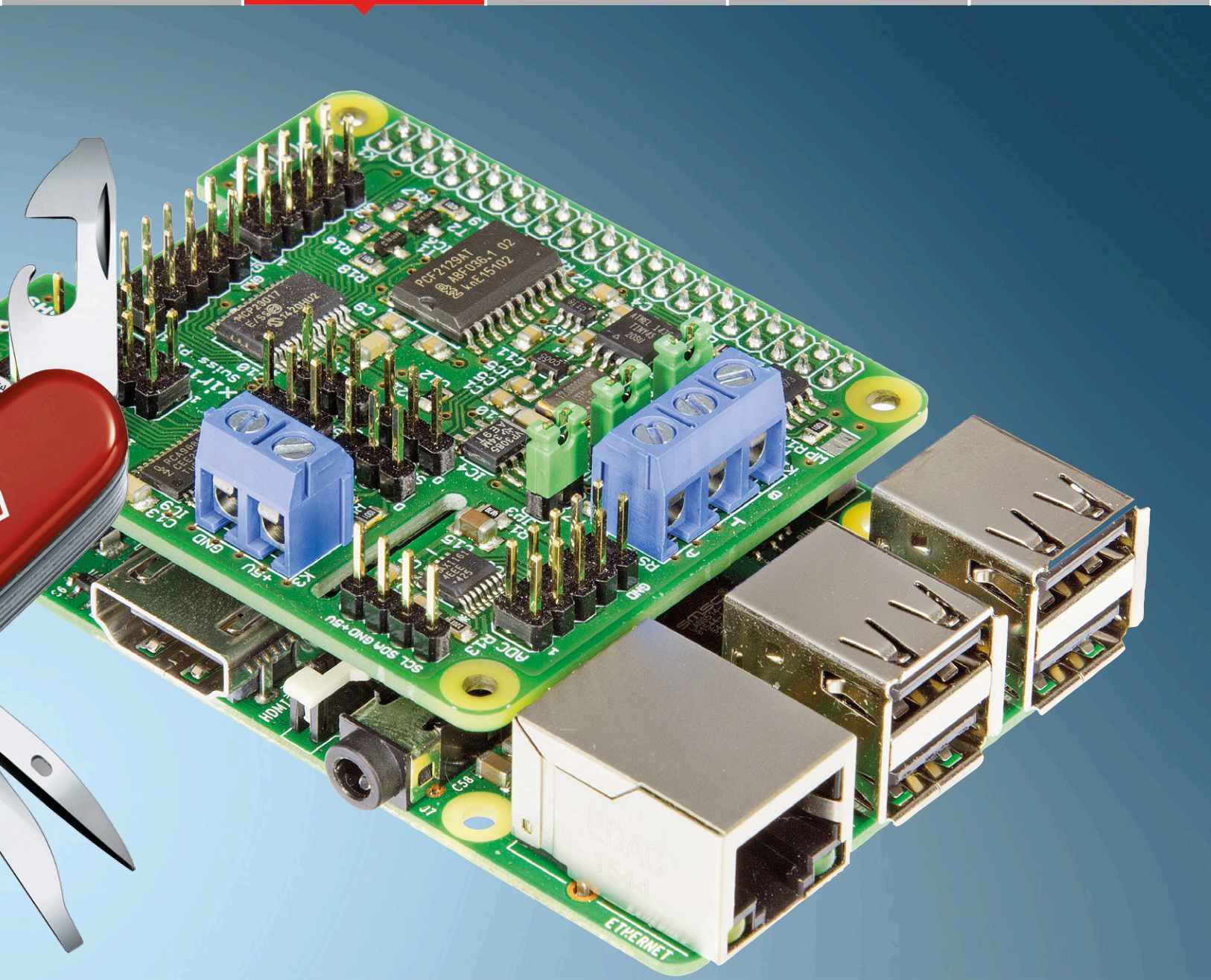
The Raspberry Pi single board computer (SBC) was launched in the spring of 2012, and now — over four years later — it is still extremely popular. Not content to rest on their laurels, the Raspberry Pi Foundation has kept busy with upgrades and rolled out the third version last February. One thing that has certainly contributed to the popularity of the Raspberry Pi is that the 40-pin GPIO connector makes it very easy to extend the basic board with custom modules. We felt challenged.

However, the GPIO connector of the Raspberry Pi has some limitations compared to Arduino SBCs. For instance, the GPIO lines are connected directly to the CPU, which makes them fairly vulnerable, and there is hardly any hardware support for generating PWM signals (only one PWM pin is available). There are also no analog (ADC) inputs. The Swiss Pi HAT described here eliminates these shortcomings and provides a number of extra features, including a real-time clock and an RS-485 interface.

The associated software, which can be downloaded for free from [1], runs under both Linux and Windows. A server program allows several clients to use different parts of Swiss Pi concurrently. The server can also be accessed using various scripting languages, including Python and PHP. Among other things, this allows the board to be controlled over the internet through a web interface. Finally, there is

Features

- 16 GPIO lines with configurable pull-up, 5-V signal levels
- 16 12-bit PWM channels, configurable 24–1526 Hz, 5-V signal levels
- 4 connectors for remote control servos (shared with PWM channels 0 to 3) and supply voltage connector
- 8 12-bit ADC inputs, range 0–4.096 V
- Half-duplex RS-485 interface compliant with TIA/EIA 485A standard, with ESD protection
- Precise real time clock with battery backup, 3 ppm accuracy
- I^2C extension connector, 5-V signal levels
- ID EEPROM



a graphical user interface program that makes it easy to operate all the functions on the board interactively, which is very handy for testing.

Hardware

As befits a Swiss army knife, the Swiss Pi provides a broad assortment of handy tools, as can be seen from the schematic diagram in **Figure 1**. They are described individually below.

I/O expander

We chose the I²C MCP23017 I/O expander to control the 16 GPIO lines. The I/O lines can be configured as high-impedance inputs with pull-up resistors or as outputs. This IC is often used together with the Raspberry Pi, so a lot of example programs and tutorials on this topic are available on the Internet.

PWM controller

Here we use the PCA9685, an I²C 16-channel LED controller, to generate the PWM signals. With a configurable PWM frequency range of 24 to 1526 Hz and 12-bit resolution, this IC can be used to drive LEDs (including RGB LEDs) as well as motors (with MOSFET bridge power stages) or remote control servos. For the latter purpose, PWM channels 0 to 3 are fed out to four three-pin connectors for direct connection to remote control servos. Due to the relatively high power consumption of remote control servos, they are not powered directly from the Raspberry Pi. An external power source must be connected to K3 for this purpose. If you want to use more than four servos, you will have to provide the additional connectors yourself.

In combination with an RC filter circuit as shown in **Figure 2**, the PCA9685 can also be used as a DAC, for example to generate an 0–10 V control signal for light dimmers.

ADC

The MAX11614EEE+ analog to digital converter IC used here has eight chan-

nels with 12-bit resolution and an internal 4.096 V voltage reference. The channels can be used in single-ended or differential mode, although the software included with the Swiss Pi only uses the channels in single-ended mode. The dimensions of Raspberry Pi extension cards are standardized by the HAT specification, which means that space on the board is limited. As a result, there are no additional protection diodes provided for the ADC inputs. If you use sensors that are powered from

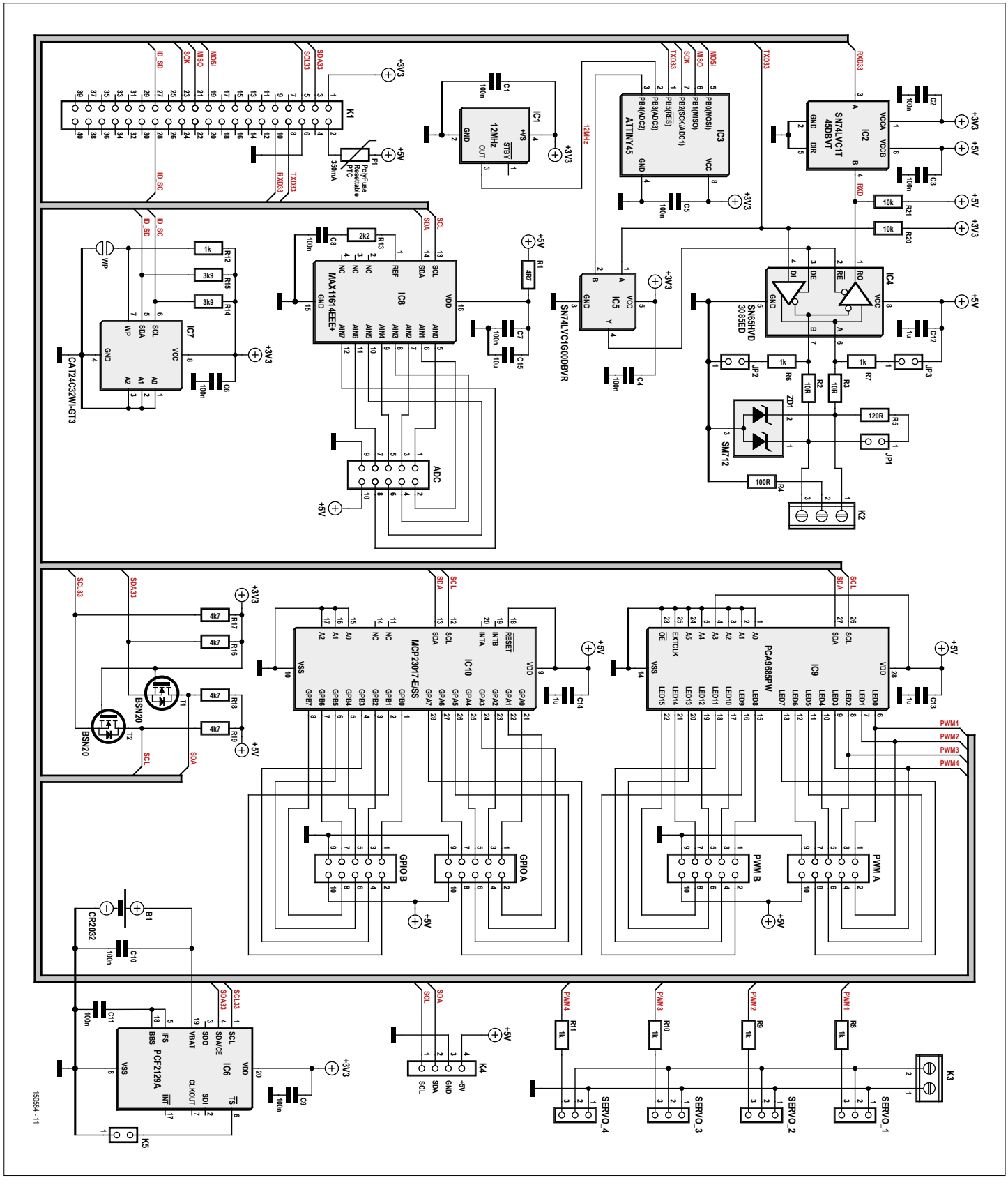


Figure 1. Schematic diagram of the Swiss Pi HAT.

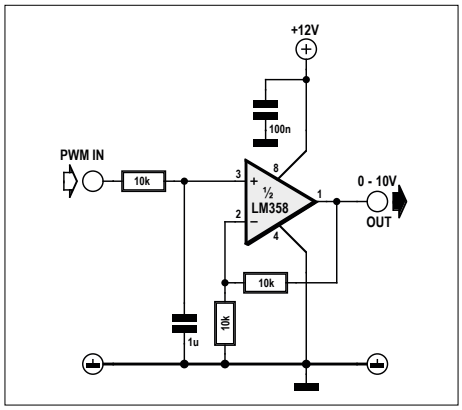


Figure 2. Converting a PWM signal into a clean analog voltage.

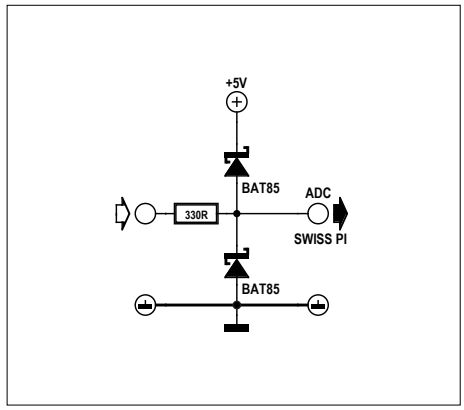
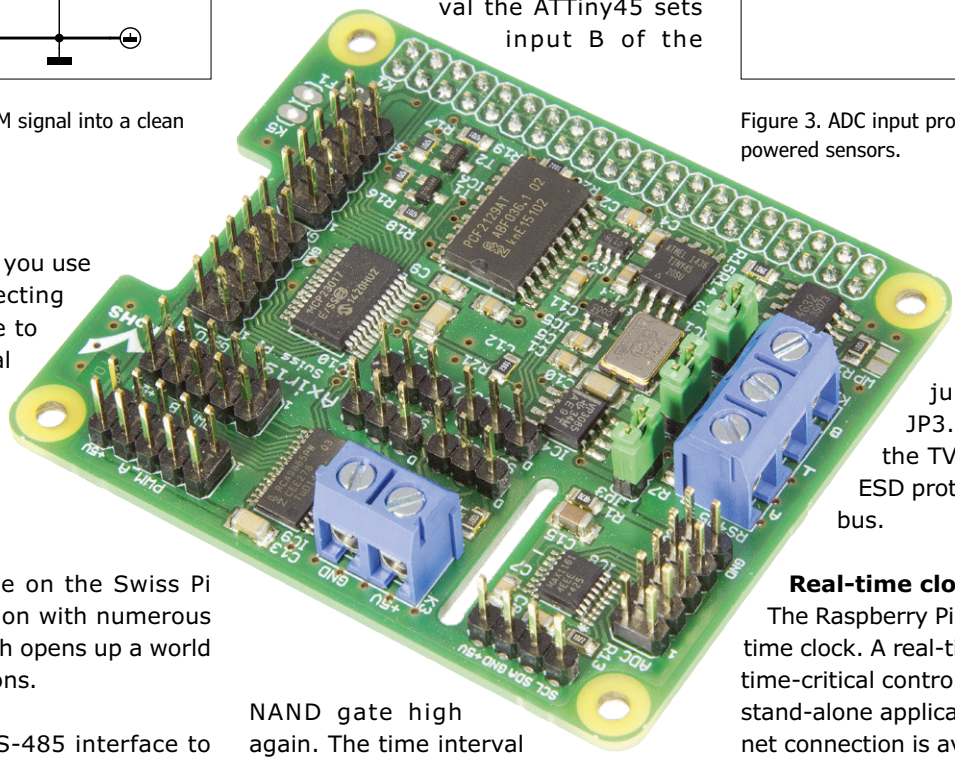


Figure 3. ADC input protection for externally powered sensors.

an external source or you use relatively long connecting cables, it is advisable to provide two additional Schottky diodes and a resistor for extra protection (see **Figure 3**).



RS-485

The RS-485 interface on the Swiss Pi enables communication with numerous Modbus devices, which opens up a world of potential applications.

We connected the RS-485 interface to the standard UART of the Raspberry Pi, rather than using a relatively expensive SPI to RS-485 converter IC. This has the additional advantage that you do not need a special driver. However, it has the disadvantage that it is not easy to switch the RS-485 transceiver (IC4) from transmit mode to receive mode with the RE/DE control line after transmitting a data word. The processor of the Raspberry Pi does not provide any hardware support for this, and using software with a GPIO pin for this control function is not particularly reliable because Linux is not a real-time operating system.

For that reason, we use a small microcontroller (an ATTiny45, IC3) and a NAND gate (IC5) to handle mode switching for the RS-485 transceiver. When the serial interface is quiescent and IC3 is inactive, both inputs of the NAND gate are high and the output is low, putting the RS-485 transceiver in receive mode. When the Raspberry Pi starts to send data, input A of the NAND gate is pulled low by the falling edge of the TX signal (start bit). This causes the output to go high, putting the RS-485 transceiver directly in transmit mode. The TX signal is also applied to pin 1 of the ATTiny45, which pulls input B of the NAND gate low when it detects a falling edge on this input. As a result, the RS-485 transceiver remains in transmit mode even after the TX signal goes high again. After a configurable time interval the ATTiny45 sets input B of the NAND gate high again. The time interval must be chosen to ensure that this occurs just before the end of the last stop bit, after which the RS-485 transceiver goes back into receive mode.

The delay time of the ATTiny45 can be configured via the SPI interface of the Raspberry Pi and depends on the baud rate, the word length, the parity bits and the number of stop bits. We could not use I²C for this because the ATTiny45 has a limited universal serial interface (USI), which means that most of the I²C protocol has to be implemented in software. That is naturally not as fast as hardware, so clock stretching would be necessary to allow the ATTiny45 to track the I²C bus. A hardware error in the I²C Master of the Raspberry Pi prevents it from supporting this clock stretching function properly, so our only option was to use SPI instead.

Here the RS-485 transceiver is powered from +5 V, with IC2 providing the required level conversion to 3.3 V for the RX line to the Raspberry Pi. The bias resistors and termination resistor for the RS-485 bus can be configured with jumpers JP1, JP2 and JP3. Finally, R2, R3 and the TVS diode ZD1 provide ESD protection for the RS-485 bus.

Real-time clock

The Raspberry Pi does not have a real-time clock. A real-time clock is handy for time-critical control tasks, particularly in stand-alone applications where no Internet connection is available and therefore access to an NTP server is not possible. Here we opted for the PCF2129A, which has a built-in crystal. This IC is inexpensive and very precise, with internal temperature compensation and an accuracy of 3 ppm. A CR2032 button cell serves as a backup battery with a lifetime of about ten years.
Up to two time stamps can be generated via K5. For more information about this, see the PCF2129A data sheet

I²C level converter, polyfuse and ID EEPROM

With the exception of the real-time clock, all I²C devices are powered from +5 V. Two n-channel MOSFETs (T1 and T2) in combination with resistors R16–R19 are used to provide the necessary level conversion for the SDA and SCL lines.
Polyfuse F1 limits the total current consumption of the circuit to 350 mA. This ensures that the Raspberry Pi still has enough current to work properly, even with a power source with relatively low capacity. That must be taken into account

when you connect extensions to the GPIO pins and/or PWM channels.

Finally, there is an ID EEPROM in the form of IC7. It is necessary to conform to the requirements of the Raspberry Pi HAT specification. The idea is that the Linux kernel reads the ID EEPROM during boot-up and then, based on this information, configures the GPIO lines of the Raspberry Pi correctly for the HAT concerned (auto configuration, device tree). As far as we know, support for this on the Raspberry Pi is still fairly limited at present.

Component List

Resistors
Default: SMD 0805
R1 = 4.7Ω
R2,R3 = 10Ω high pulse loading
R4 = 100Ω ≥250mW
R5 = 120Ω ≥250mW
R6–R12 = 1kΩ
R13 = 2.2kΩ
R14,R15 = 3.9kΩ
R16–R19 = 4.7kΩ
R20,R21 = 10kΩ

Condensatoren
C1–C11 = 100nF, 0805
C12,C13,C14 = 1µF, 1206
C15 = 10µF, 1206

Semiconductors
T1, T2 = BSN20
IC1 = 12MHz oscillator block, SMD, type LFSPXO018037
IC2 = SN74LVC1T45DBVR
IC3 = ATtiny45-20SU, programmed
IC4 = SN65HVD3085EDG4
IC5 = SN74LVC1G00DBVR
IC6 = PCF2129AT
IC7 = BR24G32FJ-3GTE2, 32 Kbit I²C EEPROM
IC8 = MAX11614EEE+
IC9 = PCA9685PW,112
IC10 = MCP23017-E/SS
ZD1 = CDSOT23-SM712, 12V Transient Voltage Suppressor

Miscellaneous
F1 = 16V 0.35A Polyswitch type 1206L035/16YR
B1 = 2032 button cell holder
K1 = 40-way (2x20) socket, 0.1'' pitch
K2 = 3-way PCB screw terminal block, RS-485-bus, 5mm pitch
K3 = 2-way PCB screw terminal block, 5mm pitch
K4 = 4-pin pinheader, 0.1'' pitch
K5 = 2-pin pinheader, 0.1'' pitch
ADC = 10-pin (2x5) pinheader, 0.1'' pitch
PWM_A, PWM_B = 10-pin (2x5) pinheader, 0.1'' pitch
GPIO_A, GPO_B = 10-pin (2x5) pinheader, 0.1'' pitch
SERVO_1 – SERVO_4 = 3-pin pinheader, 0.1'' pitch
JP1–JP3 = 2-pin pinheader, 0.1'' pitch, with jumper

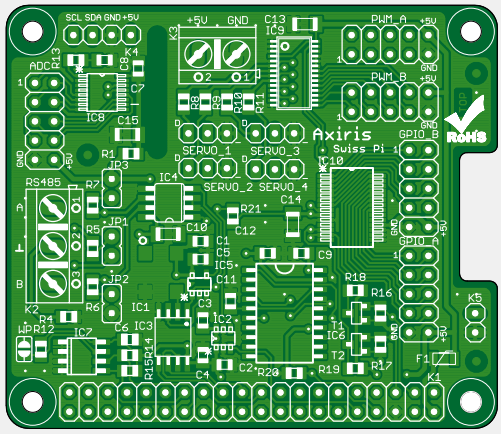
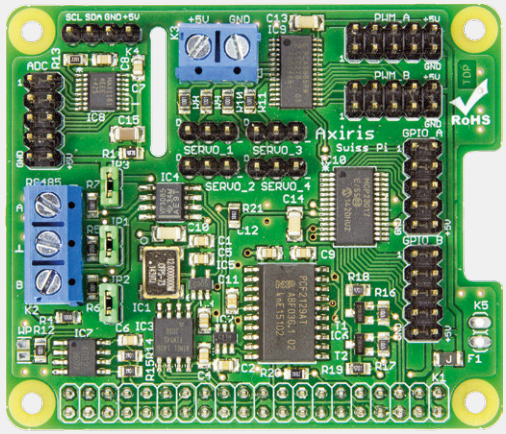


Figure 4. The component side of the compact PCB, which fits neatly on the Raspberry Pi board.



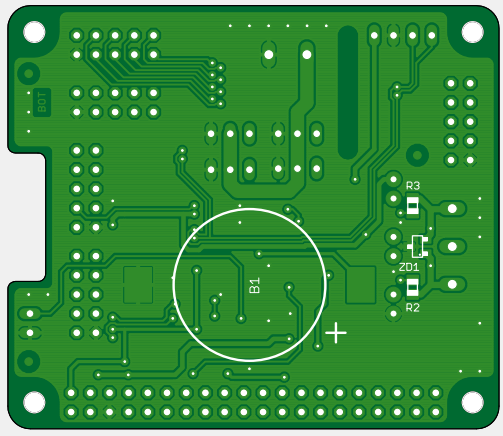


Figure 5. The solder side of the PCB.

PCB

We designed a compact PCB for the Swiss Pi. See **Figure 4** for the component side and **Figure 5** for the solder side.

Software

Swiss Server

The Swiss Server software provides the functions of the Swiss Pi module via the network ports and standard I/O according to its own communication protocol. The server can accept multiple client connections, allowing several clients to work with the Swiss Pi concurrently. This facilitates wide-ranging flexibility.
The communication protocol consists of commands and corresponding responses. A client sends a command, and the server returns a response once the command has been executed. The communication protocol is coded in ASCII format. The syntax is designed to make the communication protocol easy to use both manually and in software.

The server supports a large number of commands for working with the Swiss Pi module. The communication protocol is fully asynchronous, which means that a client does not have to wait for the response to a command before sending the next command. This allows clients to send several commands in direct succession, which dramatically increases communication efficiency. Each command can optionally be provided with an identification number, which is sent back with the response.

The server also has an option for exchanging data bytes directly between the RS-485 interface and a network port. If this option is disabled, clients can use commands to send data bytes over the RS-485 interface.

The Swiss Server works with both Linux and Windows. In the latter case the Swiss Pi module has to be connected to the computer through a USB port and an adapter board (AxiCat). More information about this will be provided in a future article on AxiCat.

RS-485 Control

This program talks with the ATTiny45 in order to configure the RS-485 functions using a registry file that can be accessed over the SPI interface. If you use the Swiss Server, you will probably

not need this program very often because the Swiss Server provides the most frequently used RS-485 functions.

The program includes a detection function to determine the time parameters for transmit mode. This detection procedure can be executed on a naked Swiss Pi board; no specific hardware setup is required.

I/O Card Explorer

This program provides a graphical user interface to make working with the I²C devices on the Swiss Pi board easy and interactive, as shown in **Figure 7**. The program can communicate with the hardware in various ways. If the program is running on the Raspberry Pi, it can control the Swiss Pi directly. In that case the program has exclusive access to the Swiss Pi.

If the Swiss Pi is operating under control of the Swiss Server, you can configure the program to communicate with the hardware through the Swiss Server.

With the Swiss Server you can also run the program on a different computer and connect to the Swiss Server over a network. Different instances of the program and other clients can connect to the Swiss Server concurrently, allowing several parties to work with the Swiss Pi module at the same time.

Extensions

To make the Swiss Pi module more useful in the real world, it is desirable to provide a number of extensions in the future. Among other things, we are thinking about a relay/SSR card, a board with galvanically isolated digital inputs, a 4–20 mA current loop adapter, a 0–10 V controller for dimmers, a DC motor controller with bridge output, a RGB driver board for LED strips, and so on. The Swiss Pi can also be extended through the I²C connector with our eight-channel 1-wire mini host, enabling the use of 1-wire devices such as DS18B20 temperature sensors.

Finally, there is also a version of the Swiss Pi available for BeagleBone, dubbed "Swiss Cape". In terms of hardware it is comparable to the Swiss Pi. The Swiss Cape uses the same software as the Swiss Pi.

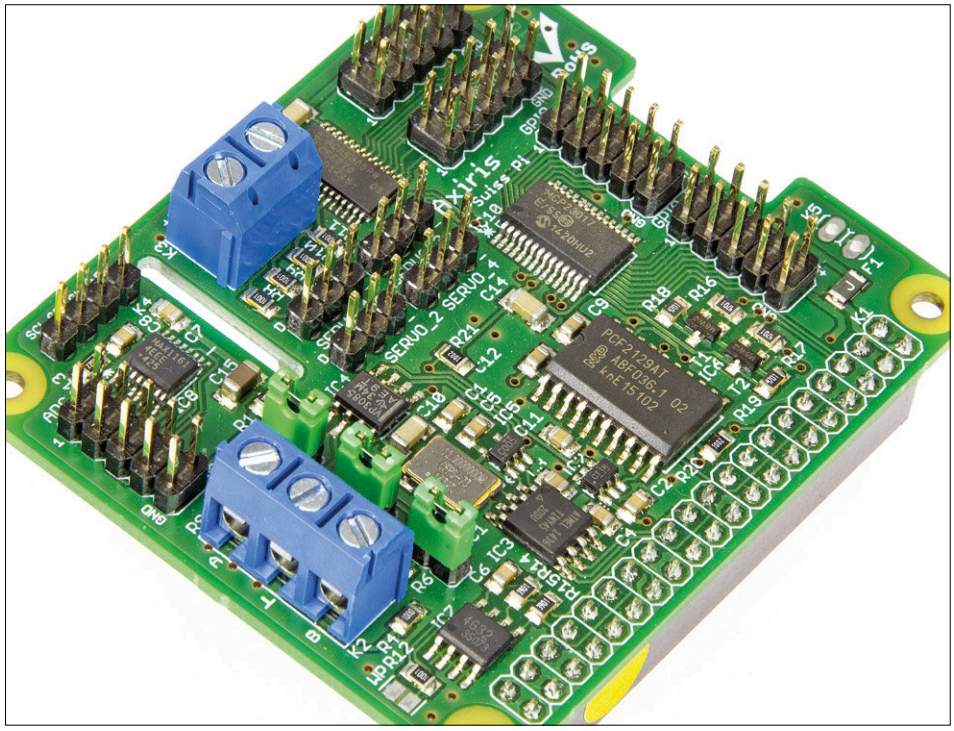


Figure 6. The fully assembled prototype module.

The layout files (Eagle) and the Gerber files are both available at [1]. Fully assembled Swiss Pi modules can be ordered from the Elektor Store [2].

(150584)

Web Links

[1] www.elektormagazine.com/150584

[2] www.elektor.com/swiss-pi

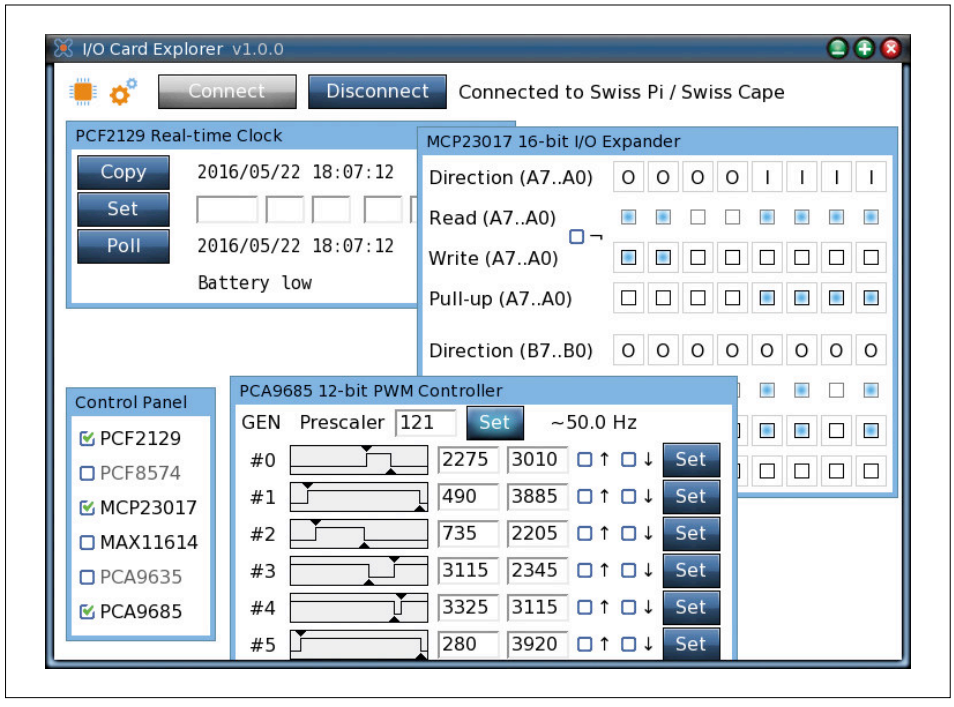


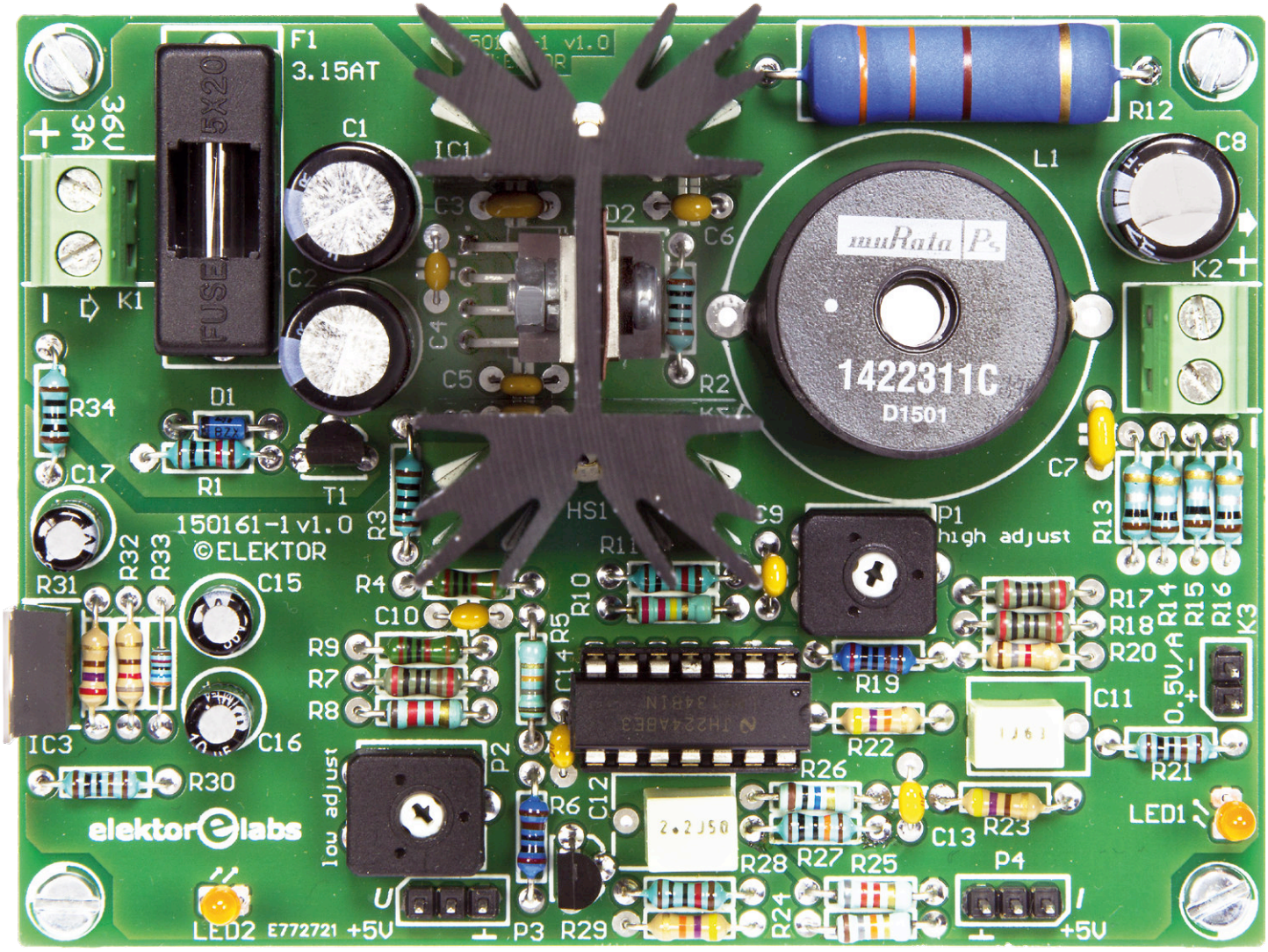
Figure 7. Screen shot of the I/O Card Explorer app for the project.

UniLab Mk. 2

0–30 V, 3 A compact switch-mode lab power supply

By **Ton Giesberts** (Elektor Labs))

Adjustable power supplies are popular DIY projects. Over the years Elektor has designed and published hundreds of circuits for all sorts and sizes of lab power supplies. One of the most popular power supplies in recent years was the UniLab from 2010, a compact unit with a foolproof design for DIY construction. Six years down the road it's time for a successor with the same good features: the UniLab Mark 2.



Specifications

- Output voltage: 0–30 V
- Max. output current: 3 A (voltage limiting at 3.2 A)
- Voltage variation from no load to 3 A: 0.17 V
- Switching frequency 260 kHz
- Maximum ripple (260 kHz): 120 mV_{pp}
- Nominal input voltage: 36–38 V
- Automatic output shutdown when V_{in} < 34.2 V
- Efficiency at 30 V / 3 A out: 93%
- Efficiency at 5 V / 3.1 A out: 80%
- Compact PCB layout

In the April issue of 2010 we published a power supply project under the name UniLab [1] — a compact design with a rated output of 30 V at 3 A. It was an amazingly popular project, and a lot of them were built by electronics enthusiasts. In November of the same year we published a follow-up article [2] that described how to build a dual lab power supply with two UniLab boards and a voltage/current display specifically developed for this purpose.

The new design presented here can be regarded as a successor to the UniLab. The maximum output voltage is again 30 V, and the maximum output current is a bit more than 3 A (up to 5 A is possible with some restrictions). For the new design we opted for a different step-down converter IC, in this case the LM2677-ADJ. It has a higher switching frequency (260 kHz) than the LM2576-ADJ used in the original design (52 kHz), which makes a smaller output filter possible. The internal architecture of the LM2677 (see the block diagram in **Figure 1**) is also completely different from that of the LM2576.

The circuit

The basic approach with this circuit is the same as with the original UniLab. Here again we opted for a switch-mode controller to keep the dimensions of the overall circuit as small as possible. Compared to a linear regulator, this also has the advantage that the efficiency is very high even at low output voltages and the heat sink can be kept fairly small.

The standard application circuit for the LM2677-ADJ has only a few additional components: a diode, an inductor, some capacitors, and a pair of resistors to set the output voltage (see **Figure 2**). A drawback of this configuration is that the minimum output voltage cannot be lower than 1.21 V, which is the value of the internal reference voltage — a problem we also encountered with the original UniLab. For a general-purpose power supply that we would like to have on our bench, it's nice to be able to turn the output voltage all the way down to 0 V. In the original UniLab design we solved the problem by using a negative auxiliary supply voltage, but this time we took a different approach.

The schematic diagram in **Figure 3** may appear rather complicated at first glance, but it's actually quite simple once you understand it. The core of the circuit consists of the buck regulator IC1 with storage inductor L1 and flyback diode D2. The output voltage is adjusted by applying the voltage on the wiper of P3 to the feedback pin of the regulator via IC2B. Here IC2B allows the output voltage to be adjusted down to 0 V, as described in more detail further on. Sense resistors R13–R16 in the negative supply line are used to measure the output current. IC2A amplifies the voltage over these resistors, and IC2D compares the voltage at the output of IC2A to the voltage set with P4, so the position of this potentiometer determines the maximum output current. If the measured output current is greater than allowed, IC2D drives T2 into conduction to reduce the output voltage by lowering the voltage from P3.

For IC2 we chose an LM6134, a fast (10 MHz) rail-to-rail in-out opamp with a supply voltage range of 1.8 to 24 V. IC2 is powered from a separate 5 V supply which is independent of the set output voltage. This supply is built around an LM317 voltage regulator. The 5 V supply voltage is also used as a reference voltage for potentiometers P3 and P4. R34 and C17 suppress any noise on the input voltage line. The LM317 can tolerate voltages up to 40 V between the input and the output, so it has no difficulties with an input voltage of about 36 V. LED2 acts as a power indicator.

To avoid strange effects when the power supply is switched off, the D1/T1/R1 network disables regulator IC1 when the input voltage drops below 34 V. This works as follows: When the gate-source voltage of the p-channel JFET T1 is 0 V or very low, the JFET conducts and dis-

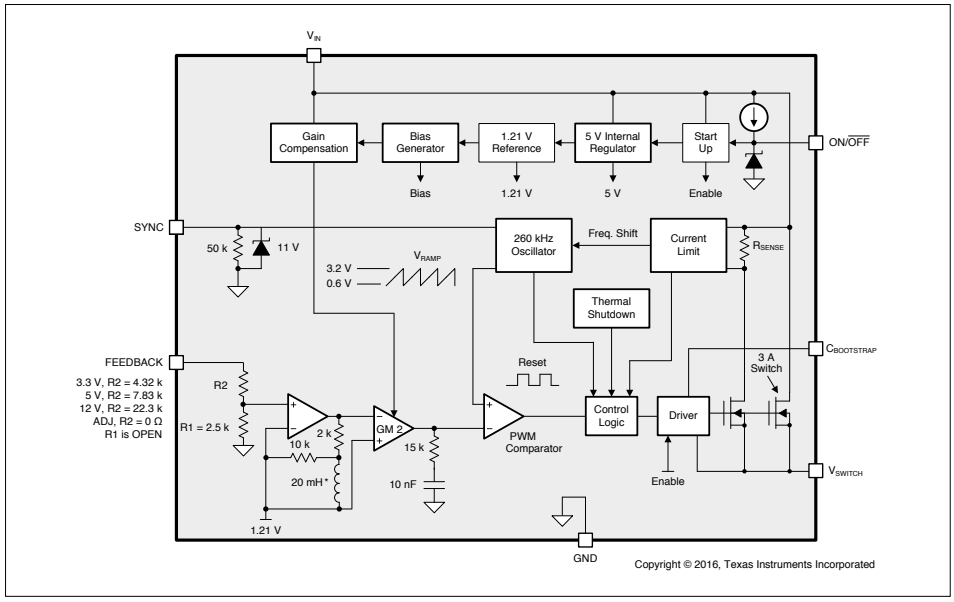


Figure 1. Block diagram of the LM2677 from the TI Simple Switcher series. (source: Texas Instruments)

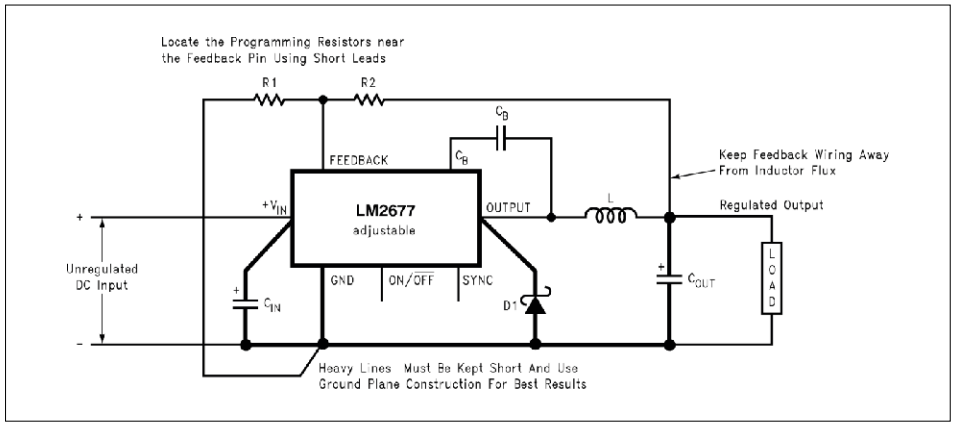


Figure 2. The standard application circuit from the data sheet consists of just a few components. Resistors R1 and R2 determine the output voltage. (source: Texas Instruments)

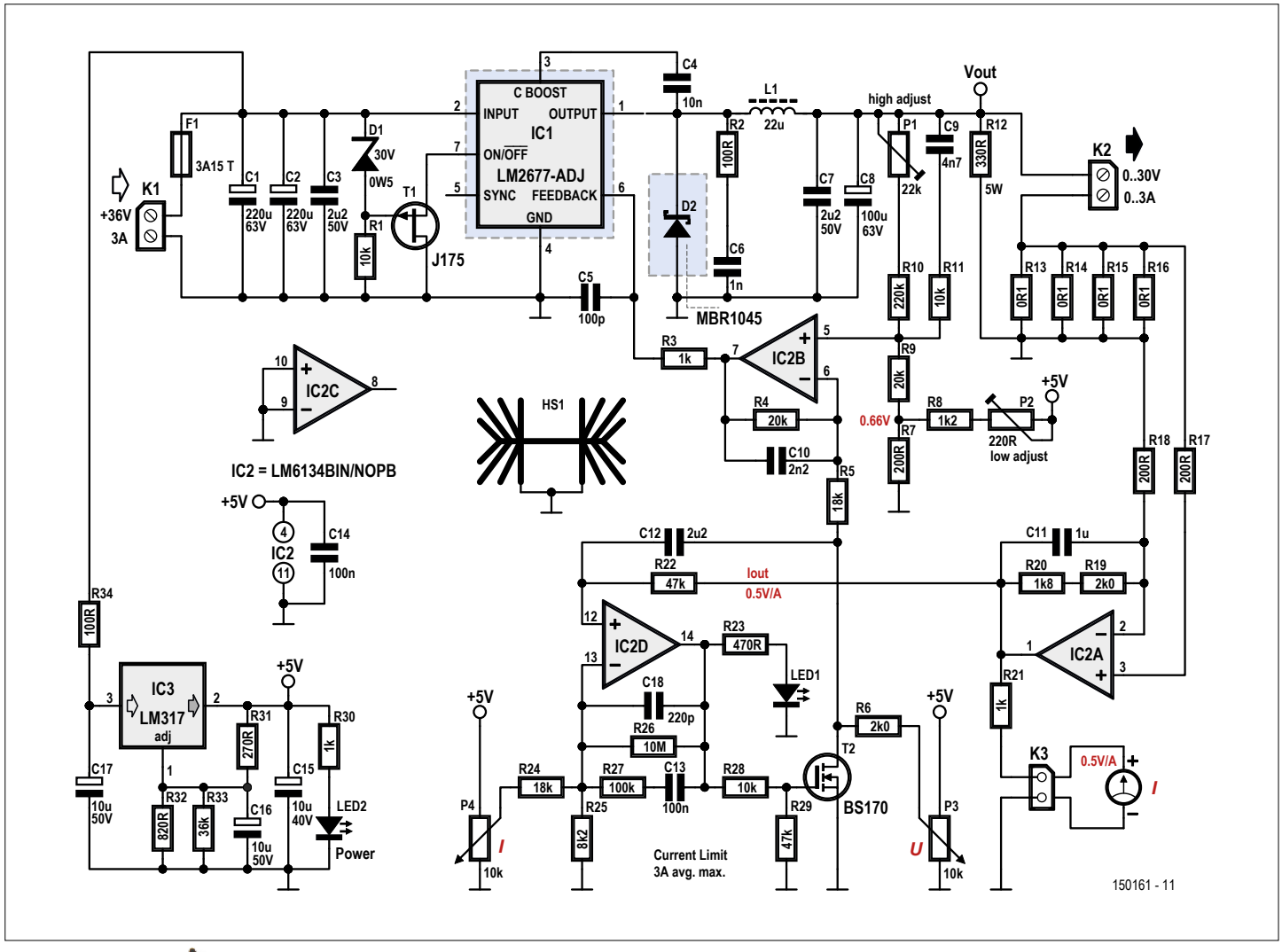
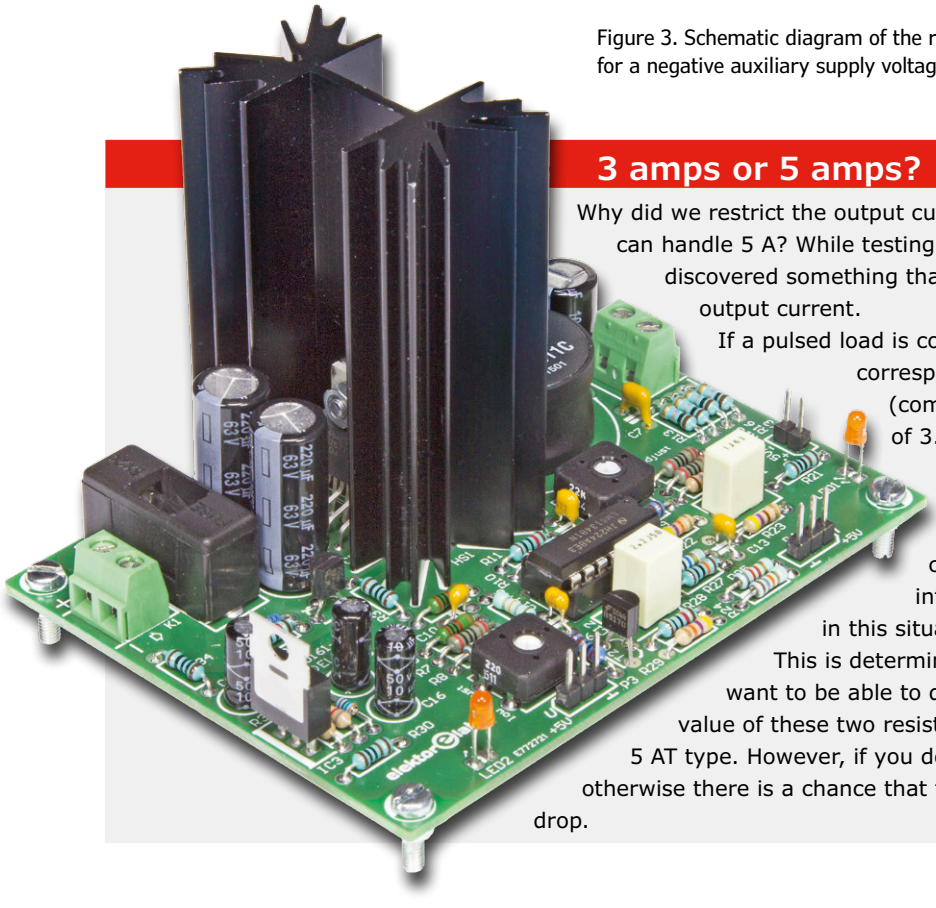


Figure 3. Schematic diagram of the revamped UniLab. Smart feedback circuitry eliminates the need for a negative auxiliary supply voltage to allow the output voltage to be adjusted down to 0 V.



3 amps or 5 amps?

Why did we restrict the output current of this circuit to 3 A when the voltage regulator can handle 5 A? While testing the first prototype of this power supply, we discovered something that made us decide not to go for the maximum possible output current.

If a pulsed load is connected to the output with a pulse frequency corresponding to the resonant frequency of the output filter (composed of L1, C7 and C8, with a resonant frequency of 3.36 kHz), it turns out that the current through the inductor can be 1.5 times the current flowing through the pulsed load. With load pulses of 3 A, this means that the voltage regulator is supplying current pulses of nearly 5 A. To avoid having the internal current limiting of the LM2677 come into play in this situation, we set the maximum output current to 3 A. This is determined by the values of R24 and R25. If you nevertheless want to be able to draw 5 A from the power supply, you can change the value of these two resistors to 10 kΩ. You must also replace fuse F1 with a 5 AT type. However, if you do this you must be careful with large pulse loads, as otherwise there is a chance that the output voltage will become unstable or suddenly drop.

ables the output of regulator IC1 by pulling pin 7 low. It remains disabled as long as the voltage on this input is lower than 0.8 V (pin 7 has a 20 µA internal pull-up current source). When the input voltage rises above 30 V, Zener diode D1 starts to conduct. If the voltage rises by another volt or two, the gate-source voltage over resistor R1 rises by the same amount, eventually reaching a level high enough to cut off the JFET (3 to 6 V). Then the output of IC1 is enabled.

The most striking change in this design compared to the original UniLab design is the altered feedback circuitry for IC1. To allow the output voltage to be adjusted down to 0 V, we have to make the voltage at the feedback input of IC1 (pin 6) match the internal reference voltage of 1.21 V. For this purpose we use a DC gain stage in the form of IC2B. The combination of voltage regulator IC1 and opamp IC2B can be regarded as a power opamp with pins 5 and 6 of IC2B as its inverting and non-inverting inputs, respectively. A control voltage in the range of 0 to 5 V from P3 is applied to the inverting input of IC2B, which is the non-inverting input of our imaginary power opamp. The output voltage of IC1 (after L1) is fed back through a voltage divider to the non-inverting input of IC2B. The gain of IC2B is -1 for the inverting input, but the attenuated output voltage on the non-inverting input is amplified by a factor of 2. With regard to the dimensioning of voltage divider P1/R10/R9, this means that with an output voltage of 30 V we need a voltage of 2.5 V on pin 5 of IC2B, so the resistance of P1 plus R10 must be about 11 times the resistance of R9.

As previously mentioned, the feedback signal must be slightly above 1.21 V to allow the output voltage to be adjusted down to 0 V. This is made possible by the additional voltage divider P2/R8/R7. It supplies a voltage of about 0.66 V to pin 5, which results in 1.21 V after amplification by a factor of 2 — exactly the same as the reference voltage. The values of P2, R8 and P7 are relatively low compared to the value of R9, so any change in the current through R9 has little effect on the voltage over R7. The voltage over R7 is negligible at relatively high output voltages. All this means that the output can be adjusted to 0 V with P2 and to the maximum output voltage with P1.

To keep the overall circuit as low-cost as possible, the current sense resistor

Component List

Resistors

Default: 5%, 0.25 W
R1,R11,R28 = 10kΩ
R2,R34 = 100Ω
R3,R21,R30 = 1kΩ
R4,R9 = 20kΩ, 1%
R5,R24 = 18kΩ
R6,R19 = 2.00kΩ, 1%
R7,R17,R18 = 200Ω, 1%
R8 = 1.2kΩ
R10 = 220kΩ
R12 = 330Ω 5%, 5W
R13,R14,R15,R16 = 0.1Ω, 5% 0.5W
R20 = 1.8kΩ
R22,R29 = 47kΩ
R23 = 470Ω
R25 = 8.2kΩ
R26 = 10MΩ
R27 = 100kΩ
R31 = 270Ω
R32 = 820Ω
R33 = 36kΩ, 1%
P1 = 22kΩ trimpot, horizontal
P2 = 220Ω trimpot, horizontal
P3,P4 = 10kΩ potentiometer, linear

Capacitors

C1,C2 = 220µF 63V, radial, 5mm pitch, 10 mm diam.
C3,C7 = 2.2µF 50V, ceramic Y5V, 0.2'''pitch
C4 = 10nF 100V, ceramic X7R, 0.2'''pitch
C5 = 100pF 100V, ceramic h C0G/NP0, 0.2'''pitch
C6 = 1nF 100V, ceramic X7R, 0.2'''pitch
C8 = 100µF 63V, radial, 5mm pitch, 10mm diam.
C9 = 4.7nF 50 V, ceramic X7R, 0.2'''pitch
C10 = 2.2nF 100V, ceramic X7R, 0.2'''pitch
C11 = 1µF 63V, 5%, MKT, 5 or 7.5mm pitch
C12 = 2.22µF 50V, 5%, MKT, 5 or 7.5mm pitch
C13,C14 = 100nF 50V, ceramic X7R, 5mm pitch
C15,C16,C17 = 10µF 50V radial, 2mm pitch, 5 or 6.3mm diam.
C18 = 220pF 200V, ceramic C0G/NP0, 0.2'''pitch

Inductor

L1 = 22µH, 15%, 11A, 0.014Ω, 23.9 or 29.3mm pitch (e.g. Murata 1422311C)

Semiconductors

D1 = BZX55C30, 30 V, 0.5W zener diode
D2 = MBR1045, TO-220AC
LED1,LED2 = LED, green, 3mm
T1 = J175_D26Z, TO-92
T2 = BS170, TO-92
IC1 = LM2677T-ADJ, TO-220-7
IC2 = LM6134BIN/NOPB, MDIP-14
IC3 = LM317, TO-220-3

Miscellaneous

K1,K2 = 2-way PCB screw terminal block, 0.2'' pitch
K3 = 2-pin pinheader, 0.1'' pitch
P3,P4 = 3-pin pinheader, 0.1'' pitch
F1 = 3,15 AT (slow), with fuseholder and cap, PCB mount, 20 x 5 mm, 500 V
HS1 = heatsink, PCB mount, 2.6K/W, 1 inch mounting pin distance (e.g. Aavid 530002B02500G)
Thermal isolation set for D2, comprising washer and ring (e.g. Multicomp MK3306/S)
PCB # 150161-1

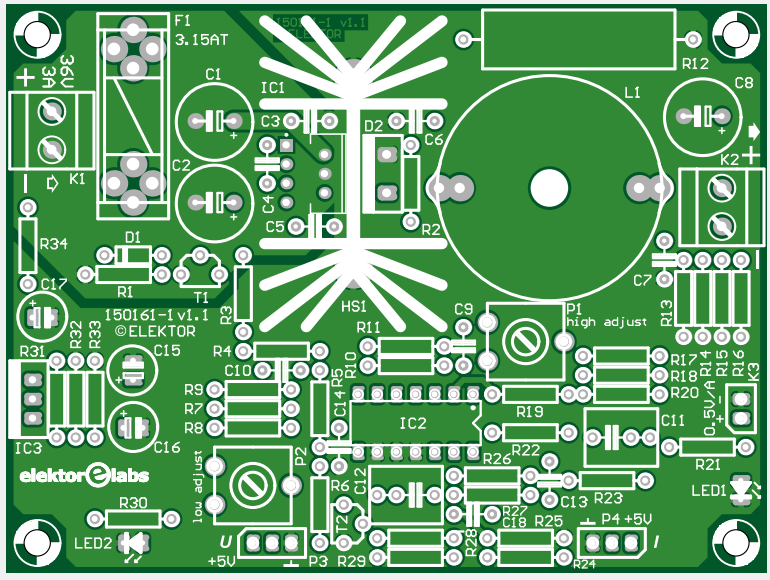


Figure 4. The double-sided PCB is pleasantly compact, and assembly is easy thanks to the use of leaded components.

consists of four 0.1 Ω resistors wired in parallel (R13–R16). They have a tolerance of 5%, which can cause a deviation of a few percent in the sensed voltage. If necessary that could be compensated by using a more sensitive meter with a suitable voltage divider. IC2A amplifies the voltage over the sense resistor by a factor of 20. This opamp has enough headroom to handle current peaks a good deal higher than 3 A — the voltage at the output of IC2A is only 1.5 V with an output current of 3 A.

IC2D compares the current limit setpoint voltage from P4 to the output voltage from IC2A. As we want to restrict the maximum output current to 3 A, voltage divider R24/R25 reduces the maximum voltage from P4 to 1.5 V. The circuitry around IC2D forms a combination of a high-gain amplifier (with the gain determined by R26) and a comparator, since excessive gain could result in unstable current regulation. When the output current exceeds the limit setting, IC2D drives MOSFET T2 into conduction to reduce the voltage from P3. Here R6 prevents T2 from short-circuiting the supply voltage when the wiper of P3 is at the top end. The resistance of R5 plus R6 is equal to the resistance of R4, so the inverting gain of IC2B is exactly -1. LED1 lights up when current limiting is active.

Several other details are worth mentioning: Resistor R12 is an open-circuit load resistor that prevents the output voltage from rising excessively when the output is lightly loaded or not loaded. It also improves regulation stability. However, it increases power dissipation somewhat (2.7 W at 30 V), reducing the overall efficiency. Resistors R11, R22 and R27 and capacitors C5, C9, C10, C11, C12, C13 and C18 are included to improve the stability of voltage and current regulation. Their values were determined empirically, so there may be room for improvement. The values were obtained by observing the output behavior of the power supply with a variable switched load at various switching rates and output voltages. Resistor R3 is placed between the output of IC2B and the feedback input of IC1 because it is probably not a good idea to connect the output of an opamp directly to the feedback input of a voltage regulator. It also allows C5 to be added to the circuit. Finally, resistors R2 and R6 suppress noise pulses from switching.

The input voltage range of the circuit is 36 to 38 V. You can use a switching power supply as the power source. They are readily available and inexpensive (for example, from Mean Well). Compared to a heavy-duty power transformer with a rectifier and filter capacitor, a switching power supply is less expensive, has much lower ripple, and is usually more compact.

Construction and adjustment

As you might expect, we also designed a PCB layout for the UniLab Mk. 2, which is shown in **Figure 4**. You can buy the board from the Elektor Store, but the layout files are also available at [3] if you want to make it yourself. All of the components are leaded types, so assembly is not difficult. Voltage regulator IC1 and diode D2 are mounted on the same heat sink, which is also mounted on the PCB. Note that D2 must be insulated from the heat sink by using a thermal tab and an insulating washer, as indicated in the components list. Apply a bit of thermal grease to IC1, D2 and the thermal tab (not too much, as otherwise you do more harm than good). Secure IC1 and D2 with a single M3 x 10mm screw with a nut and lock washer. Mount R12 spaced a bit above the PCB. Voltage regulator IC3 does not require a heat sink. Unlike the original UniLab design, the two potentiometers for adjusting the output voltage and current (P3 and P4) are not mounted directly on the board, but instead connected with short pieces of stranded wire. That gives you more freedom for fitting the circuit in an enclosure. Keep the connecting wires as short as possible. When installing the board in the enclosure, make sure that the rear of the LM317 (IC3) does not touch any metal parts of the enclosure, since it is internally connected to the 5 V output.



As previously mentioned, you can use a standard switching power supply to provide the input voltage for the circuit. For our prototype we used a 200-watt PSU from Mean Well (type RSP-200-36). It can supply 5.5 A, which means it is also suitable if you want to modify the circuit for a maximum output current of up to 5 A (see the “3 amps or 5 amps?” inset). You can add two voltmeter modules to display the output voltage and current. If necessary, you can use voltage dividers to obtain accurate indications.

After assembling the board and mounting it in an enclosure, you have to adjust the minimum and maximum voltage settings, but that is fairly easy. First set the voltage potentiometer P3 to zero, set P4 to maximum, and connect a 12 Ω / 5 W load resistor to the output. Slowly turn P2 from the zero position until the voltage just starts to increase. The minimum voltage will be about 10 mV at this point. Disconnect the load resistor; the minimum output voltage will rise to about 0.3 V. Then set P3 to maximum and adjust the maximum output voltage to 30 V with P1. That’s it!

We hope you have a lot of fun with your new universal lab power supply.

(150161)

Web Links

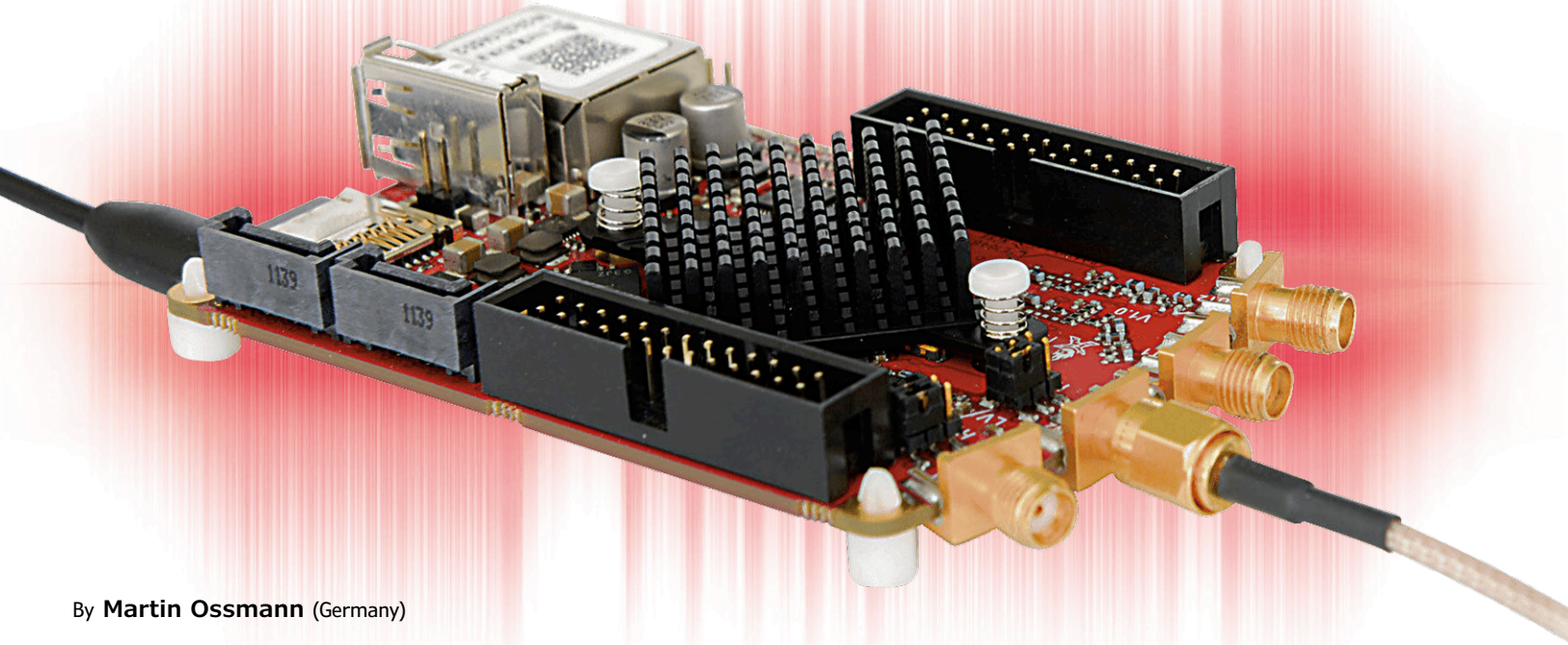
[1] www.elektormagazine.com/090786

[2] www.elektormagazine.com/100529

[3] www.elektormagazine.com/150161

Filtering on the Red Pitaya

Part 2: IIR filters



By **Martin Ossmann** (Germany)

In the first part of the series we got to know the FIR filter, which has the disadvantage that it becomes rather cumbersome when we want to implement a filter with a sharp frequency response. More frequently filters with demanding specifications like this are implemented as IIR (infinite impulse response), or 'recursive', filters. Here we will look at how IIR filters can be designed and implemented in FPGAs.

The structure of an IIR filter is shown in **Figure 1**. The left-hand half forms an FIR filter, and this is followed, on the right, by a filter with feedback 'wires'. One disadvantage of this filter is that it does not have a linear phase response (in other words, its group delay is not constant). The filter is often not implemented in the form shown, but rather as a number of second-order IIR filters (called 'biquads') connected one after the other.

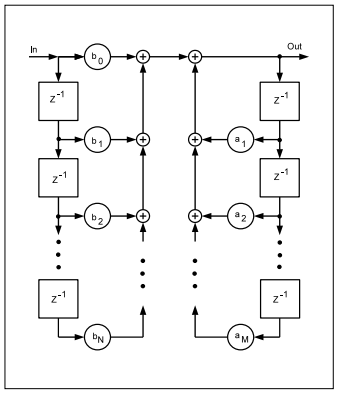


Figure 1. An IIR filter.

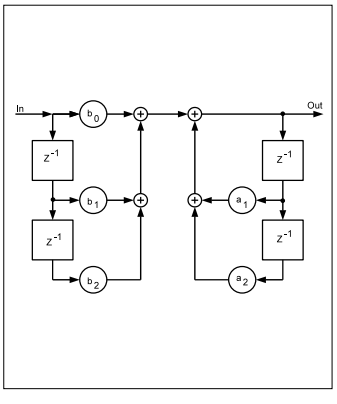


Figure 2. An IIR biquad stage.

Biquads

The circuit of a biquad stage can be seen in **Figure 2**. In the world of analog electronics a biquad corresponds to a resonant circuit such as an LC-network or an opamp circuit with two energy storage elements.

Figure 3 shows the impulse response (the output when presented with the input sequence ..., 0, 0, 0, 1, 0, 0, 0, ...) of a biquad stage: the similarity of the the waveform to that produced by a resonant circuit is clear. Biquad filter stages are indeed also used in analog circuit technology when it is required to construct a complicated filter.

If we want to implement the biquad shown in Figure 2 in an FPGA we will need five multipliers and four adders, plus four registers to provide the signal delays represented by z^{-1}. Each of these elements will have to be wide enough to provide the desired accuracy. In our implementation we work with signals represented as 25-bit words, while the filter coefficients are 18 bits wide. This makes maximum use of the multiplier units in the Red Pitaya's FPGA. Most design programs for IIR filters can deliver their output in the form of coefficients for biquad stages.

Need for speed

The length of the critical path is what determines the maximum possible sample rate at which a filter can operate. In this case the critical path will be a signal path between two registers

clocked at the sample rate. The design of Figure 2 has a path running though one multiplier and three adders; we might wonder if it is possible to improve on this.

The filter structure shown in **Figure 4** has exactly the same behavior as the one shown in Figure 2, as you can verify using a bit of mathematics. We have added extra delay to the horizontal path, but this does not affect the frequency response. In this 'transposed direct form I' the critical path now runs through one multiplier and just two adders instead of three, and is hence shorter. We will therefore adopt this structure.

In this version a biquad requires five multipliers, four adders and five intermediate registers. The Verilog code (with an additional output register called ykReg) is shown in **Listing 1**.

Programming and debugging

Although this piece of code might look very short and simple, it nevertheless required considerable expenditure of effort to optimize it to make it work reliably and efficiently with the DSP48 slices in the FPGA. Various techniques can be employed to test the code. For example, I simulated the circuit with the help of the Icarus Verilog simulator [1], using a filter whose first few output values are all integers. The parameters and the results are shown in **Listing 2**.

I also tried synthesizing and implementing a number of variations on the design using the Xilinx Vivado [2] FPGA development environment, with the aim of determining whether Vivado could in fact find a good implementation. I also tried a number of partial circuits to check that they could operate in real time with the A/D and D/A converters on the Red Pitaya board to provide analog input and output. I used programs written in C to insert test values in order to verify the operation of the logic in the FPGA under controlled conditions.

The circuit diagram of the implementation in Vivado is shown in **Figure 5**. In fact the figure shows a simplified implementation with a word length of 11 bits. The five large yellow blocks are DSP48E1 slices, implementing the five multiplier units. The three columns each containing eleven small yellow rectangles are the 11-bit-wide registers, comprising a total of 3 x 11 = 33 flip-flops. The four square groups of modules are 11-bit adders, each formed from eleven LUTs and three carry-chain blocks: this is how the four adders with an 11-bit word length are implemented in the FPGA. It is noteworthy that only three registers are implemented with flip-flops, whereas the design called for five registers plus an output register. The answer to this is that three of the six registers are implemented within the DSP48 slices.

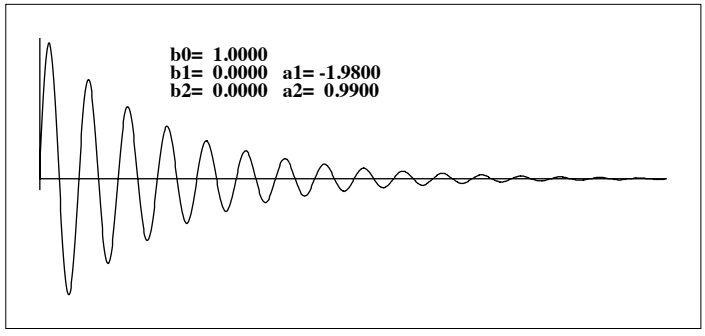


Figure 3. Impulse response of a biquad stage.

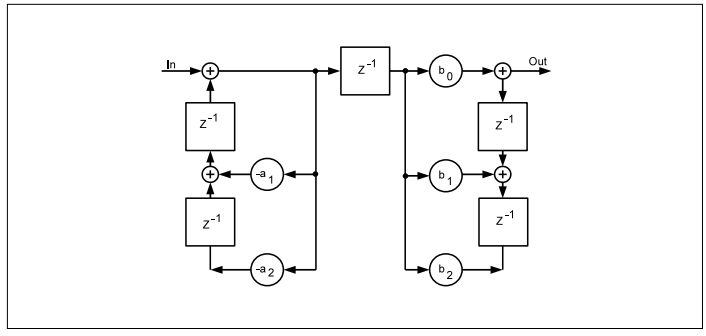


Figure 4. Alternative biquad implementation.

Listing 1. Biquad example code in Verilog.

```
assign vk = IIRinput_i+s1 ;
assign productA2 = NEGa2_i*vk ;
assign productA1 = NEGa1_i*vk ;
assign productB0 = b0_i*vkReg ;
assign productB1 = b1_i*vkReg ;
assign productB2 = b2_i*vkReg ;
assign IIRoutput_o = ykReg ;

always @( posedge IIRclk_i ) begin
  if ( sampleStrobe ) begin
    s1 <= s2+ productA1[25-1+16:0+16] ;
    s2 <= productA2[25-1+16:0+16] ;
    v1 <= v2 + productB1[25-1+16:0+16] ;
    v2 <= productB2[25-1+16:0+16] ;
    vkReg <= vk ;
    ykReg <= v1 + productB0[25-1+16:0+16] ;
  end
end
```

Listing 2. Test output of a biquad stage.

```
Parameters:
b0= 1.0   b1= 2.00   b2=-1.00   a1=-1.00   a2= 0.50

Results:
  K          input          output
  0            0.0            0.0
  1          512.0          512.0
  2            0.0         1536.0
  3            0.0          768.0
  4            0.0            0.0
  5            0.0         -384.0
  6            0.0         -384.0
  7            0.0         -192.0
  8            0.0            0.0
  9            0.0           96.0
 10            0.0           96.0
 11            0.0           48.0
```

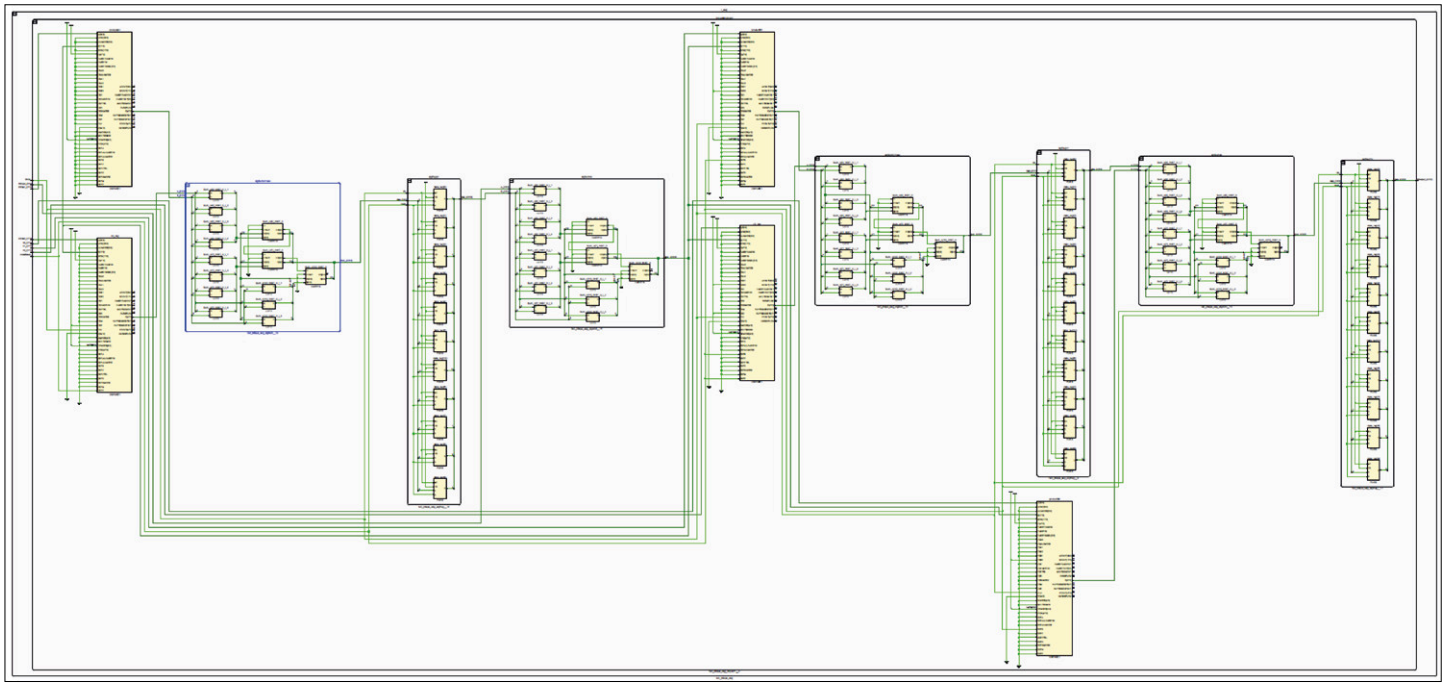


Figure 5. Biquad stage implemented in Vivado.

If we attempt to realize this design with a sample rate of 125 MHz, Vivado will tell us that there is a (tiny) timing violation to the tune of 0.5 ns. Despite this 'overclocking', we have tried the filter and in practice it runs without problems. Our FPGA filter program allows us to use up to six biquad stages; this could be increased, as the FPGA has a total of eighty DSP48 slices available for use.

Filter design

In the above we tacitly assumed that we already knew the coefficients b_0, b_1, b_2, a_1 and a_2 of the biquad stage. In order to calculate these values we can use a filter design program. A simple design program written in C can be run on the Red Pitaya (download available at [3]), which means that the board can implement filters autonomously. Of course it is also possible to use an alternative external design program to compute the coefficients and then transfer them to the Red Pitaya.

Lowpass

We will look first at the design of lowpass filters, as this will form a basis for the development of other types such as highpass and bandpass. The simplest filter that our program can design is a Butterworth lowpass. This is uniquely specified by just the cutoff frequency f_c and the filter order p. The characteristic feature of the Butterworth filter is that its frequency response starts out as flat as possible. The higher the filter

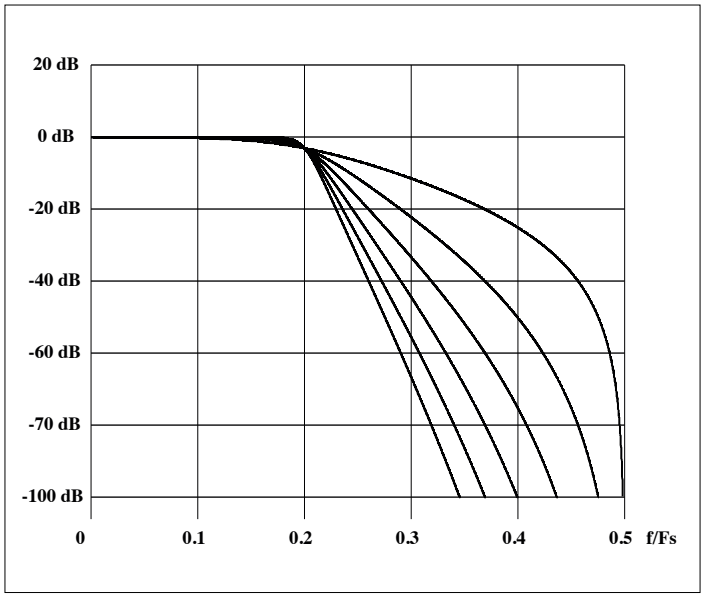


Figure 6. Frequency responses of a range of digital Butterworth filters.

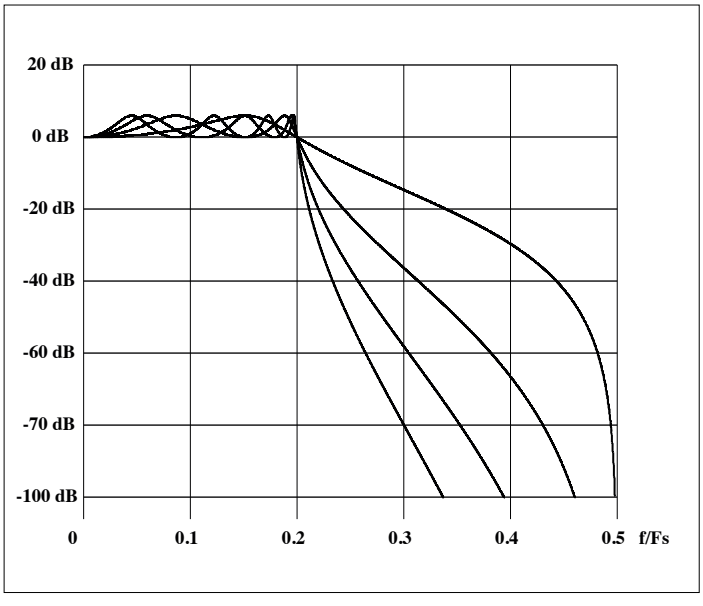


Figure 7. Frequency responses of a range of Chebyshev filters.

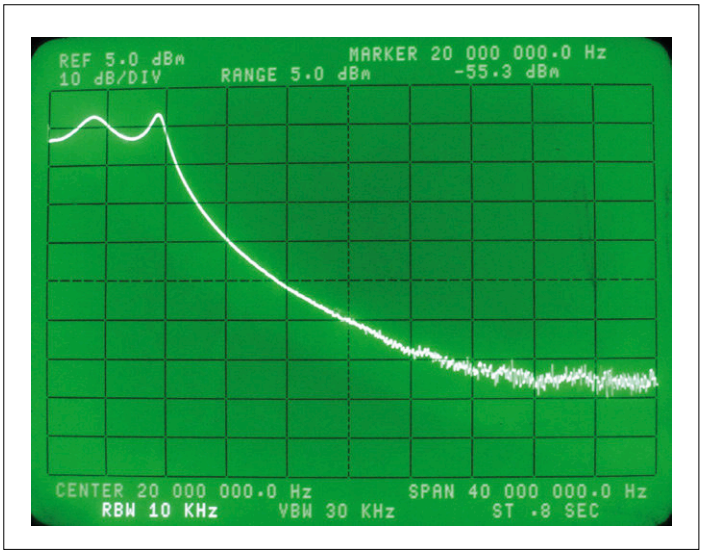


Figure 8. Measured frequency response of a lowpass filter.

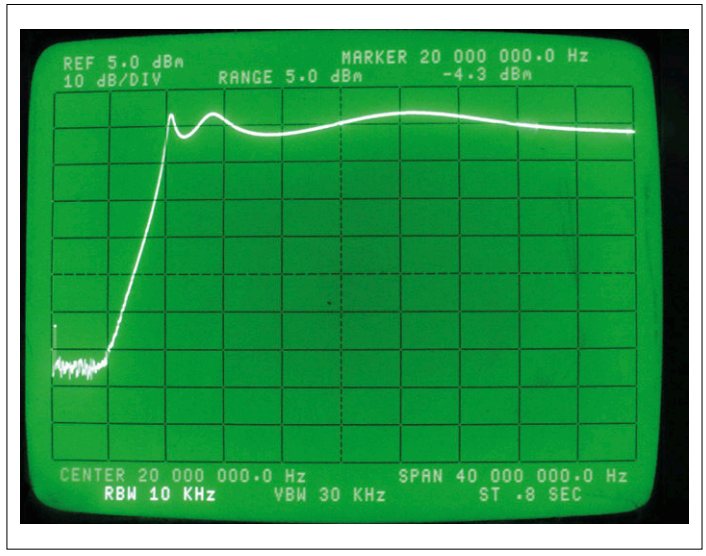


Figure 12. Measured response of a highpass filter.

order, the sharper its cutoff. **Figure 6** shows the frequency response of Butterworth filters with orders p = 2, 4, 6, 8, 10 and 12. The number of biquad stages required, N, is given by $N = p/2$: we only use filters whose order is even.

An alternative lowpass design is the Chebyshev filter. The characteristic feature of this design is that the frequency response falls off as rapidly as possible in the transition band; the price one pays for this is ripple in the passband. The filter is specified by its cutoff frequency, its order and the amount of ripple. **Figure 7** shows the frequency response of some Chebyshev filters of various orders (p = 2, 4, 6 and 8). The narrow transition band can be seen clearly: if a sharp cutoff is required, it will be achieved more easily using this kind of filter.

Figure 8 shows the measured frequency response of a Chebyshev lowpass filter in the range from 0 MHz to 40 MHz. At higher frequencies the attenuation is so great that the signal is lost in the noise of the spectrum analyzer.

Our design program is also capable of implementing elliptic filters. These allow the ripple in the passband to be specified along with the width of the transition band and the minimum permissible attenuation in the stop band. Elliptic filters also have ripple in the stop band: the response does not fall off monotonically, but instead the attenuation oscillates between infinity and the specified minimum value.

Some theory: poles and zeros

The design program creates highpass, bandpass and notch filters by starting from a lowpass design. In order to understand how this works we need a bit of theory. Any filter has an associated (complex) transfer function $H(z)$. This is a rational function and so has zeros z_0, such that $H(z_0) = 0$, and poles z_p, such that $H(z_p)$ is infinite. For any given filter we can therefore draw a pole-zero diagram where we represent the zeros by circles and the poles by crosses. The diagram can help us to understand the response of the filter in qualitative terms.

Now consider **Figure 9**, and imagine a point z moving counterclockwise on the unit circle $|z| = 1$, beginning at $z = 1$ and ending at $z = -1$. As the point moves this corresponds to a frequency *f steadily increasing from $f = 0$ to $f = F_s/2$, half the sample rate. The angle φ between the horizontal axis and the line joining the point to the origin is given by the expression $\varphi = 2\pi f/F_s$*. When the moving point comes into the vicinity of a zero N the frequency response $H(z)$ will be low, and if the point passes exactly through a zero, the response will be exactly zero; in other words, the attenuation at that frequency will be infinite. If the point passes near a pole the response will increase. It is necessary that all the poles lie within the unit circle for the filter to be stable. If the coefficients of the filter must be real, as is normally the case, the poles and zeros must

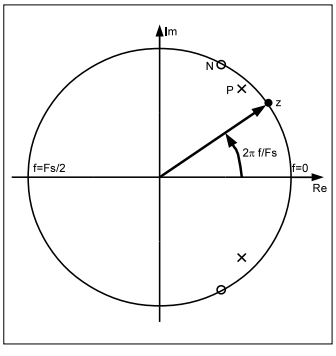


Figure 9. Pole-zero diagram.

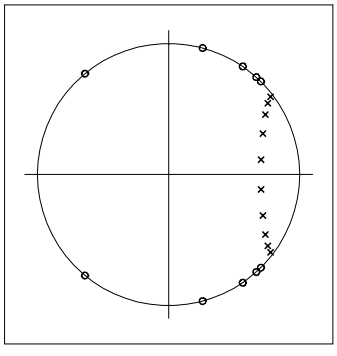


Figure 10. Poles and zeros of an elliptic lowpass filter.

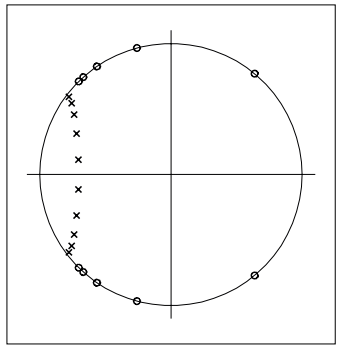


Figure 11. Poles and zeros of a highpass filter.

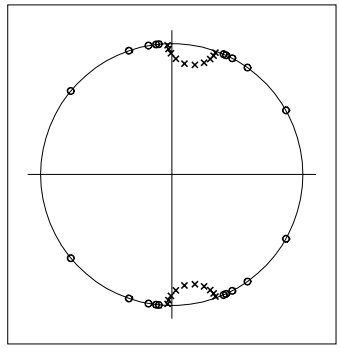


Figure 13. Poles and zeros of a bandpass filter.

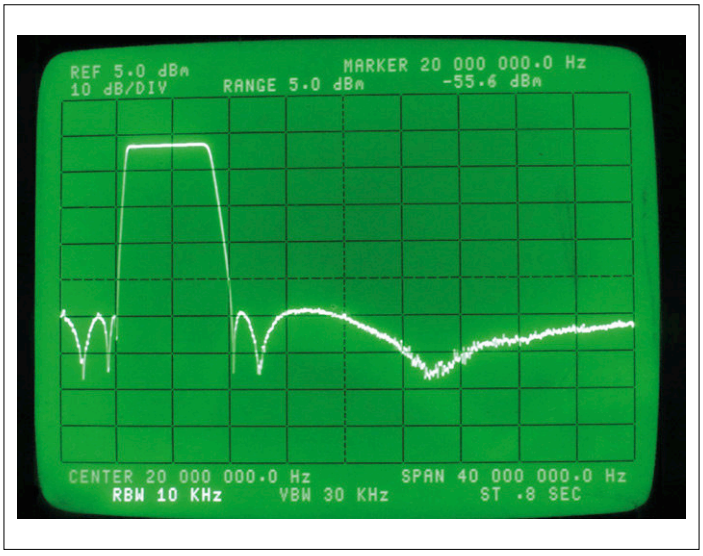


Figure 14. Bandpass filter as implemented.

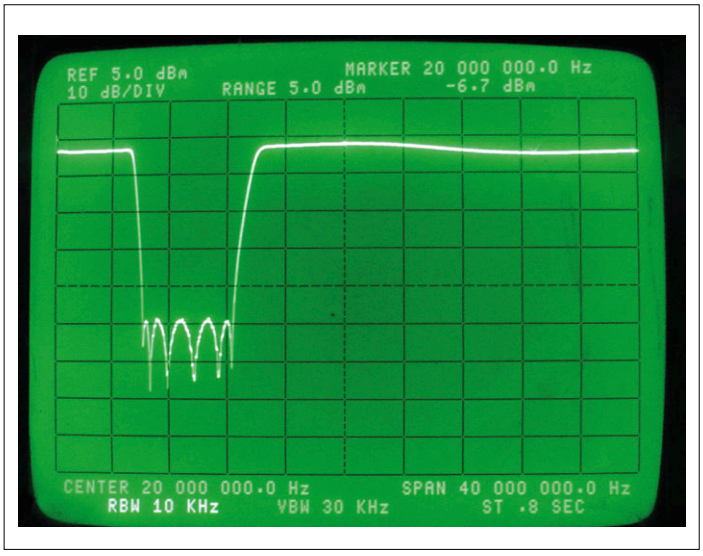


Figure 15. Response of a notch filter.

come in complex conjugate pairs.
Figure 10 shows the pole-zero diagram of an elliptic lowpass filter. As can be seen, at lower frequencies our point is near to poles, and each of these poles can create a bump in the frequency response if it is close enough to the unit circle. At higher frequencies we encounter zeros, all lying on the unit circle, arranged so that in the stop band we are always close to a zero. As a result the frequency response $H(z)$ is small and we have a high degree of attenuation. The filter design program positions the poles and zeros so that the specification for the filter is met. The number of poles or zeros is called the 'order' of the filter; each biquad stage implements two (complex conjugate) poles and two (complex conjugate) zeros.

Highpass

Using this graphical interpretation we can see an easy way to convert a lowpass filter (called the 'prototype' filter) into a highpass filter: we simply reflect the poles and zeros in the vertical axis. The response of the new filter at high frequencies will be the same as that of the prototype at low frequencies, and vice versa. More precisely, the response will be reflected about the frequency $F_S/4$.
Figure 11 shows the pole and zero positions for a highpass filter. All the poles (corresponding to the passband of the filter) are now clustered around the higher frequencies, in the neighborhood of $F_S/2$. **Figure 12** shows the measured response of a Chebyshev highpass filter, illustrating the characteristic ripple in the passband.

Bandpass and notch filters

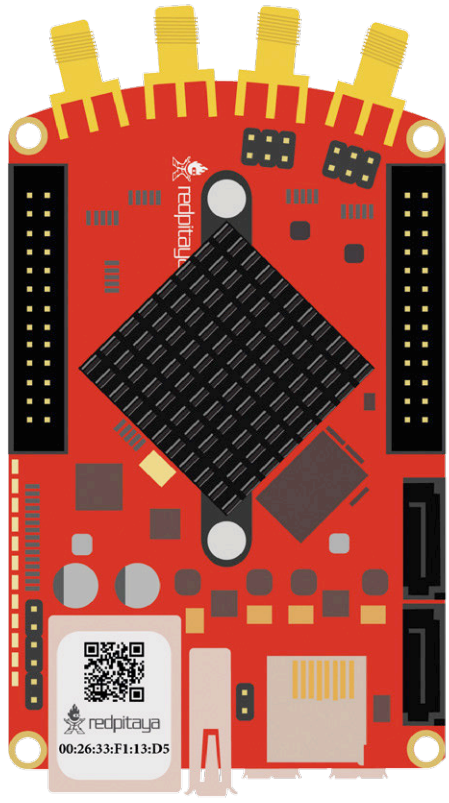
The same principle can obviously be used to construct a bandpass filter. We simply slide the poles around the unit circle to where we want the passband to appear, and slide the zeros to the stopband. Since the poles and zeros must always come in complex conjugate pairs, the result has double the order of the prototype filter. The resulting pole-zero diagram is shown in **Figure 13**. To create a notch filter we simply swap the passband and stopband positions. For these examples we used an elliptic filter as the prototype, with low ripple in the passband.

Figure 14 shows the measured response of a bandpass filter constructed in this way, and **Figure 15** the response of the corresponding notch filter.
With that we have now implemented all the classical filter types used in communications technology, opening up a wide range of new applications for the Red Pitaya board.

(160111)

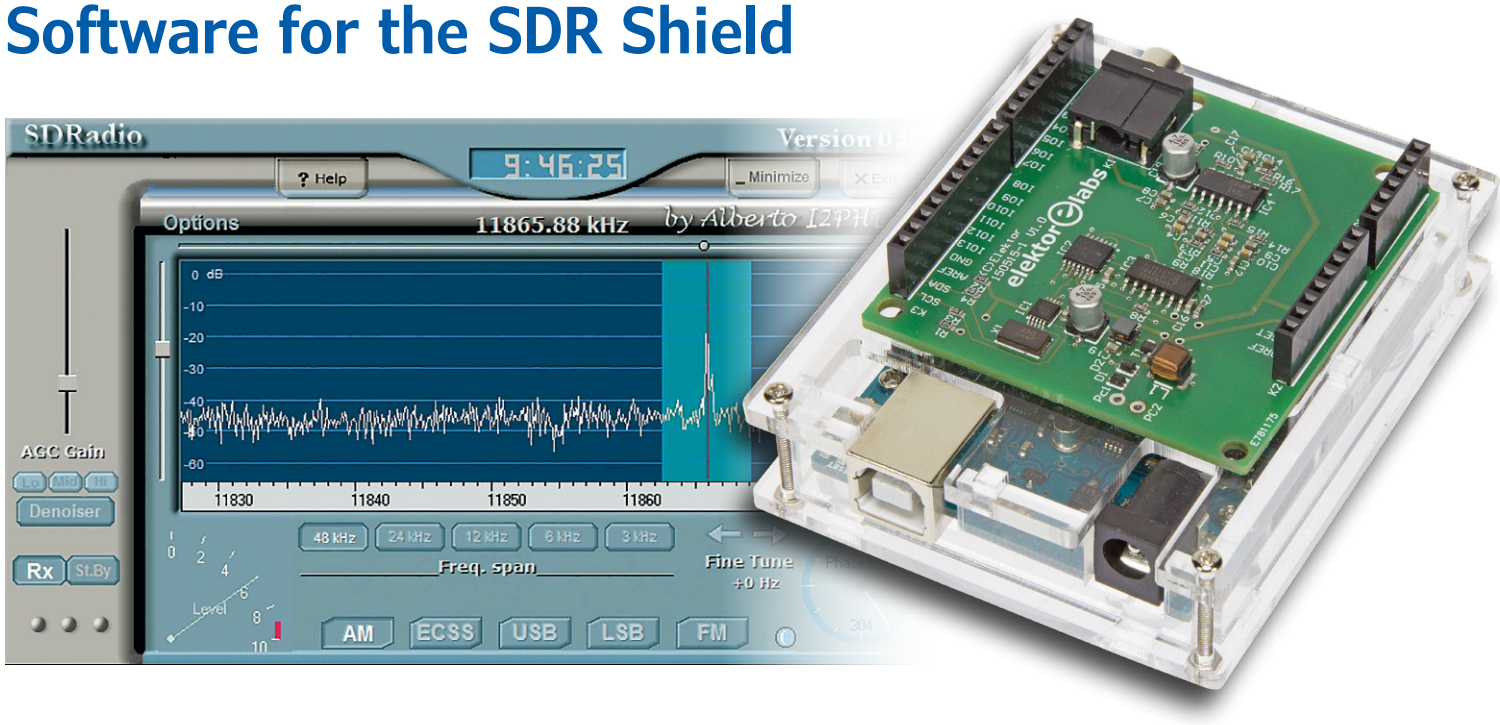
Web Links

[1] http://iverilog.icarus.com/
[2] http://www.xilinx.com/products/design-tools/vivado.html
[3] http://www.elektormagazine.com/160111



Elektor SDR Reloaded (2)

Software for the SDR Shield



Without appropriate software the 'Software Defined Radio' Shield simply cannot function. And there are now so many functions and variations to consider that it's not simple to keep track. Here we detail your first moves when putting the SDR receiver into successful operation.

By **Burkhard Kainka** (Germany)

The Arduino side of the software used was already discussed in the previous edition [2], also how the PC is used to tune the receiver. With the receiver now delivering the IQ signal to the sound card, what happens next depends entirely on which SDR software you use. Numerous programs are available, of which just three will be examined here.

Preparations

In point of fact all you need do is connect a wire antenna to the receiver input. Grounding (earthing) is normally already provided by the ground (earth) wire of the PC. Everything else to do with antennas will follow in a later article. Here and now we'll describe a really simple method for assuring success with your first attempts at reception. It's an attenuator using a potentiometer (**Figure 1**) and this gadget can work wonders. Even a relatively short antenna can in fact, on many bands and at certain times, deliver signals so powerful that the receiver can be overdriven. So you just throttle back the pot a bit and straightaway the reception is crystal-clear again.

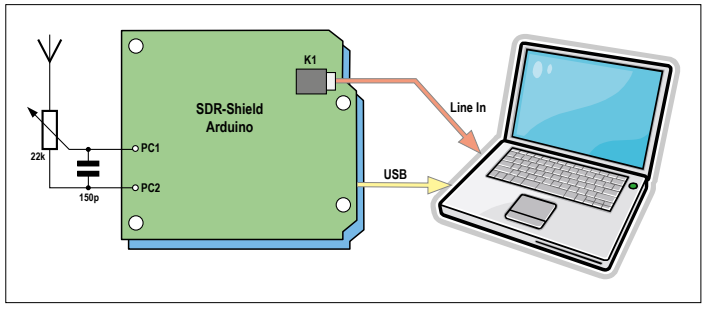


Figure 1. Attenuator and lowpass filter on antenna input.

SDRadio

Everyone who has already experimented with an SDR will have heard of the program SDRadio by Alberto di Bene [1]. Recently the author has written a significantly larger program (HRSDR, see next section), but SDRadio still works extremely well on the newer versions of Windows and stands out for its simplicity. There is no need to install the software as such; it's much simpler to save it in a folder of your choice and just start the program SDRadio.exe. The first thing to do is select the sound card you are using and the audio input (**Figure 2**). If the sound card supports a sampling rate of 96 kHz, you can choose whether to work at 48 kHz or 96 kHz. Clicking on RX starts the decoding.

Along with the SDR software we need to start the tuning program (**Figure 3**), which can be downloaded here [2]. When the VFO in this has been installed successfully on the Shield, the image displayed in SDRadio changes and at the very least you will notice a significant increase in the noise level.

For our first test we select the Long Wave band. The tuning software lets us switch to individual broadcast and amateur

radio bands. A click on LW-MW sets the frequency to 531 kHz. Using the slider control we can tune up or down in 9 kHz channel steps. Clicking on the ends of the slider alters the frequency in small steps of 9 kHz (in the higher bands by 5 kHz), whilst a click on the slider itself gives larger steps of 45 kHz. Because the SDR window displays 48 kHz at a time, this gives us the opportunity to scroll quickly across a band and take an overview to see where the action is.

In this case we are tuning downwards, on order to receive the BBC on Droitwich 198 kHz. The Long Wave has become more interesting now that many local-area Medium Wave transmitters have gone off the air. Another thing is that on Long Wave we have relatively constant propagation conditions during the daytime, meaning that we can expect to find some strong signals. With the antenna potentiometer fully opened up the carrier level reaches -20 dB (**Figure 4**) and the noise floor lies at –55 dB. Reception has relatively severe background noise. If you wind the pot back a bit to an optimum setting, the carrier is inevitably reduced, by 5 dB in fact, but the underlying noise is now –70 dB down (**Figure 5**). In the process the signal-to-noise ratio has been improved from 35 dB to 45 dB, a clearly audible enhancement. Now we can broaden or reduce the receive bandwidth to tease out the best-listening sound. The width of the filtering is indicated by a pale blue stripe. By clicking on the edges you can drag the bandwidth in or out to make it wider or narrower. In fact this ability to vary the bandwidth is a key advantage over traditional receiver concepts using filters of fixed bandwidth.

The BBC signal now occupies 12 kHz. At –3 kHz we see a signal that belongs to Europe 1 on 183 kHz and lies outside the 9-kHz channel spacing. In SDRadio we can switch directly to this frequency and listen in there without altering the tuning. That's the good thing with software defined radio: you can always see what's going on either side of the receive frequency you have set. And if any interfering signals do crop up, you can often remedy this by shifting the receive window slightly.

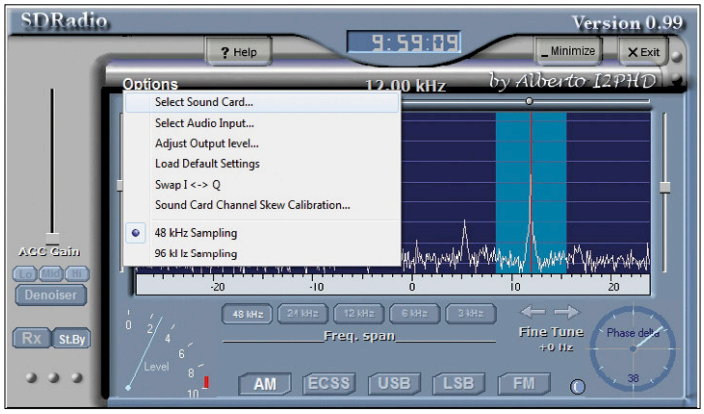
HDSDR

Also from Alberto di Bene is the significantly more potent program HDSDR, downloadable at [3]. The program comes with an installer and the first time you start it, it opens with a window for setting the sound card (**Figure 6**).

Following this we need to select an input. In the Options Menu (F7) we click on *Select Input* and choose the *Sound Card* (**Figure 7**). The *Playback* window appears next, with additional settings (you can, however, close this without making changes). After this we can initiate reception with *Start*. Any time you re-start HDSDR subsequently, the input must be selected afresh.

The program starts initially in the SSB operating mode. So let's click on AM and slide the reception window to 12 kHz. The LO (Local Oscillator, VFO) is set initially at 28.5 MHz and needs to be reset to zero, making the display cover from -24 kHz to +24 kHz. To do this you guide the mouse pointer onto individual figures and change these using the mouse wheel.

An interesting feature is the AGC setting (automatic gain control), for which you can select four settings (Fast, Med, Slow and Off). *Slow* is particularly suitable for AM, although the regulation can react negatively to pulse interference. The Noise



igure 2. SDRadio options.

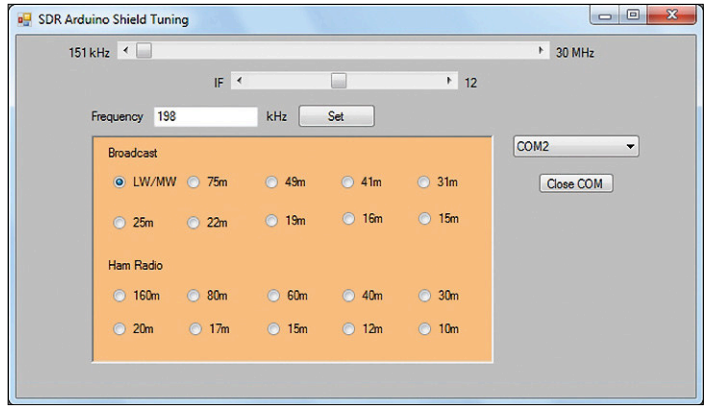


Figure 3. Tuning-in on BBC Droitwich 198 kHz.

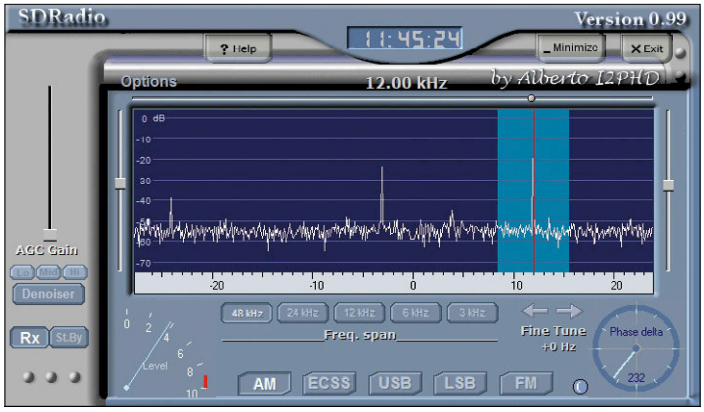


Figure 4. BBC on 198 kHz without attenuator.

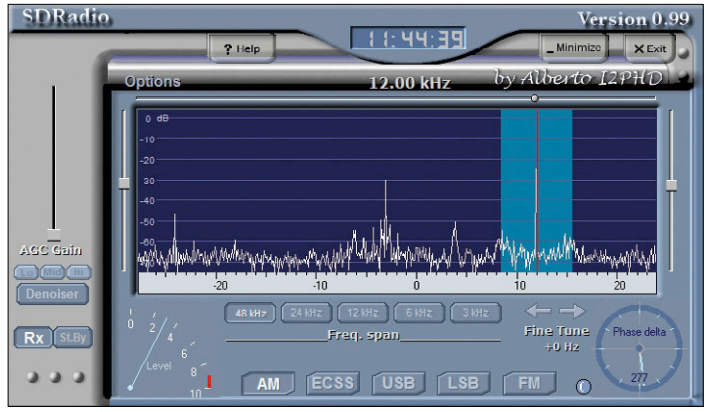


Figure 5. Optimally set attenuator with lowpass filtering.

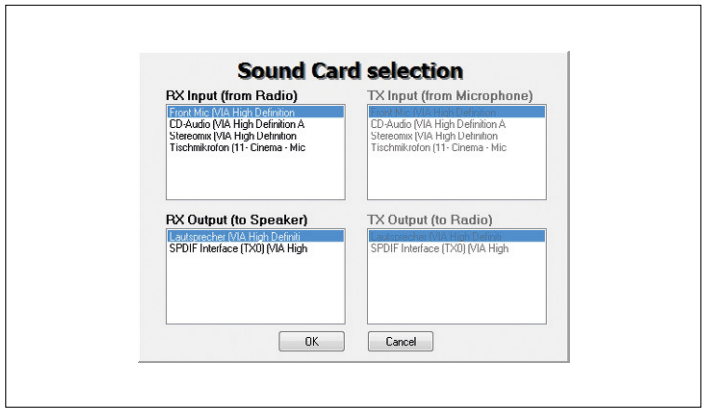


Figure 6. Settings in HDSDR.

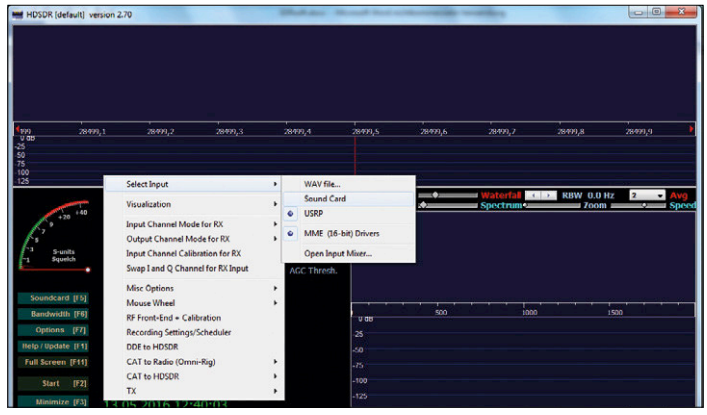


Figure 7. Selecting the sound card as signal input.

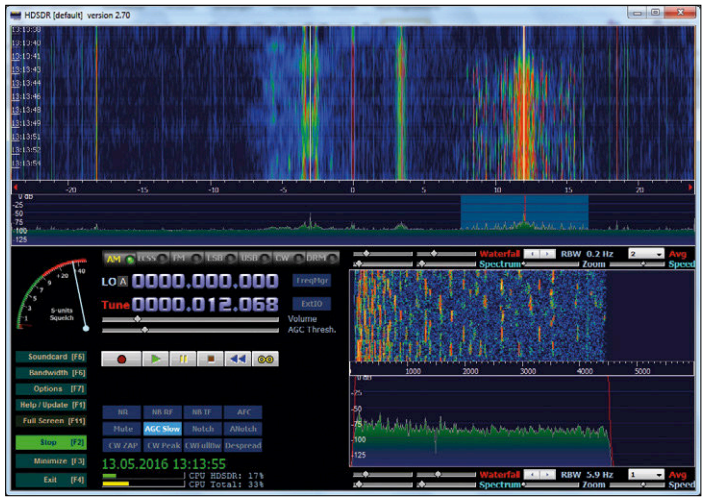


Figure 8. Optimal AM reception.

Blanker (NB RF) can help mitigate this, whilst Noise Reduction (NR) attenuates the noise floor noticeably. When you have found the ideal settings for everything, you can switch out the AGC and thereby achieve a very agreeable, uniform sound.
The program has a window of its own for displaying the demodulated audio signal (**Figure 8**). A thin red line indicates the flanks of the filter. You can adjust these boundaries with the left mouse button and thus alter the bandwidth.
In the evening time the Short Waves come alive and become fascinating. It's well worth wandering around the broadcast bands and seeking out far-off stations of interest. But you can also listen to amateur radio stations and commercial services, where many transmissions are in SSB or CW.

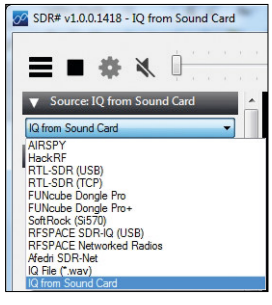


Figure 9. Selecting the inputs.

SDRadio stands out by its simplicity...

SDR-Sharp (SDR#)

SDR# is another piece of software for universal application. The program supports cheapo USB DVB-T dongles and other UHF receivers, and tends to be used mostly on the higher frequencies. However, it is equally suitable for use with the SDR Shield. The software can be downloaded from [4]. At startup you select the hardware to be used along with the sound card as input (**Figure 9**). These settings are retained the next time you use the software.

Decoding begins by using the Start/Stop symbol seen at upper left in **Figure 10**. You have in addition a mute function and the loudspeaker volume control. Using the Menu symbol you can switch the menu on and off, so as to rationalize everything down to the most mission-critical (**Figure 11**). The receive frequency in the 48-kHz range and the bandwidth can also be set then. To later the bandwidth you click using the left mouse button on the edge of the filter and reposition it according to your needs. The maximum AM filter width is 32 kHz, which in fact enables you to hear several adjacent transmitters. This setting makes sense if you want to check out individual Short Wave bands rapidly. During the daytime you may find only gaping emptiness here. But if you keep pressing and maintain the wide AM setting, you can scan across the broadcast bands in rapid 5-kHz steps and find the few signals of interest in a short time. After sunset everything looks entirely different again. It's time now to keep focused on the 41-m band, but also do watch out elsewhere for signals like All India Radio on 7550 kHz.
Among the Menu functions you should examine the AGC settings. All the important functions here can be tried out individually. Normally the Noise Blanker should be deactivated, as it can degrade the sound quality of an AM station. Nevertheless, it can help effectively when you are fighting intensive interference. The same goes for the Noise Reduction options. What would Short Wave be like without amateur radio traffic? On the 40 m band above all others countless SSB speech and CW Morse signals can be found. Normally you would look for an interesting segment using the tuning software and then move the precise receive frequency into the SDR window. In this way you always maintain sight of the current band occupancy. In the CW operating mode (**Figure 12**) you can nominate a very narrow bandwidth in order to suppress close-by signals and fish extremely weak signals out of the noise. A specialized hobby of many radio amateurs is radio contact

using very low transmit power (QRP), often sent at very low speed (QRS) in order that they can still be read among the noise. In the waterfall diagram you can frequently see weak signals that you can scarcely even hear.

SSB speech traffic (**Figure 13**) operates without any carrier, transmitting only one out of the two sidebands. On the lower bands as far as 40 m people use the lower sideband (LSB) and on the upper bands the upper one (USB) is used. To receive these knowing the correct frequency is decisive, since imprecise tuning otherwise leads to a kind of Mickey Mouse voice effect. However, it is very simple to spot the exact position in the spectrum because normally the deeper components of speech predominate. Using the mouse you can adjust the receive window at lightning speed in order to clarify the speech. Once again this is another advantage of being able to see the current band occupancy. In this way you can rapidly flip over to the adjacent station and listen in for a moment. The waterfall also displays clearly those stations that are almost already slipping down in the noise.

Reception software is an extensive field and this article can focus on only the most important programs along with the settings for the most important applications. So far we have covered the crucial first steps but plenty more lies ahead. You need think only about the many digital operating modes like weatherfax, radio teletype (RTTY) and DRM broadcasting. With the Elektor SDR Shield all roads are open.

(160048)

Web Links

[1] http://digilander.libero.it/i2phd/sdradio/

[2] www.elektormagazine.com/150515

[3] www.hdsdr.de

[4] http://airspy.com/download/

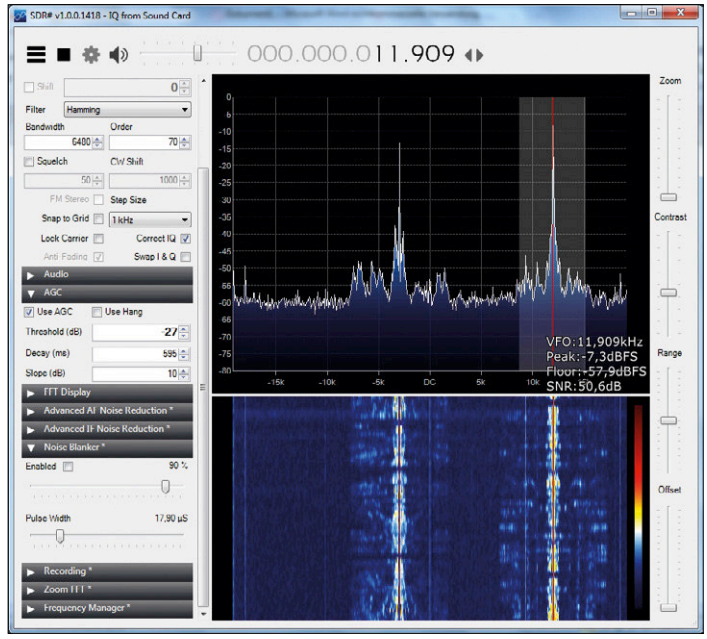


Figure 10. AM reception and SNR display.

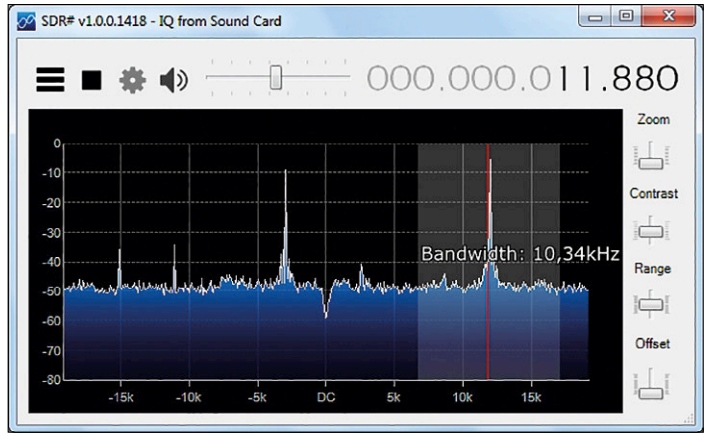


Figure 11. Compact SDR# window.

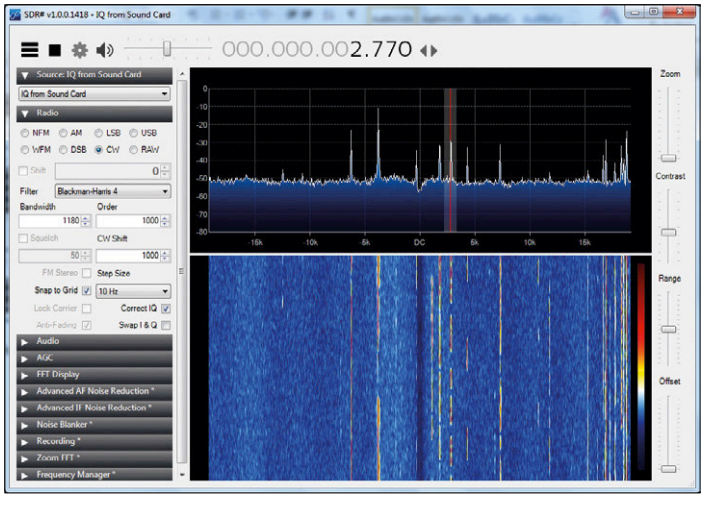


Figure 12. Narrow bandwidth and CW signals.

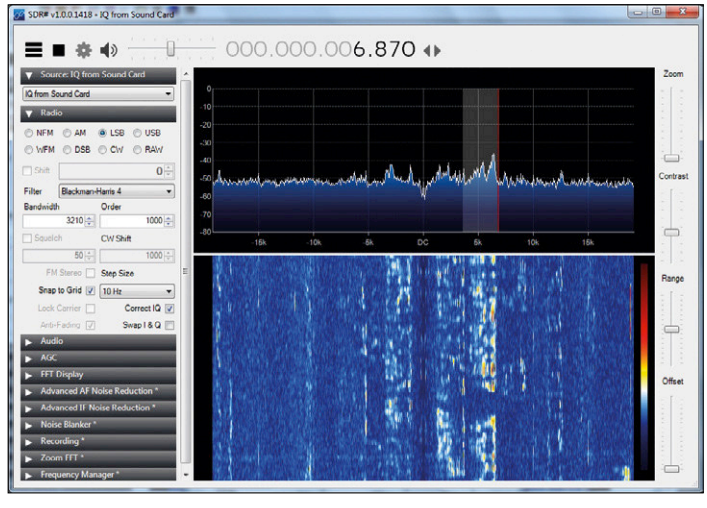


Figure 13. SSB reception on the 40 m band.

LEDitron

A seven-segment display made from LED filaments

By **Ilse Joostens** and **Peter S'heeren** (Belgium)

Conventional low-energy lamps are a dying breed as we have entered the LED era. LEDitron is a seven-segment display module made from LED filaments, similar to a vintage Numitron tube but with better energy efficiency.

Features

- 35 to 100 V adjustable output voltage range
- 100 mA maximum output current
- 12 V / 1 A supply voltage
- Suitable for four to six LED filaments, depending on the brightness setting
- Fully designed with through-hole components

Low-energy lamps first appeared in the 1980s — actually not all that long ago. Unfortunately, the first compact fluorescent lamps (CFLs) were not particularly compact and looked more like a jam jar with a screw base. Fortunately, they became smaller in the course of time. The EU prohibition on incandescent lamps has accelerated the evolution of low-energy lamps, and the first LED lamps – as replacements for CFLs – started to penetrate the market around 2010.

A recent innovation is low-energy lamps with LED filaments, which look more like traditional incandescent lamps (see Figure 1). For nostalgia fans there are even types designed to resemble old-fashioned carbon-filament lamps.

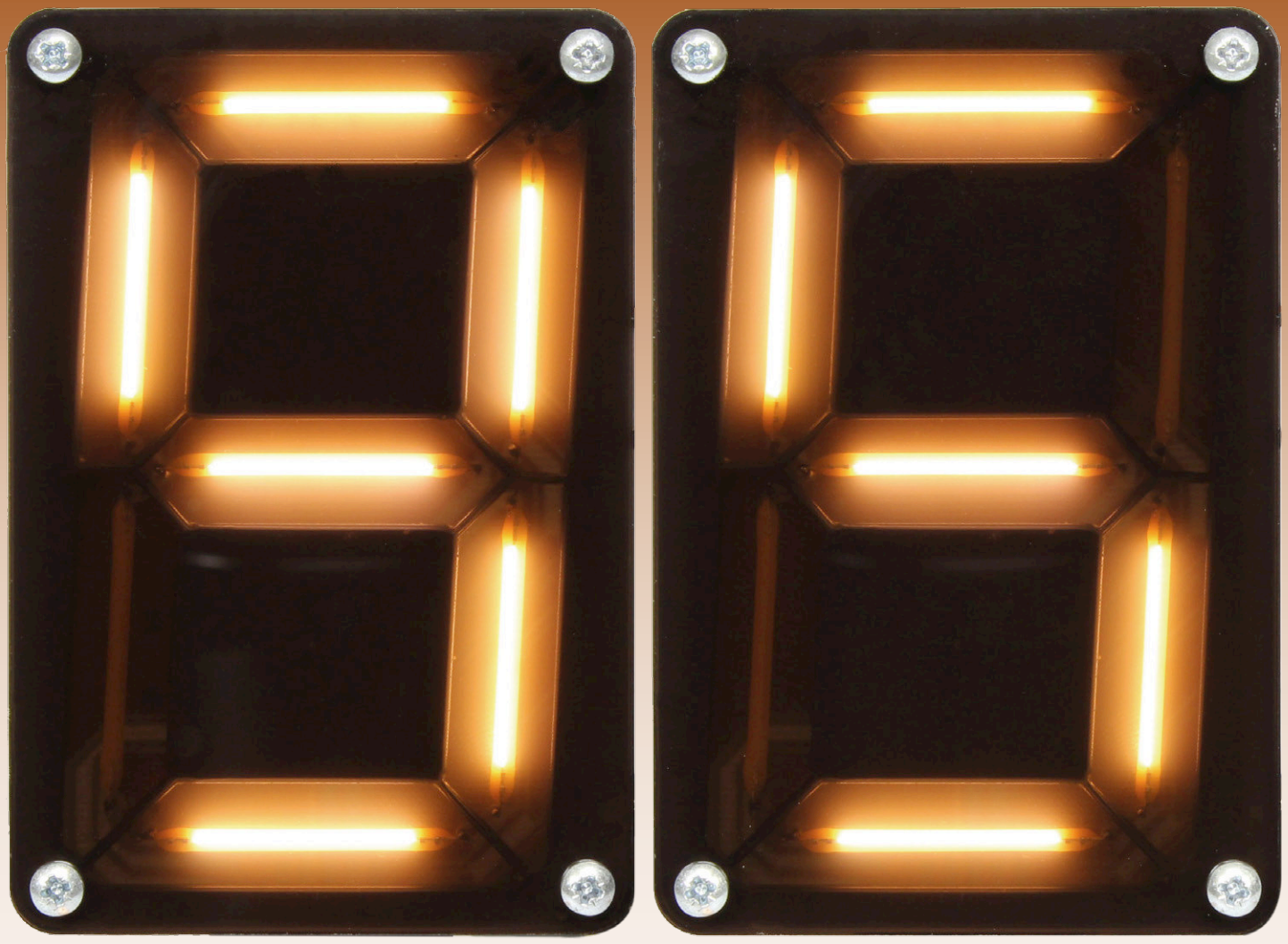
If you know your way around vintage electronics, you are surely familiar with Numitron tubes. They are basically incandescent lamps with several filaments arranged in the form of a seven-segment display. Recently LED filaments have become available as separate items (see Figure 2), mostly from Chinese sellers on eBay, which makes them attractive for experimenting and using for purposes other than originally intended. A disadvantage of Numitron tubes is that they are usually fairly small, which was why the authors decided to build a seven-segment display module with LED filaments as described in this article. The result is not only very attractive, but also highly visible and easy to read, even in direct sunlight.

How LED filaments work

The invention of high-brightness blue LEDs by Shuji Nakamura at the Nichia Corporation in 1993 paved the way for the practical use of LEDs in lighting applications. A white LED consists of a super-bright blue LED coated with a phosphorescent material that converts the blue light into a broad spectrum perceived as white by the human eye.

LED filaments are simply a logical extension of white LEDs. A LED filament consists of a long, thin substrate, usually made from a special plastic, glass or sapphire, with a large number of blue LEDs mounted on it and connected in series. At very low current, the individual LEDs can be seen as separate bluish dots (Figure 3). This assembly is covered by a phosphor coating, and terminals are attached to the ends to allow the filaments to be spot-welded onto supports in simulated incandescent lamps.

Because each filament consists of a large number of blue LEDs in series, a relatively high operating voltage (about 70 volts) line voltage, this is an advantage because it allows the electronics in the lamp base to be kept fairly simple. However, for our application it forms an additional challenge.



Powering and driving the display modules

The circuit basically consists of three sections as shown in Figure 4: a boost converter that raises the 12 V supply voltage to 70-80 V, a group of seven segment drivers, and two anode drivers.

The output voltage of the boost converter can be adjusted over the range of approximately 35 to 100 V with trimpot P1, and it can supply a maximum output current of about 100 mA. As the output capacitor has a rated operating voltage of 100 V, it's advisable to avoid setting the voltage higher than about 80 to 85 V, which is more than adequate for our purposes. The converter is built around a 555 timer IC using a conventional design. This sort of converter can be built at relatively low cost, and it is fairly stable and not particularly demanding with regard to PCB layout.

Of course, we optimized the circuit layout as much as possible. That includes keeping the connections between the input capacitors, MOSFET, inductor, diode and output capacitor as short as possible and using broad copper tracks. The feedback circuit composed of R38, R10, P1 and T1 is located as far away as possible from the areas on the board carrying large currents, and we chose input and output capacitors with low ESR. The



Figure 1. LED filaments in a simulated incandescent lamp.



Figure 2. Loose LED filaments get the creative juices flowing.



Figure 3. At very low current you can see the individual LEDs on the substrate.

result is a circuit that works well despite its simplicity and stays cool even with a relatively high load current of 100 mA. With an output voltage of 82 V and an 820 Ω load, the load current is 100 mA (8.2 W). Under these conditions the input current is 790 mA with a 12 V input voltage (9.48 W). This means the efficiency is better than 86%.

The board is intended to be permanently mounted as part of a larger unit, so we decided to omit reverse-polarity protection for the supply voltage connection. The relatively high current would cause significant diode heating, degrading the efficiency, and using a P-channel MOSFET would drive up the cost of the circuit. However, we did include a 2 A polyfuse.

The LED filaments are connected in common anode configuration for this application. The segment drivers (cathode drivers) are built around MPSA42 NPN transistors, which have a reasonably high maximum operating voltage. Logic high levels on pins 1 to 7 of connector K2 cause the corresponding LED segments to light up, with the currents limited by resistors R2 to R8. The brightness of the LED segments can be varied by adjusting the output voltage of the boost converter or by adjusting the values of resistors R2–R8. Optionally, it could also be adjusted by software using pulse drive signals with variable duty cycle.

The final part of the circuit consists of two anode drivers, allowing two display modules to be operated in multiplex mode. These drivers consists of complementary PNP/NPN transistor pairs (MPSA42 and MPSA92). If you want to connect more than two display modules, you can simply build more anode drivers on a piece of perfboard. The maximum number of display modules that can be connected to a single power supply is four to six,

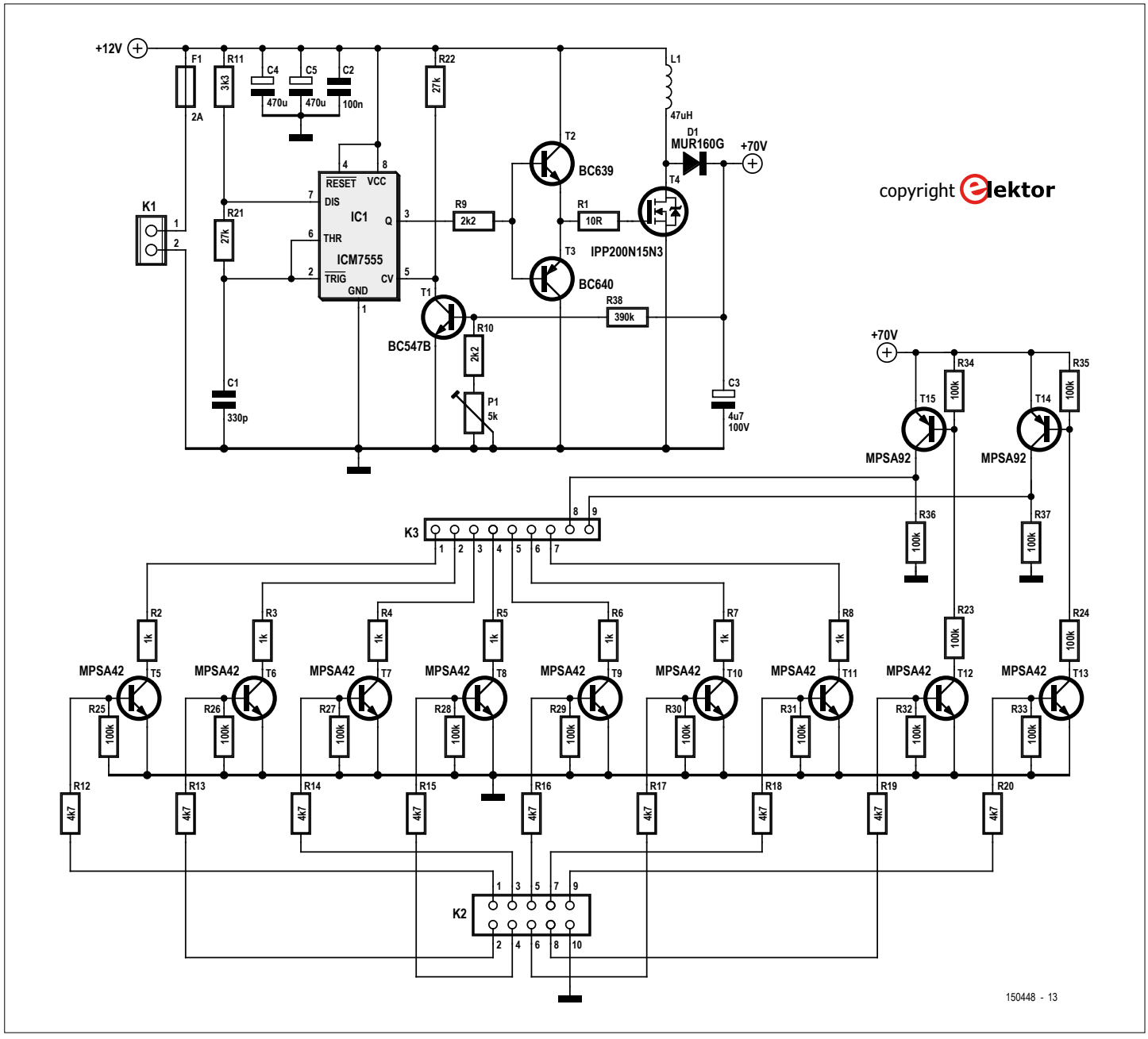


Figure 4. The switching power supply has high efficiency and low power dissipation.

Component List Power Supply Board

Resistors
R1 = 10Ω
R2–R8 = 1kΩ
R9, R10 = 2.2kΩ
R11 = 3.3kΩ
R12–R20 = 4.7kΩ
R21,R22 = 27kΩ
R23–R37 = 100kΩ (R36,R37 optional)
R38 = 390kΩ
P1 = 5kΩ trimpot

Capacitors
C1 = 330pF
C2 = 100nF
C3 = 4.7µF 100 V, low ESR
C4,C5 = 470µF 25V, low ESR

Inductors
L1 = 47µH, Würth WE-PD3 744 591 47

Semiconductors
D1 = MUR160G
T1 = B547B
T2 = BC639
T3 = BC640
T4 = IPP200N15N3 G
T5–T13 = MPSA42
T14,T15 = MPSA92
IC1 = ICM7555

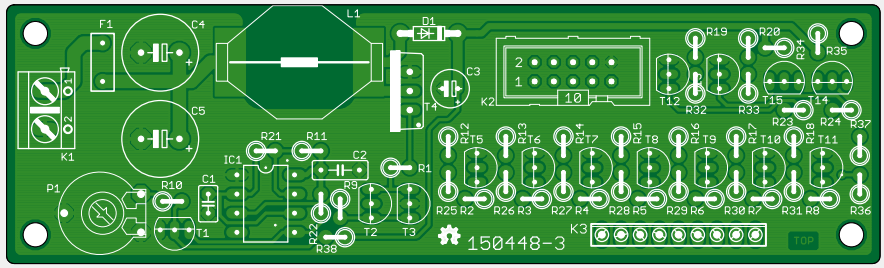
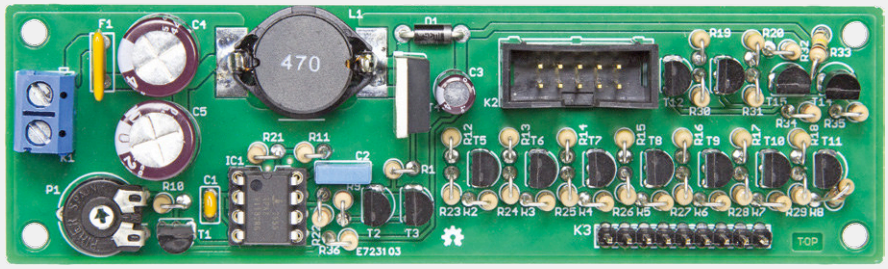


Figure 5. The PCB makes building the circuit a good deal easier.



Miscellaneous
F1 = 1.85A, 30V MCC33161
K1 = 2-pin PCB screw terminal block, 5mm pitch
K2 = 10-pin (2x5) boxheader, 0.1'' pitch
K3 = 9-pin pinheader
PCB # 150448-3 from Elektor Store

depending on the desired brightness.
To make construction a bit easier, we designed a PCB for the driver circuitry (see Figure 5). The LED filaments are mounted on a separate PCB. The schematic for this is shown in Figure 6, and the PCB layout is shown in Figure 7. For the setup shown in the lead illustration, we also designed an adapter board (Figure 8). All PCB layouts can be downloaded for free at [1].

Software

The display can be connected to existing circuits that work with seven-segment displays. In some cases you may need to invert the drive signals. You can also use the display with a microcontroller, an Arduino board or even a single-board computer.

With an Arduino and the example program for the IV-3 shield published in the September & October 2015 issue of Elektor [2], you can use the display modules to build a clock, a thermometer or a voltmeter. However, in that case you should adjust the timing for driving the display to avoid the risk of slight flickering. In the new example sketches for this article [1], the refresh rate for the display modules has been raised from 250 Hz to 500 Hz.

Building the seven-segment displays

The easiest way to mount the LED filaments is to use machined sockets removed from IC sockets or socket headers. You can mount the sockets on a PCB or insert them in predrilled holes (diameter 3 mm) in a black acrylic sheet, and then make the connections with wires on the rear.

Bend the ends of the filament terminal strips at right angles and carefully push them into the sockets. Pay attention to the polarity — the anode of the LED filament is indicated by a red marking or a small hole in the terminal strip. You can use yellow or white 3 mm LEDs for decimal points or colons. By the way, in our experience the LED filaments are very fragile and often do not survive shipping. A safer alternative is to buy some light bulbs with LED filaments in a local store and carefully dismantle them to remove the filaments. In any case, you should always handle the filaments very carefully.

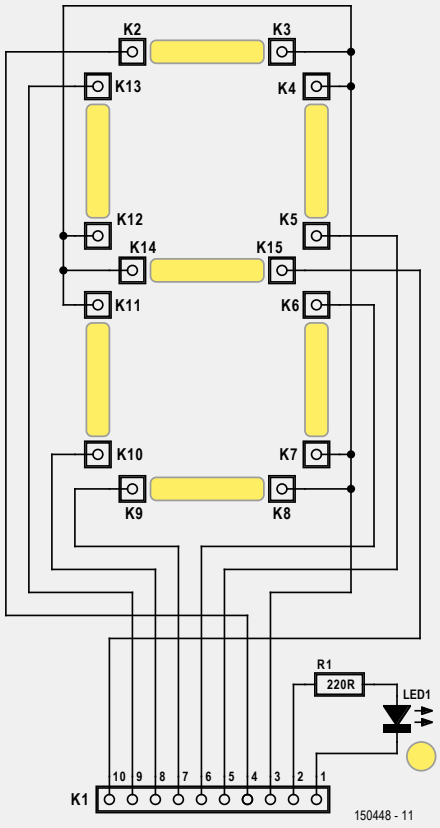
To improve the legibility of the display, we recommend screening the individual LED filaments, for example with strips of black acrylic, and covering the display module with an optical filter. We obtained good results with a filter made from 3 mm Umbra Plexiglas, which has a sort of brownish-gray tone that gives the unit a nice retro look. Of course, you can also try other colors or even a matte acrylic sheet.

If you have access to a laser cutter, a CAD drawing for a seven-segment display with dimensions of approximately 67 x 100 mm is available at [1].
You can also buy a kit with a piece of Umbra Plexiglas and a black PCB for the display module from the Elektor Store.

Building and testing the circuit

Building the circuit is fairly straightforward. Start by mounting the low-profile components, such as the diode, inductor, trimpot, IC socket and ceramic capacitors. Next come the resistors, followed by the transistors in TO92 packages. After this you should mount the connectors, the 4.7 µF output capacitor and the polyfuse. Finally, mount the tall components such as the input capacitors and the MOSFET. Then insert the IC in the socket and turn P1 fully counterclockwise. Connect the circuit to a 12 V power source (with a rated output current of at least 1 A) and measure the output voltage between the cathode of D1 and ground. It should be approximately 35 V. Adjust the trimpot

Component List 7-segment display



Resistor
R1 = 220Ω

Semiconductor
LED1 = 5mm diam.

Miscellaneous
K1 = 10-way pinheader receptacle, 0.1'' pitch
PCB # 150448-1 from Elektor Store

Figure 6. The seven LED filaments are driven through a 10-pin connector...

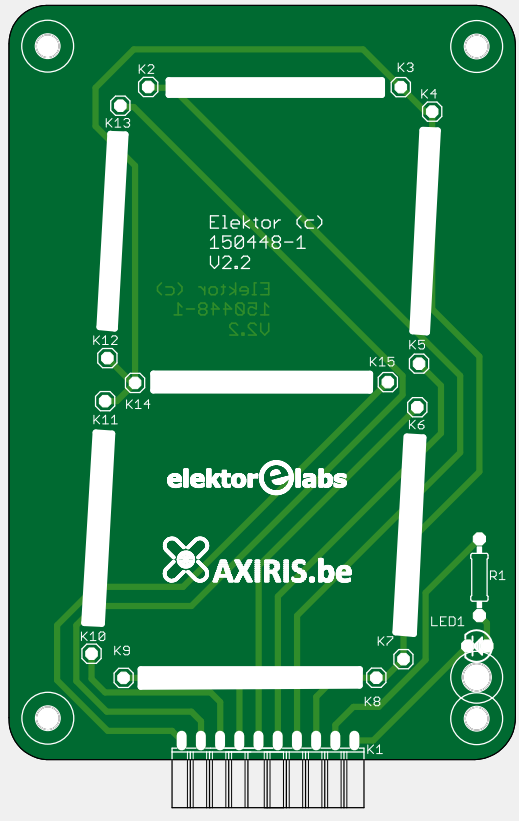


Figure 7. ...and arranged on the PCB to form a seven-segment display.

to obtain a voltage between 75 and 80 V. Now the circuit is provisionally ready for use. After the display modules are connected, you can adjust the trimpot again to obtain the desired brightness level. Sometimes the leakage currents of T14 and T15 (MPSA92) in the off state are high enough to allow the LED filaments to glow dimly when they should be dark. If this is undesirable in the specific application, you can insert optional 100-kΩ resistors (R36 and R37) between the anode terminals of the display modules and ground.

Finally, we warn that the output voltage of this circuit is potentially hazardous. Avoid contact with the output terminal while building and testing the board, and install it in a safely insulated enclosure. ⏮

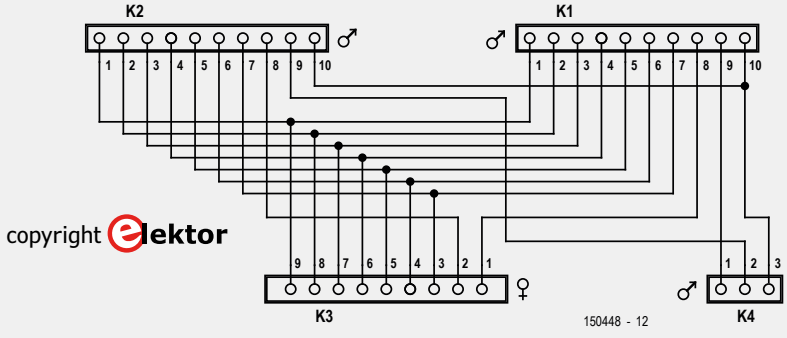
(150448-I)

Component List Stacking Board

Miscellaneous
K1,K2 = 10-pin pinheader, 0.1'' pitch
K3 = 9-way pinheader receptacle, 0.1'' pitch
K4 = 3-pin pinheader, 0.1'' pitch
PCB # 150448-2 from Elektor Store



Figure 8. With this adapter board you can drive two display boards simultaneously.



Web Links

[1] Project page: www.elektormagazine.com/150448

[2] VFD Shield article: www.elektormagazine.com/150064

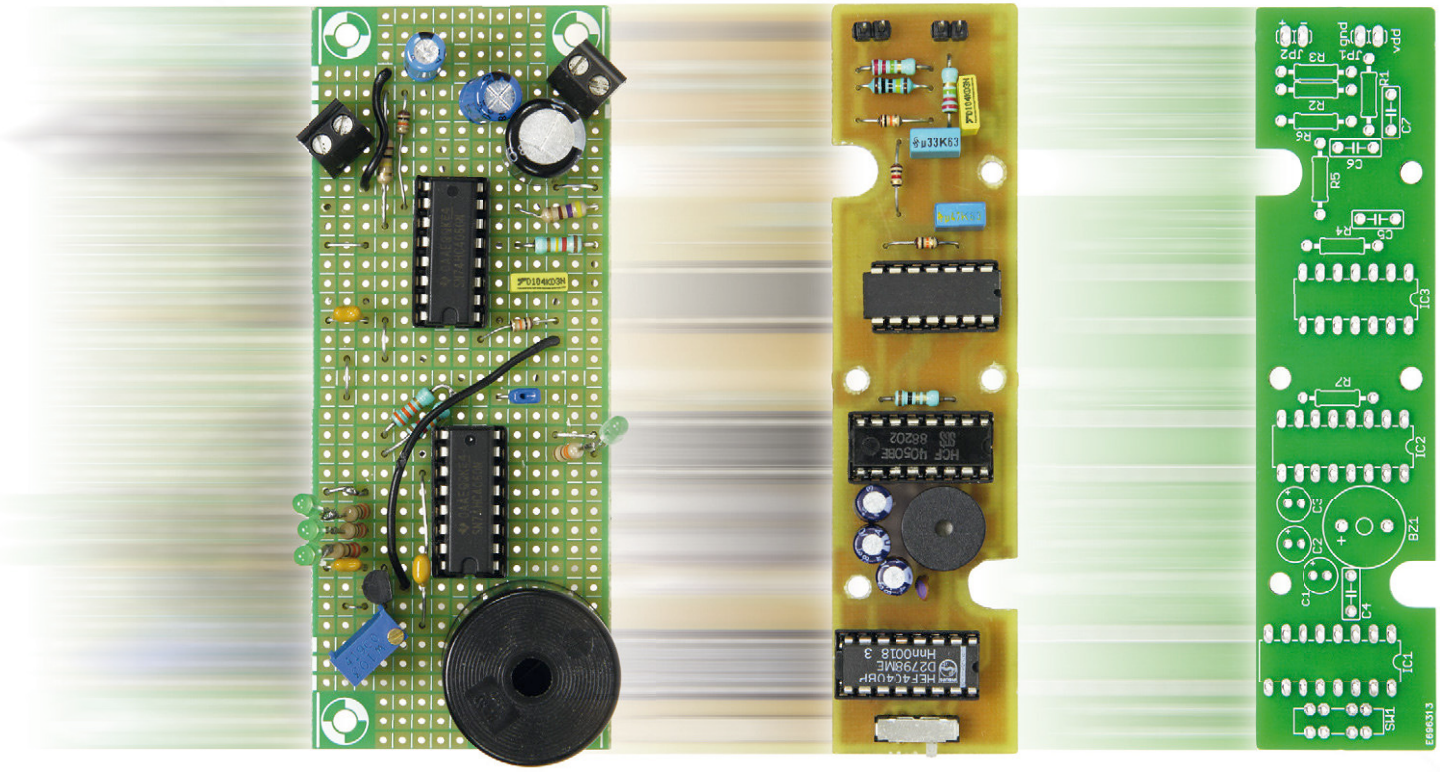
LEARN DESIGN SHARE

elektor PCB Service

offered in collaboration with EURO CIRCUITS

Generate your own PCB
using the Elektor PCB Service

Affordable **High Quality** **Reliable**



The Elektor PCB Service is the most extensive fully customized service for printed circuit board production in Europe. With convenient online tools allowing you to visualize and analyze your design before you order and pay .

- For beginners, there is the **NAKED-Prototype Service**:
  This produces single and double-sided PCBs without solder masks.

- For a more advanced service, there is the **PCB Visualizer** that shows you how your PCB will look after production, with a **PCB Checker** performing a DRC for you and the **PCB Configurator** that lets you customize your order details.

Smart menus and select options guide you through the ordering process. You can see in advance exactly what our machines can produce so there won't be any surprises!

So start your next project here:

www.elektorPCBservice.com

UV Exposure Unit with PIC-based Timer

54 UV LEDs in a matrix



By **Steve Griffin** (United Kingdom)

The idea of using ultraviolet LEDs to expose photo-sensitive circuit boards is not a new one but this design hopefully provides a solution with consistent timed exposure using very simple circuit construction, downward exposure and using off-the-peg boxes to house the unit.

This article is a pointer to the full story Elektor Members can find in the Labs section of the Elektor Magazine website. While we have space here to discuss the schematic and some other aspects of the project, the complete article on our website has the nitty gritty on construction and various options available to builders, not forgetting important resources like the PIC software and PCB artwork. So for "the works", go to [1].

What's in the box?

The basic system consists of two circuit boards; the UV exposure circuit and the microcontroller timer circuit. The UV exposure board has 54 UV LEDs arranged in an offset matrix and the control board is a simple PIC controller intended for mounting above the UV exposure board. It is programmed as an adjustable timer with a 7-segment display, two timed outputs; one for single-sided PCB exposure and another for anyone wanting to extend the unit to include double-sided PCBs. Also, a sounder output is included to indicate when the selected exposure period is over: usually about 2.5 minutes.

Controller board

The controller circuit (**Figure 1**) is constructed on a double-sided circuit board and is designed to mount directly above the inverted UV exposure board to create a compact configuration. An in-circuit programming facility (ICSP) has also been included (J7). Please note that if the ICSP programmer provides the supply voltage then ensure that the UV unit supply is turned off whilst programming.

The PIC 16F877A was selected because it is a commonly available chip with many easily accessed ports. It is supplied by a linear 5-V regulator which is powered from a 24 VDC / 1 amp wall-socket

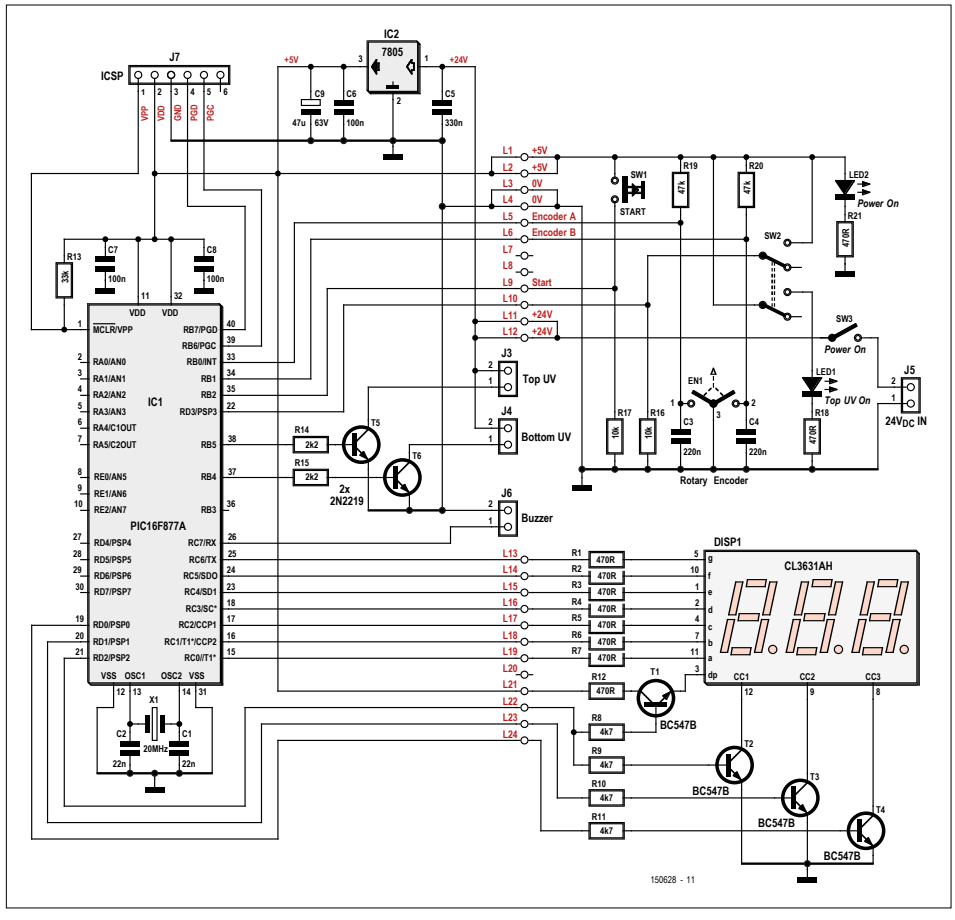


Figure 1. Schematic of the controller board, also comprising the control elements and the 3-digit 7-segment display.

adapter which is also used (raw) to supply the UV exposure board(s). LED D2 indicates when the power is switched on. The clock is set at 20 MHz by an external crystal, X1, which means that the setup is more than adequate for the task. Ports C and D are set as outputs and used to control the triple, 7-segment display with the display digits strobed by transistors T2, T3 and T4 with the decimal point used to separate the minutes and seconds display is strobed by T1. Of the remaining bits, port C7 has been used to operate the buzzer and port D3 is used to activate the second UV exposure bank if required. This would be indicated by LED D1.

All of the other functions are supported by port B. Two bits are used to operate the unprotected open collector transistors which control the UV exposure board/s and the others are set as inputs to detect the operation of the Start button and the rotation of the encoder (EN1) which is used to set the exposure period.

Most of the operations are very straightforward but the encoder implementation is more complex but key to the system operation. In this application, the encoder has 16 divisions and can rotate through 360 degrees in either direction. Every clockwise change of division causes an additional 15 seconds of time to be added and every anti-clockwise change of division reduces the time by 15 seconds. The full range of delay times available in this design is from 15 seconds to 9 minutes 45 seconds. Because rotating the encoder causes switching i.e. step changes, there needs to be a de-bounce circuit for each output included. These comprise of the R19/C3 and R20/C4 combinations giving time constants (RC) of about 10 ms. Operation of the unit is simply a case of switching on, selecting the required time delay using EN1 and pressing the Start button. The UV LEDs illuminate immediately (provided the lid is in place) and the display counts the seconds down to zero and then switches off the UV LEDs and activates the buzzer for five pulses.

UV exposure board

The exposure circuit is designed on a standard 160 mm x 100 mm single-sided Euro board. The "circuit" (**Figure 2**) consists of nine identical columns of six UV LEDs having a forward volt drop of 3.4 volts each at a forward current designed at 20 mA. By using a standard AC power plug-in 24 VDC supply — which actually supplied 23.7 volts under load. The current limiting resistor value was found from:

$$23.7 - (6 \times 3.4) = 3.1 \text{ V}$$

and then doing $R = V / I$.
With I = 20 mA:

$$R = 3.1 / 20 \times 10^{-3} = 165\ \Omega$$

Hence 150 Ω was chosen as the preferred value.

The UV LEDs are arranged so each one is approximately equidistant from its neighbor in any direction therefore giving an even illumination. They are also designed so that two boards could be mounted side by side and electrically connected in parallel to give a larger exposure area (160 mm × 200 mm) if desired. However, each open-collector output (T5/T6) is capable of driving up to four identical UV boards. The distance between the UV LEDs and the exposure surface is best between 100 mm and 150 mm.

Total flexibility

The controller board and the UV LED / Buzzer board are secured to form a sandwich construction, see **Figure 3**. The assembly is secured to the top panel of the exposure box using four screws and standoffs. Many more configurations are feasible depending on the board(s) you want to illuminate, like single-sided 160 × 100 (180 mA) or 160 × 200 (360 mA); or the same PCB sizes but double-sided (360 mA / 720 mA respectively). The supply voltage is invariably 24 VDC. All options are explained in detail in the full article [1]. There is also a UV-Board type B with an additional 18 wide-angle white LEDs arranged to give a safe illumination allowing the standard unit to double as a light box with the flick of a switch. The LEDs selected have a very wide angle of view therefore requiring fewer components to cover the area required. However, there is a variable resistance included to achieve the desired level of illumination. This circuit also includes a safety mechanism to prevent the user from looking directly into the UV LEDs in error. The current consumption is about 420 mA.

(150628)

Web Link

[1] Full article: www.elektormagazine.com/labs/uv-exposure-unit-with-built-in-pic-based-timer

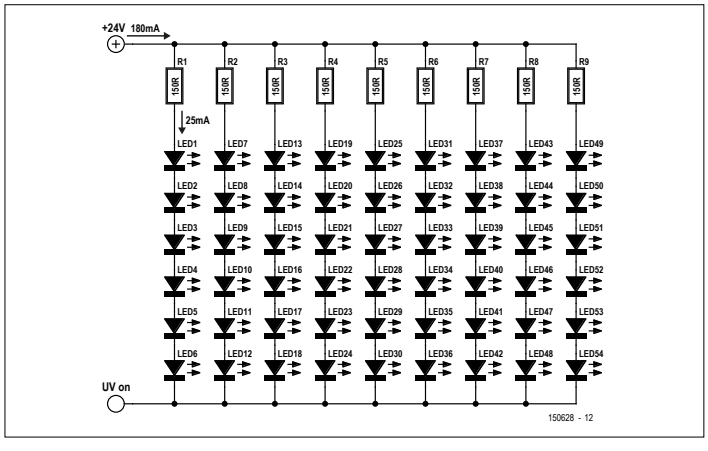


Figure 2. Schematic of the LED matrix board.

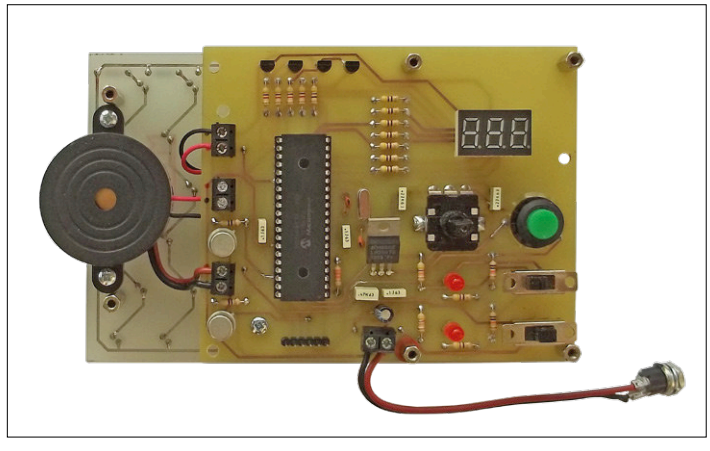


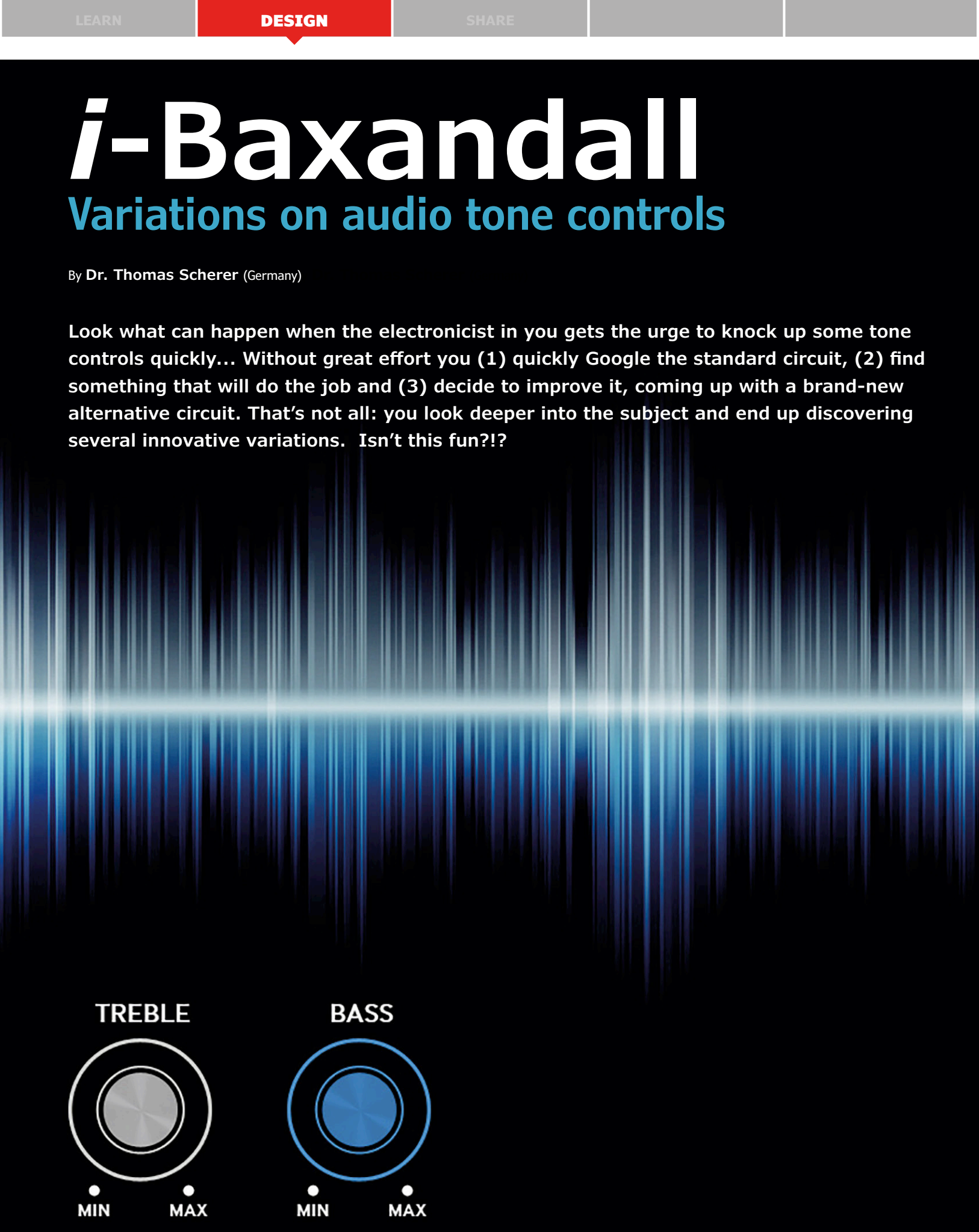
Figure 3. Control board and UV board assembly. Electrically the buzzer belongs to the control board but it is in fact mounted on the rear of the display board.

i-Baxandall

Variations on audio tone controls

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

Look what can happen when the electronicist in you gets the urge to knock up some tone controls quickly... Without great effort you (1) quickly Google the standard circuit, (2) find something that will do the job and (3) decide to improve it, coming up with a brand-new alternative circuit. That's not all: you look deeper into the subject and end up discovering several innovative variations. Isn't this fun?!?




TREBLE
VOL
BASS
MIN
MAX
MIN
MAX

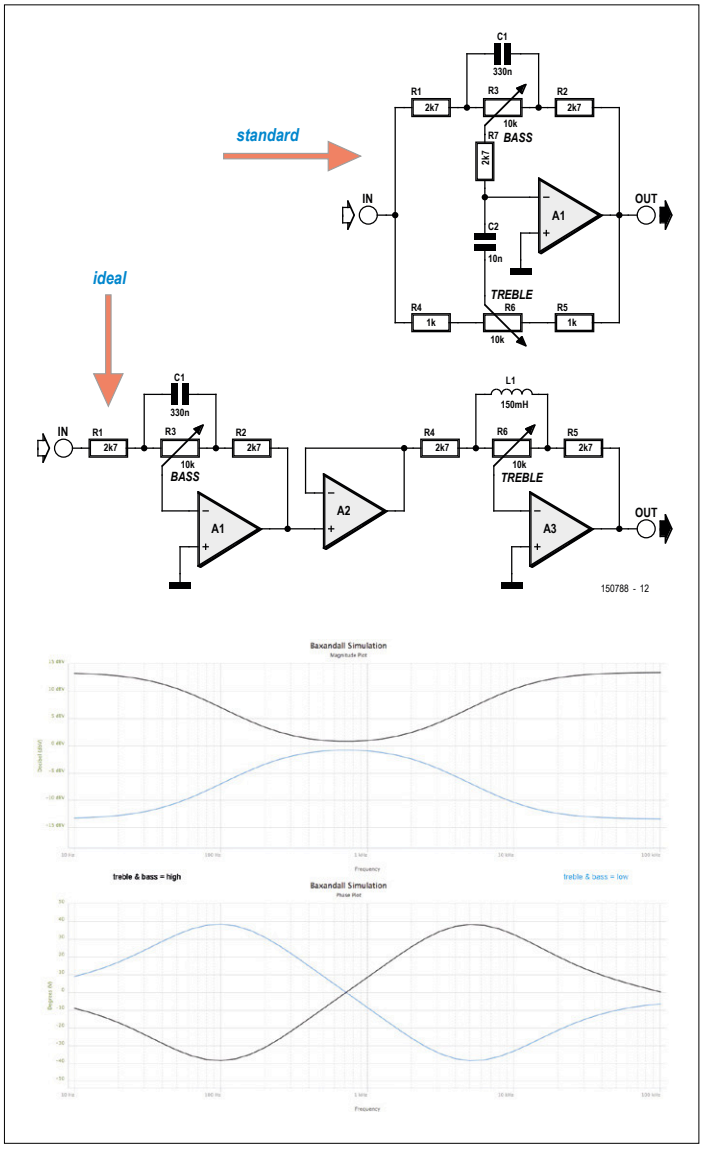


Figure 1. Standard circuitry (above) and idealized variants (below) of tone controls for frequency and phase range.

Recently I had a cheap and simple digital power amplifier — yes, I know: the spawn of the devil! — for which I wanted to make some equally simple tone controls. Doing this digitally is no big deal these days, yet on this occasion the exercise struck me as overelaborate and worse, I didn't have the required ICs 'in stock' in my small parts stores. So it had to be analog. Doing this for a stereo amp you need only one dual opamp and a couple of passive components plus a few pots. Easy! I even had the standard circuit for this somewhere in the back of my head, although to be on the safe side I naturally turned to Google for advice. What came out of this search was hardly surprising, being the kind of stuff that not just you but I too was expecting to find.

Incidentally, the answer to the question 'Who invented this circuitry?' was this time not 'someone from Switzerland'. In fact it was proposed as far back as 1950 by Peter Baxandall. For his efforts he received a watch worth $25 as a prize from the precursor of the AES (Audio Engineering Society).

So that was that and everything was hunky-dory? Stand by for the predictable answer, "In principle yes, but..."

Standard tone controls

For the search term 'tone control circuit' Google delivers several million hits (roughly 3,590,000 to be precise). Particularly popular on the Net in this regard is the entirely passive basic circuit, which I ignored on account of its associated weaknesses. So it had to be active. If you average out the numerous active designs, which differ only marginally, the result is something like what can be seen in the upper half of **Figure 1**. For each channel we need, besides the two pots for treble and bass, just one opamp, two capacitors and five resistors. For tone controls in the stereo version a dual opamp is completely adequate. Opamps with specific audiophile characteristics like, for example, the NE5532 dual opamp [1] or the quad version LM837 [2] are plentiful today. If you next consider that opamps of this kind should ideally have low-impedance for lowest noise figures, but need to be configured as high-impedance for lowest distortion, we obtain the ideal compromise for loading the opamp's output in the region of around 2 kΩ to 10 kΩ. With a desired adjustment range of around ±12 dB plus –6 dB boundary values of 100 Hz for bass and 5 kHz for treble we arrive at the component values given in Figure 1. So far, so good.

Inconsistencies

In point of fact this tale came to an end before it had properly begun. The components had already been calculated and selected, whilst by using a quad opamp, a low noise and distortion stereo tone control plus volume control had been designed. However, before realizing a circuit with physical components, the modern electronicist introduces a intermediate 'virtual' stage. I simulated the standard circuit using differing resistance values for both of the pots. For such a simple project I opted for the freeware and very easy-to-use online simulator 'PartSim' [3], available from the distributor Digikey. The software includes a simple circuit editor and a configurable simulator.

In the lower half of Figure 1 you can see the frequency and phase range of the standard circuit. The signal source here and in all the other plots has an amplitude of 1 volt. The simulations using various potentiometer settings demonstrate that the two controls for bass and treble have mutual influence on each other to a degree. This is little wonder really, because when you consider that the inverting input of the opamp is connected to an *R-C* element whose frequency response cannot separate high and low frequencies very sharply. This behavior is hardly surprising and in general domestic application the aspect described would be of minor importance. More serious is the fact that the load impedance delivered by the opamp is not really in the range from 2 to 10 kΩ. Either the pots are too high-resistance or else the load drops significantly below 2 kΩ. Incidentally the last-mentioned defect applies also for the input, if this is fed from another opamp configured as a buffer or a volume control. Summing up, the values shown are a fairly acceptable but rather lazy compromise. So we need to ask, is there another way?

Scope for improvement

Once we begin to consider improvements, one thing leads on

to another. And just as in software development, when we reach the point at which it's better to start afresh, radically novel circuit ideas burst in. For instance you might stumble over the impedance problem and the mutual influence of the two tone controls while you are separating the setting of high and low frequencies. The resulting 'ideal' variant in the lower half of Figure 1 makes available an extra buffer in A2 , which reduces the loading on the output of A1.

If we push this radical approach further, then we can eliminate this crutch especially at higher frequencies in the form of a *C-R* highpass filter and go the whole hog to make it symmetrical with regard to frequency. Using L1 we can insert an inductor at the point where the bass section has a capacitor (C1) located. As circuit techniques go, this is extremely elegant. Unfortunately reality disagrees and gives this nice idea the thumbs down. The resulting value of 150 mH for L1 is so impractical that the attractive symmetry will have to remain just a dream. Inductors with values in that range are monsters dimensionally. And in any case, they are inductors! These electromagnetic components are truly far from ideal. They are not only bulky but also have suboptimal series resistance, pick up magnetic interference and, for types with cores, unfortunately are not as low-distortion as they should be for demanding audio purposes. They cannot hold a candle to modern capacitors.

Gyrator

What's to do? Drop the idea? Not so fast: in electronics there is (almost) always a solution! In this case it goes by the beauteous name of *gyrator*. We are talking here about an impedance converter [4] that can transform the behavior of a capacitor into that of an inductor, so to speak. The great thing is that effectively all you require for a gyrator is one (extra) opamp. Here's how you simulate it. On the left-hand side of **Figure 2** we have fabricated a gyrator around A1, which together with R3 forms an *R-L* element, in other words a highpass filter. The circuit is very simple and so are the calculations. With the values given for C2, R1 and R2 we obtain an inductance of 100 mH, which is well within our target range. The impedance values of C1 and R2 are inside the ideal range too. The low value of resistance for R1 is not problematic, as the loading on the opamp output is composed of R1 + R3 || (C1 + R2).

Using 2.7 kΩ for R3 we get a decent highpass with a boundary frequency

$f_g = R / (2\pi L) = 4.3$ kHz.

But our problems are not over yet. The inductance realized with the gyrator displays a series resistance that corresponds exactly to R1. That is still not bad because thanks to the large

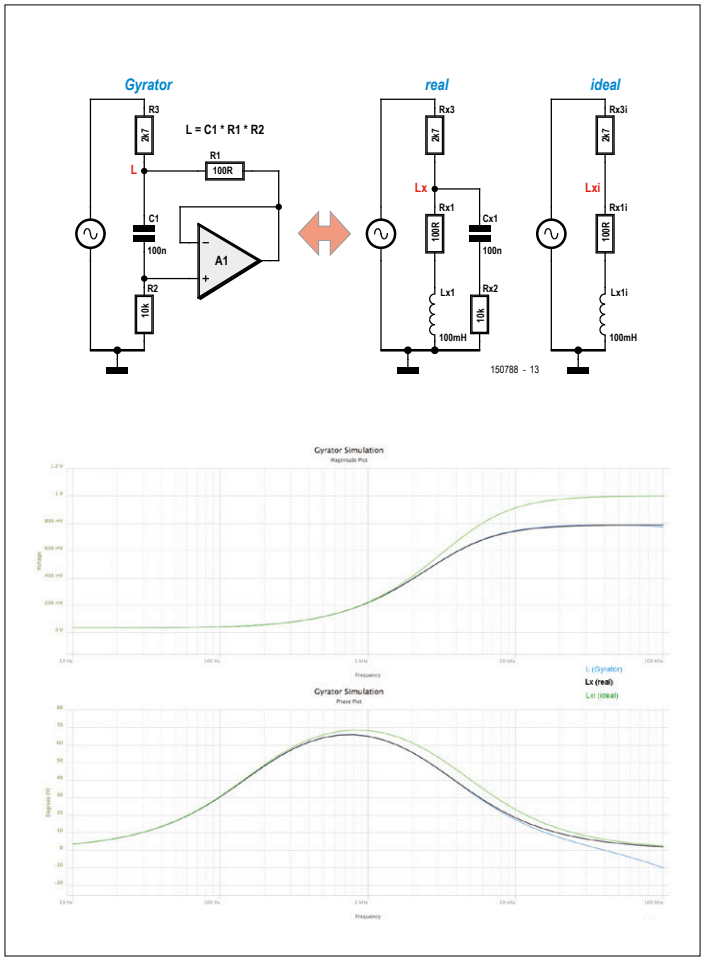


Figure 2. The classic gyrator used as an *R-L* highpass filter (left) plus the real (center) and idealized equivalent circuit (right) for frequency and phase range.

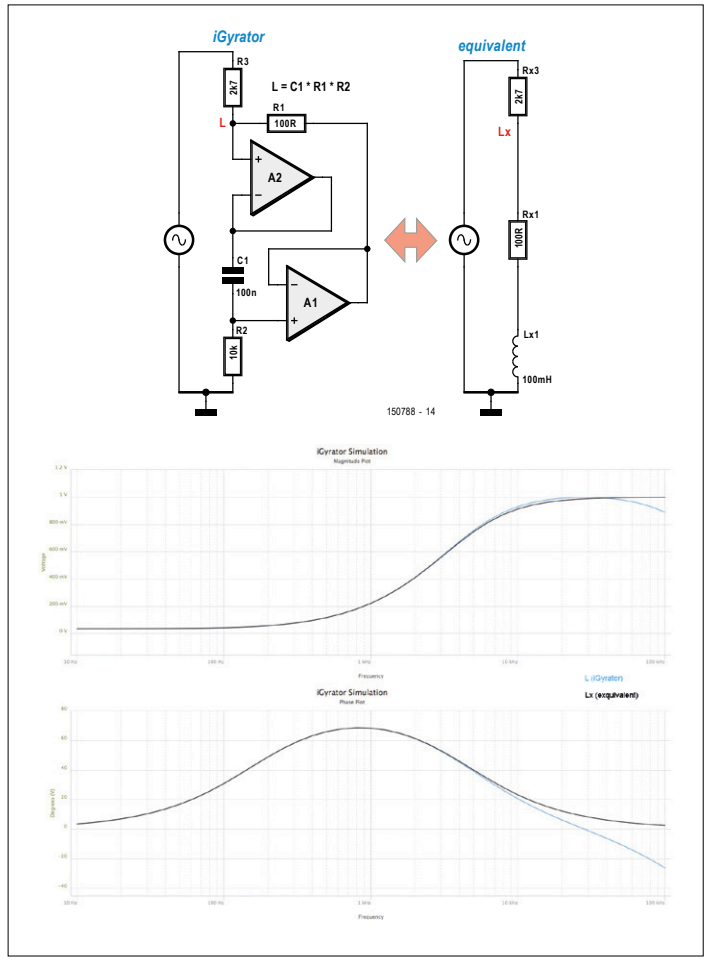


Figure 3. Improved *i*-gyrator used as an *R-L* highpass filter (left) along with the equivalent theoretical circuit for frequency and phase range.

value of the resulting inductor, the 100 Ω can be ignored, and relative to the circuitry impedance and the value of R3 defects arising in the frequency response will be negligible. Much worse is the fact that parallel to the intended ideal inductor between point 'L' and ground we still have the *R-C* element formed of C1 and R2, and this has a marked effect on the frequency response and suchlike.

To demonstrate the relationships, PartSim was used to generate not only the frequency and phase response of the *R-L* element with the gyrator but also the plots of the effective real (center schematic) and the idealized equivalent circuit without this *R-C* element (right). As you can see, the gyrator and the real equivalent circuits are virtually identical. The phase divergence at higher frequencies can be blamed exclusively on the restricted frequency response of the opamp used in the simulation; it is not so very important audio-wise. At the same time you can see that only the idealized equivalent circuit behaves as it ought to. Also confirmed is that R1 can be ignored.

i-Gyrator, *f*-Gyrator & *f*-Gyrator2

For the newly emerged problem involving the disruptive *R-C* elements formed by C1 and R2 there is once more an (electronic) solution. Impedance-wise we can isolate the *R-C* element from point 'L' really very simply with the help of an additional opamp configured as a buffer. The circuit in **Figure 3** is the result. As we are now dealing with an 'improved gyrator', I have baptized it *i-gyrator* for short. As you can see, the equivalent circuit corresponds completely to the idealized equivalent circuit in Figure 2. Here the frequency and phase response of the *i*-gyrator and its equivalent circuit have broadly the same coverage in the audio frequency range and come even closer if we use fast opamps.

But even now we have not solved all of our problems, since unfortunately the gyrator operates at its point 'L' only referenced to ground. If you examine the idealized circuit in Figure 1, you will see that the inductor L1 needs to be floating (voltage-free) if you actually wish to replace it by a gyrator. And that you can in fact do, as there is a really simple solution of

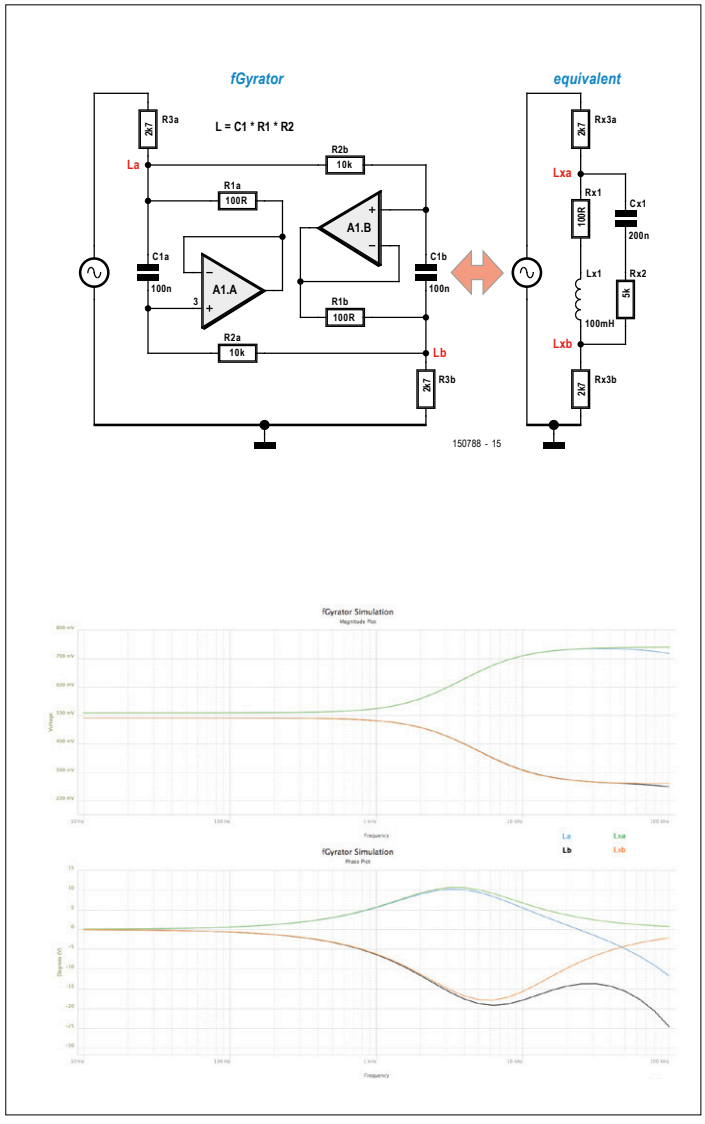


Figure 4. If you arrange two gyrators in anti-parallel you get a floating (voltage-free) *f*-gyrator that you can use as you wish in a circuit just as you would an actual inductor. Doing this unfortunately increases the loading of the parasitic *R-C* element more than in a simple gyrator circuit.

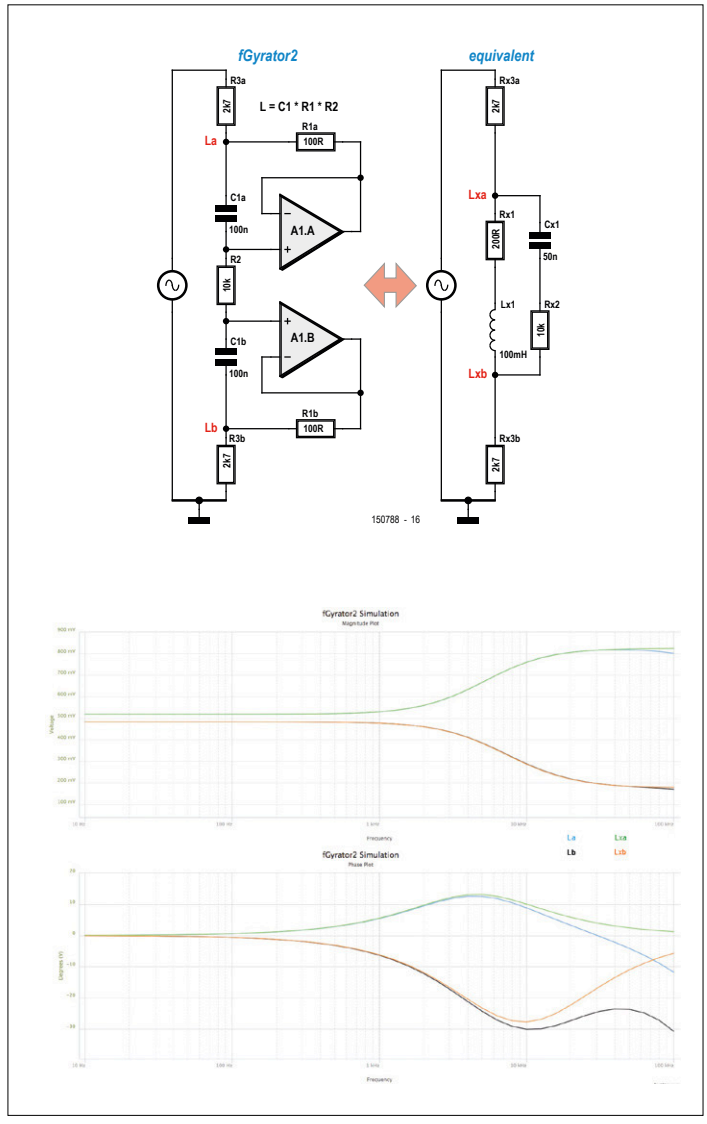


Figure 5. The second variant of *f*-gyrator2 uses anti-series gyrators. The loading of the *R-C*-element drops again, consequently raising the series resistance of the virtual inductor.

connecting two identical gyrators in anti-parallel. The *f-gyrator* circuit in **Figure 4** is a 'floating gyrator' and in fact functions just as the simulation indicates. The two R-C elements of C1a and R2a plus C1b and R2b now lie between the connections of the virtual inductors 'La' and 'Lb'. However, it also follows from this that the two R-C elements are connected in parallel with the virtual inductors. The equivalent circuit takes account of this, in that *C*x1 displays double and *R*x2 half of the values. Unlike *i*-gyrator, *f*-gyrator offers the advantage that the inductance is floating. But against this you must figure in the disadvantage that the influence of the two *R-C* elements is in total twice as strong as in the original gyrator.

Isn't there another way of achieving the floating state? It goes without saying that we could also simply connect the two gyrators in anti-series mode rather than in anti-parallel. **Figure 5** shows the resulting variant *f-gyrator2*, which even manages to use one less component, since R2 is used by both gyrators simultaneously. The interesting thing is that in this way the effective capacitance of *C*x1 in the equivalent circuit becomes a quarter of that in the *f*-gyrator and *R*x1 is doubled. Simultaneously the series resistance of the inductor Rx1 is also doubled but this is of no consequence, however. All in all, *f*-gyrator2 is somewhat better but unfortunately the *R-C* element wired in parallel with the inductor has not been eliminated.

if-Gyrator, *f*-Gyrator2 and *i*-Treble

To obtain a floating gyrator without the problem of the disruptive *R-C* element, we must combine the two principles. This immediately offers the prospect of the *if-gyrator* circuit for an 'improved floating gyrator' (**Figure 6**). In comparison with its unimproved predecessor, the influence of the parasitic *R-C* elements has vanished totally and in the process, the curve of the frequency response has shifted to the right in the direction of the ideal values. The price of this well-nigh perfect floating gyrator is that with two additional buffers we now require four opamps. That may appear excessive but hey, you can get four opamps in a single tiny quad opamp and by using SMDs the space occupied by this type of gyrator is relatively modest.

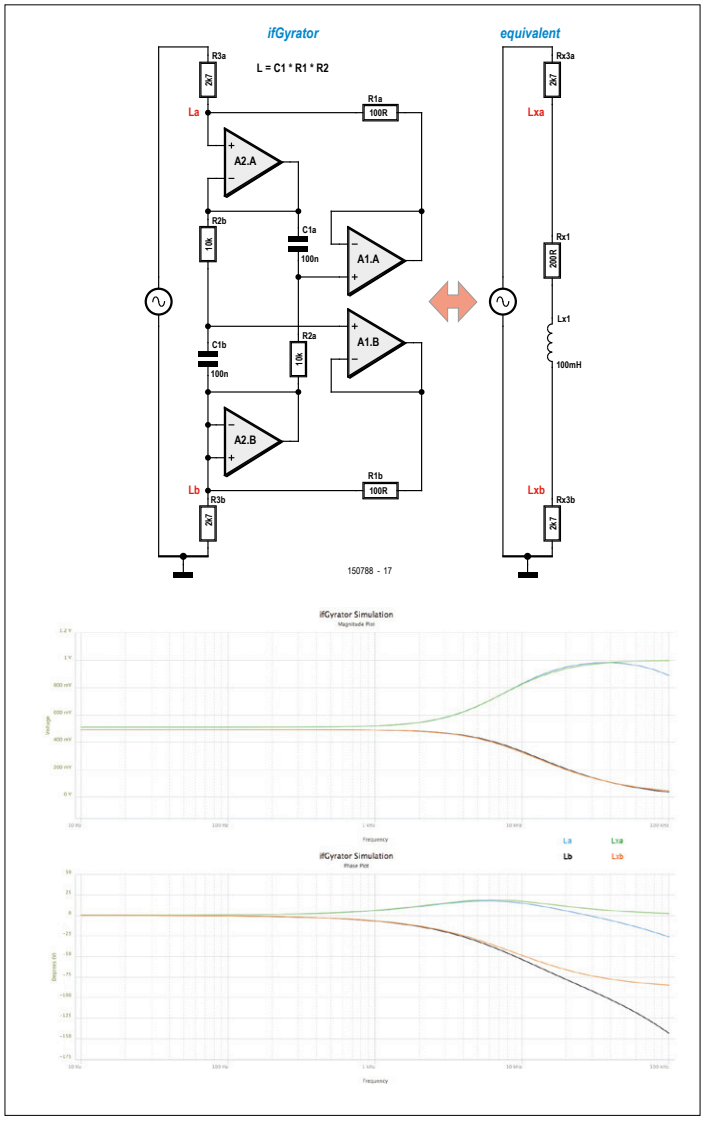


Figure 6. Combining *i*-gyrator and *f*-gyrator produces the *if*-gyrator circuit, which functions as a floating gyrator unencumbered by a parasitic *R-C* element.

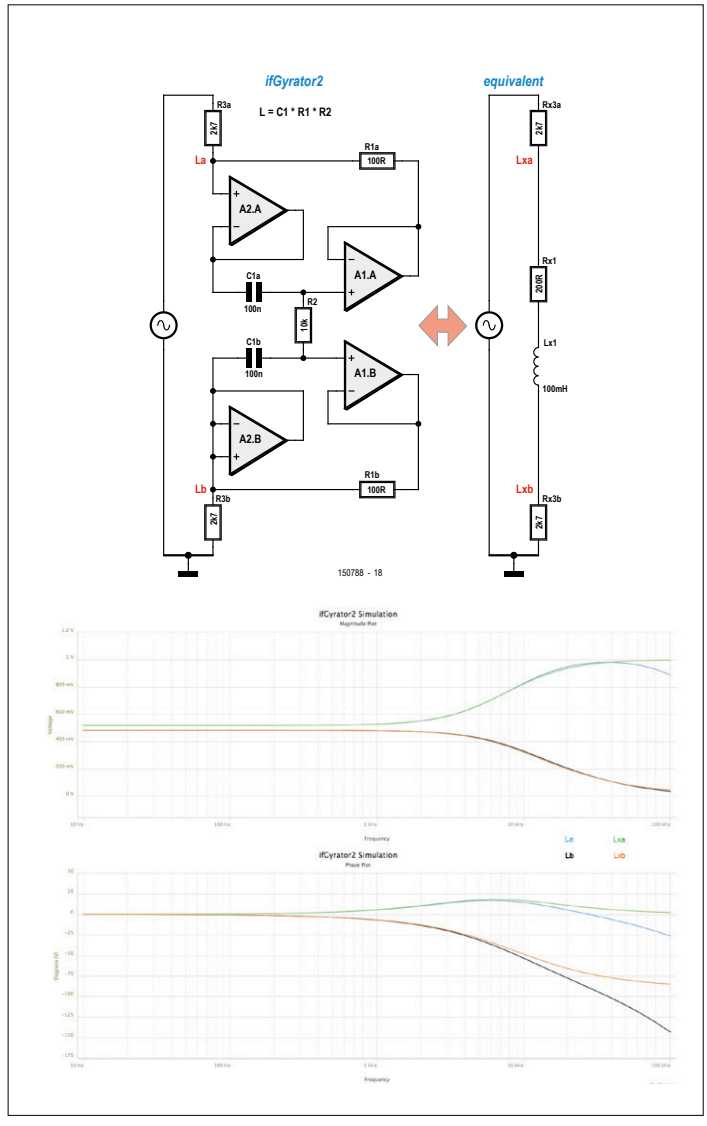


Figure 7. If you combine the anti-series *f*-gyrator2 with the *i*-gyrator, the consequence is an *if*-gyrator2. You obtain a lower resistance than with the *if*-gyrator, along with increased series resistance.

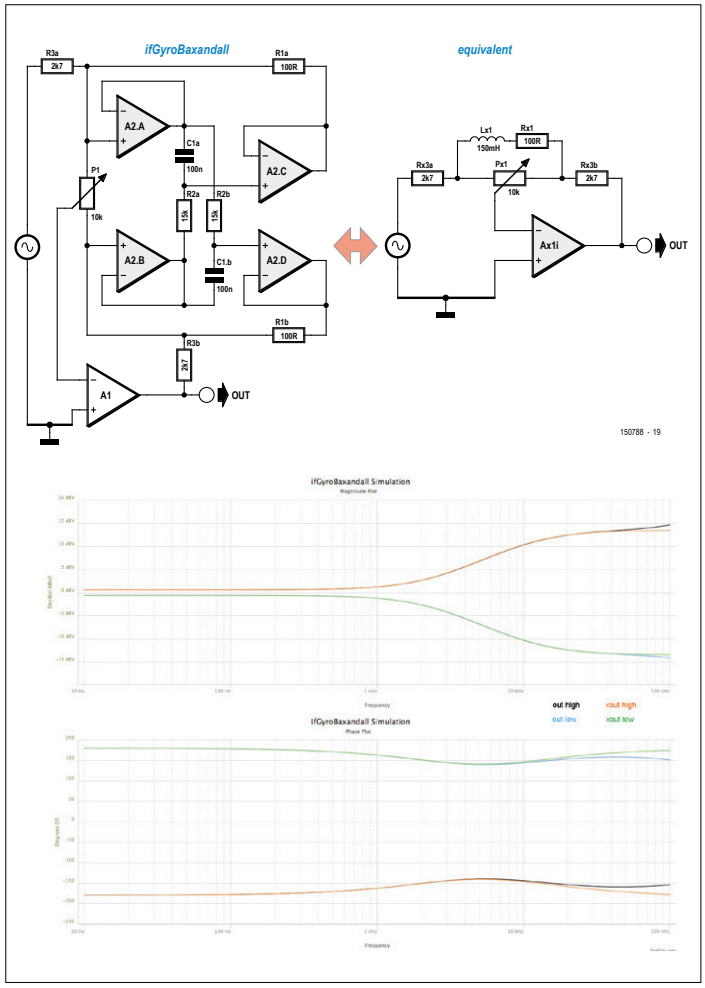


Figure 8. Replace the inductance in the idealized treble tone control with the *if*-gyrator and the result is the 'improved treble' circuit. A bit excessive perhaps, but it works.

You are bound to have guessed already: the if-gyrator also has a second variant with an anti-series based circuit. **Figure 7** shows the *if-gyrator2*, whose greatest advantage over the *if*-gyrator is that only one single R2 is needed. This admittedly

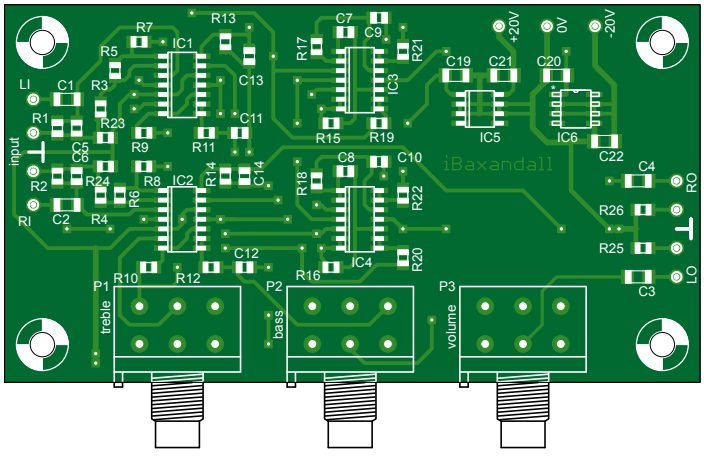


Figure 10. The PCB for i-Baxandall offers you the opportunity follow up this multitude of theory with practical experiments.

is offset by twice the series resistance, which is why this economy measure is hardly justified. If you then wish to replace the inductor L1 in the ideal circuit if Figure 1 by a gyrator, then you may as well go the whole hog and opt for the *if*-gyrator, which also no longer depends on a single resistor.

If you now use the *if*-gyrator for replacing the inductor in the treble control, the result is *i-treble*, the improved treble control in **Figure 8**. As you can see, the frequency range for elevated 'highs' and deep low frequencies in the audio frequency range produces largely the same coverage as with the equivalent circuit using a 'real' inductor. This is the proof that Baxandall tone controls can be made fully symmetrical by purely electronic means and entirely without actual inductors. Even in the phase response the variations between the high frequencies are kept within limits; they could be improved still further by using faster opamps, as the simulation used just standard types with a bandwidth of 10 MHz.

i-Baxandall

Summing up, the circuit technology insights acquired so far bring us to the synthesis of the *i-Baxandall* improved tone controls of **Figure 9**, in which bass and treble controls no longer conflict against one another. The stereo version of the circuit — complete with volume control and buffer opamps (IC1A and IC1C plus IC2A and IC2C) — employs the impressive number of 16 opamps! This is not really so extreme because using quad opamps reduces the total count at a stroke to four ICs (plus the two voltage regulators). Is this worth the outlay? Readers must answer this question for themselves. In any case, evidence has been presented to prove that surprises are still to be found in analog circuit technology. I myself was utterly astounded where this journey led me to when I attempted to improve upon an old basic circuit arrangement. For those who would like to test out these ideas in a practical way a circuit board (**Figure 10**) has been designed. Measuring just 92 x 50 mm including all potentiometers, it has turned out remarkably compact.

Using the component values indicated produces a control range of ±12 dB at 40 Hz or 10 kHz. The total amplification lies around a factor of 2 and can be adjusted using the ratios of (R3+R5)/R3 or (R4+R6)/R4. With two-fold amplification, even with maximum bass and treble at standard line level (0 dBm = 775 mV) and ±15 V power rails, there is sufficient headroom to avoid signal crushing. The distortion remains minimal — as long as the power amplifier and loudspeakers connected cooperate absolutely. R5 to R16 are specified as 15 kΩ — this is still a low-noise interpretation, since each of C7 to C10 is in parallel from an alternating current viewpoint. These values produce an inductance of 150 mH for the gyrators.

No use was made of a balance control by the way — intentionally. If you want to listen to music in good quality, then your seating position and loudspeaker placement should already be reasonably correct. Finally you should not be watching a TV receiver in a 45 ° angled corner either. People who wish to build audiophile tone controls should select high-grade components for the frequency-determining capacitors C7 to C12. Normal ceramic SMDs age progressively and consequently their

capacitance is not very stable. Anyone wishing to use large foil types for these can adapt the PCB layout files (in Eagle format) available for free download at the Elektor web page for this article [5]. Enjoy your experimenting!

(150788)

Web Links and References

[1] NE5532: www.ti.com/product/ne5532

[2] LM837: www.ti.com/product/LM837

[3] PartSim: www.partsim.com

[4] Gyrator: https://en.wikipedia.org/wiki/Gyrator

[5] Elektor Magazine web page: www.elektormagazine.com/150788

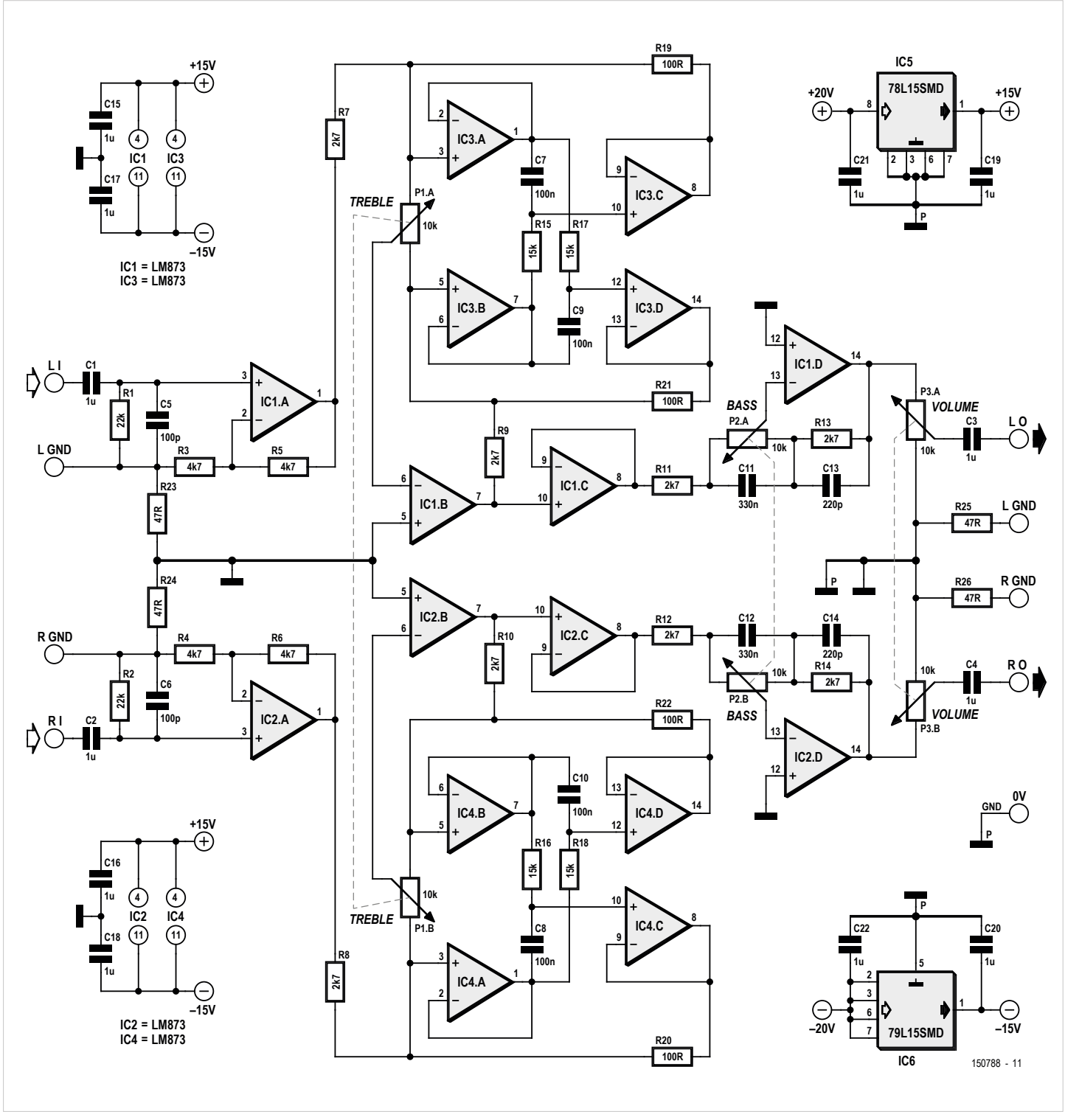
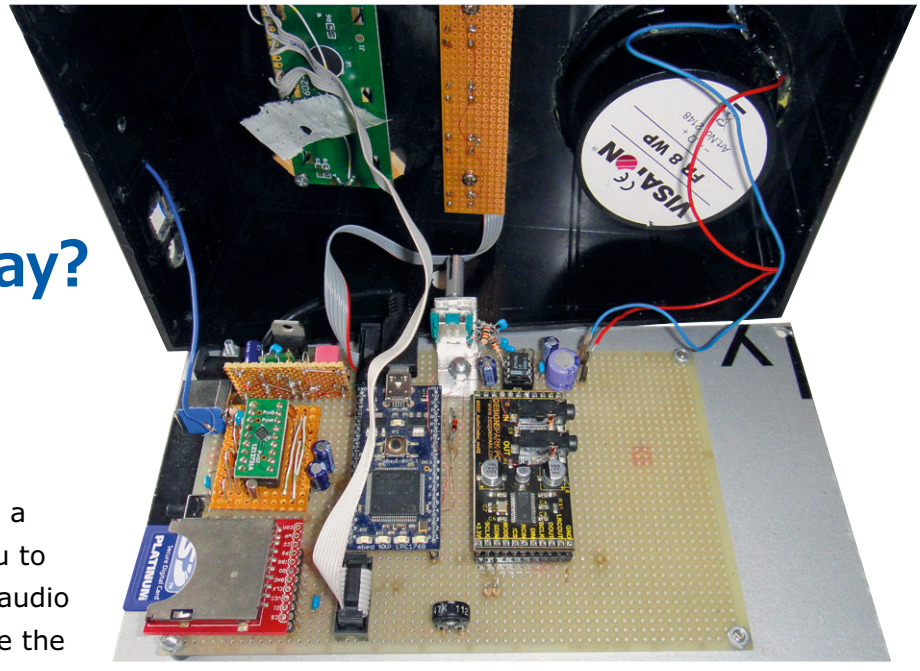


Figure 9. Lengthy deliberation, snappy result: the schematic of enhanced tone controls with the 'improved Baxandall' title. The 16 opamps are tucked inside four quad opamp ICs.

Timeshift Radio

What did they just say?



By Hans Oppermann (Germany)

This project saves audio data received into a circular buffer (ring memory), enabling you to replay it on demand. A Silabs receiver, an audio codec and a small ARM processor board are the ingredients you'll need.

Does this sound familiar? You're sitting comfortably at the breakfast table, chatting with your family and in the background you happen to hear the tail end of a traffic advisory on the radio — but you failed to catch all of it. Instant replay … wouldn't that be a marvelous facility?

On most satellite receivers (and some DAB receivers) a timeshift or 'wind back a bit' function is one of the standard features. But on normal radios I have never heard of anything like this and intensive research on the Internet came up with no satisfactory result. And with that an idea was of course born: if you don't have something, go and make one!

From the idea…

Lying around in my lab was an mbed module [1] with an LPC1768 ARM processor that up to now I had used only for experimental purposes. It struck me that its performance and I/O capabilities would be very suitable for this project. At last I had a practical application for the module and all I had to do was capture the received radio signal and replay it concurrently. But how?

Having stumbled across the mbed Cookbook web page [2] under the heading Audio, I spotted the perfect article: *TLV320AIC23B — High quality audio CODEC using I²S interface — can both play and record!* [3]. Can both play and record? Those were the exact words I was looking for! All the same, the more I reflected on this and the longer I considered the matter, the more I wondered whether this CODEC was really capable of simultaneously recording and replaying. Ioannis Kedros and Daniel Worrall, the developers of the relevant software library, allayed my misgivings on this score. So I must offer them my thanks at this point.

Now I had the task of getting hold of the TLV320AIC23B, of which I had never heard of previously, even though this Texas chip had been doing the rounds for a good ten years already. Here we are using the IC on a development board, albeit not the original tool from Texas Instruments but a tiny board from DesignSpark. This name will doubtless be familiar to Elektor readers in connection with the free PCB layout software from the well-known subsidiary of the distributor RS Components. The board is supplied only through RS Components [4], which may be problematic for private individuals in some territories, where you may have to call in a favor from a friend who has an account with RS. Being in Germany I surmounted the challenge and after this experience I had no major problem procuring the remaining components.

…to a practical modular design

So precisely what does the key component of this circuit, the audio CODEC module, do? Well, it transforms analog stereo audio signals into a digital serial signal (and vice versa). This is carried out using the I²S protocol for data transfer (I²S standing for Inter-IC Sound, not to be confused with I²C) developed by Philips and detailed in the **inset**. The audio CODEC module is equally well suited for recording as it is for replaying audio content in multiple resolutions and sampling rates.

For the receiver, as the block diagram in **Figure 1** shows, we're using a real 'old faithful', the Si4735 from Silicon Labs. This complete radio receiver, housed in an SSOP package, has, it's true, been discontinued by the manufacturer but remains available from many suppliers mounted on a small DIL adapter board. The Si4735 provides (via coupling capacitors C1 and C6) the analog stereo output signal ROUT and LOUT, exactly as required by the audio CODEC module. From this the audio Codec IC generates the digitized signal DOUT that we feed to the I2S function module of the processor. The processor saves the signal onto an SD card, which functions as a ring memory with a capacity of 1 GB, and returns it via DIN to the audio Codec. The analog signal ends up in the audio amplifier, consisting

mainly of an LM386, which peps it up for replay in the loudspeaker.
The timeshift function is achieved by continually saving and reading back the audio data. When you are listening 'live' the data is identical, whereas in 'timeshift' mode it's displaced in time. More on this later, when we come to describe the software.
The other elements of the circuit are for controlling the receiver (setting the frequency, storing the stations and so on), managing the timeshift function and displaying relevant data on the LCD readout. The RPG (rotary pulse generator) calls for special mention. The idea was to use a single knob for all control functions, which was only partly successful. The knob handles just audio volume and the menu selection. For the remaining functions, in particular the timeshift function, extra press buttons are necessary.
What else? The on/off module is definitely unusual but indispensable. The processor must be able to monitor activity also when the radio is switched off (by S5). You cannot just 'pull out the plug', as we still need to save (to the SD card) the information entered previously such as the station buttons, together with the last frequency and chosen audio volume. Switching off is signaled to the processor by P27, which saves the data and then 'pulls the plug' itself, reconfiguring P27 as an output and setting it back to zero. Switching on (S5) involves applying supply voltage to the processor. This reads in the data stored previously in order to continue operation using the same transmitter and audio volume as before. This circuitry is not my own invention but based on [6].

Now we'll cover the menu options for operating the receiver. These are selected by press button S4 and indicated on the LCD readout:

Select Station
A sub-menu lets you assign the digits 1 to 6 to transmitters selected.

Dial Frequency
The RPG enables a receive frequency from 87.50 MHz up to 108 MHz to be entered, turning the knob either forwards or backwards.

Set Station
Assigns the frequency currently selected to one of the stations numbered from 1 to 6.

Exit Menu
This terminates the menu functions.

My initial idea was to press the knob of the RPG to access the selected menu function but this proved to be awkward. Things worked out better when the On/Off button (S5) had this secondary function assigned to it. When using the menu functions this button activates the selected function, whereas in normal operation it switches the device on and off.
When you operate menu functions, listening to the radio needs to be interrupted, since performing both operations at once would exceed the capability of the processor. The Search button (S3) can be used to find a new transmitter while you listen to the radio. Audio volume is set the using the RPG.

Now for the timeshift functions, which you engage using the Start/Stop (S1) and Live (S2) buttons together with the RPG. A quick press on S2 halts live listening. The number of seconds stored already in the SRD card appears on the display. The RPG can now be used to 'wind back' the recording. Press S1 again and now what you hear starts back at the point to which you wound back. Pressing but-

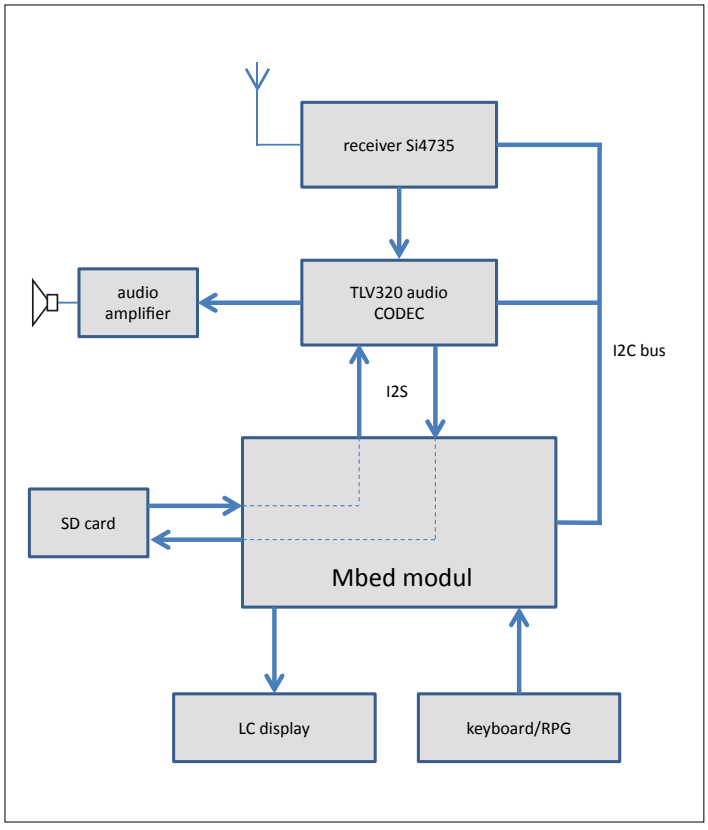


Figure 1. Block diagram of the Timeshift Radio.

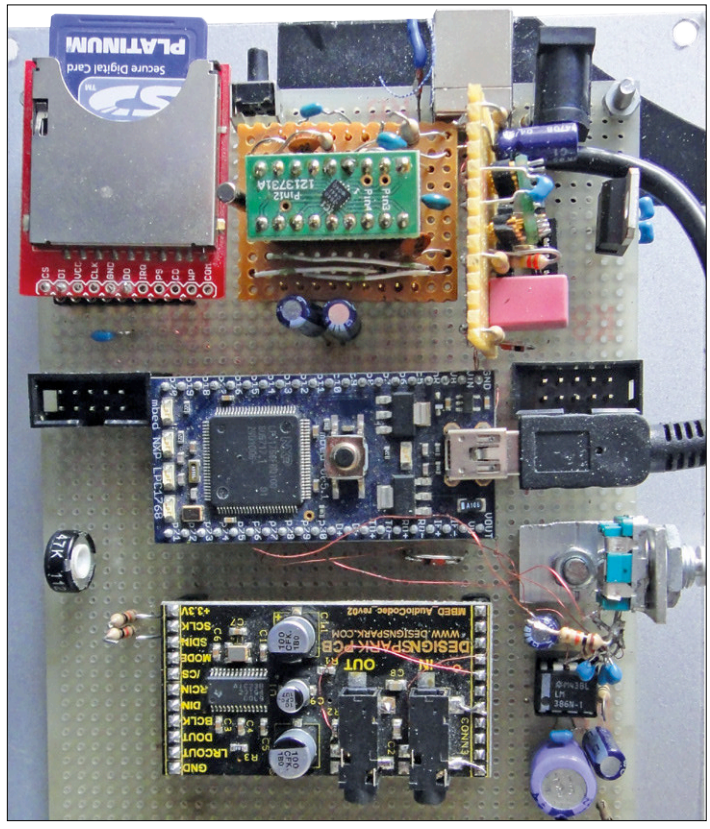
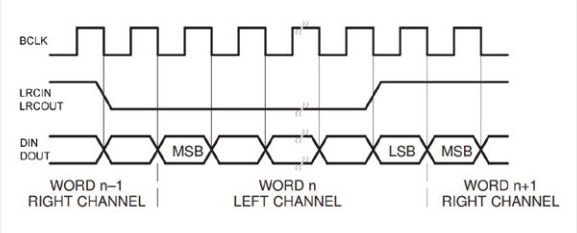


Figure 2. Board on board — genuinely modular construction.

The I²S interface

The synchronous inter-IC sound interface is used mainly for transferring digital audio data between analog-to-digital or digital-to-analog converters and digital signal processors. It has two signal lines (DIN and DOUT), two control lines for recording and replaying (LRCIN and LRCOUT) plus a clock line BCLK. The Word Select lines LRCIN and LRCOUT determine the timing of the data signal and the allocation of the signal to the audio channel. The frequency of the Word Select signal corresponds to the sampling rate of the signal.



Source: Philips

Pulse code modulation (PCM) is normal for the data signal, although other coding systems are also employed. The PCM signal always carries two audio channels, alternately one left and then one right data word.

ton S2 cancels the timeshift function and reverts to live listening.

Timeshift software

The software that I devised, all of which is available for download [7], consists exclusively of the module `main.cpp`. Library functions from the mbed project account for the rest. For that reason, the description here is confined to this module alone. At the head of this module are the libraries for recognizing the hardware components employed. For each of the components we instance an Object. Next comes Declaration of different Variables and a large number of auxiliary functions, which are documented comprehensively in the Source Code.

In the actual Main Function the following are initialized sequentially: the LCD display, the menu functions, the 1GB SD card and reading the configuration mentioned above. Using the function `initButtons()` an Interrupt function is assigned to the press buttons and is called when the button is pressed. Next comes initialization of the receiver, the TLV320 and the RPG. At the same time the TLV320 is put into `BOTH` mode, enabling it to record and play back at the same time. Additionally the function `record_play` is assigned to the audio interrupt, which is called when the audio buffer for the received data is full.

The following `while` loop (see **Listing**) handles the timeshift function itself with the two Integer Variables `read_sector` and `write_sector`. The received audio data is written to the SD card using `write_sector`, whilst `read_sector` reads it back from the SD card and passes it to the audio amplifier.

Pressing the Stop button S1 sets the `playing` Variable to `false` in the function `pausePlaying()`. This stops live listening, because `fillBuffer()` is no longer called. Now we can change the `read_sector` in `rewindPlaying()` and `forwardPlaying()`. If the button S1 is operated again, we now have `playing=true` and playback, using the changed `read_sector`, is resumed or repeated as appropriate. Pressing the Live button (S2) simply makes `read_sector=write_sector`, reverting to normal live listening.

The software can also be summed up like this: using `record_play()` audio data is received and `write_sector` is written respectively to the SD card. The `fillBuffer()` function reads audio data according to `read_sector` from the SD card and outputs this to the audio amplifier. Meanwhile, if applicable, `read_sector` may be modified by the timeshift function if initiated by the user.

The software was developed on the mbed web page [8], where it can also be compiled online and mounted as a Load Object onto the mbed module using USB. Details of the complete development process using an mbed module can also be consulted on this page.

Construction

I built the Timeshift Radio on perfboard (stripboard) as wiring the separate modules was paramount. For that reason there's no PCB layout. **Figure 2** shows how I arranged the individual modules. In the middle we have the mbed module, below this the LCD connections, with the connector to the press buttons above, and to the right of this the audio CODEC module. At bottom left you can make out the slot for the SD card, then above this, the Si4735 receiver, which is mounted on a tiny DIL PCB of its own but requires in addition a couple of external components. For that reason the complete receiver — in truly modular fashion — is constructed on an offcut of perfboard in the same way as the on/off section that can be seen above the receiver, even though it is mounted vertically.

On the left-hand side of the board there are also the connectors for power and USB (a short snippet of USB cable links the connector to the mbed module), the antenna and the reset button. Apart from a couple of individual components, the upper right-hand section of the PCB is

The timeshift function

```
while(recording) {
   streamToSD();
   if ((read_sector < write_sector) && playing) fillBuffer();
                                    //continually fill circular buffer
   if (playing) {                   // no pause
       if (iValue > lastRPG) {Vol--; audio.outputVolume(&Vol);}
       if (iValue < lastRPG) {Vol++; audio.outputVolume(&Vol);}
   } else {
       if (iValue > lastRPG) rewindPlaying();
       if (iValue < lastRPG) forwardPlaying();
   }
   lastRPG = iValue;
}
```

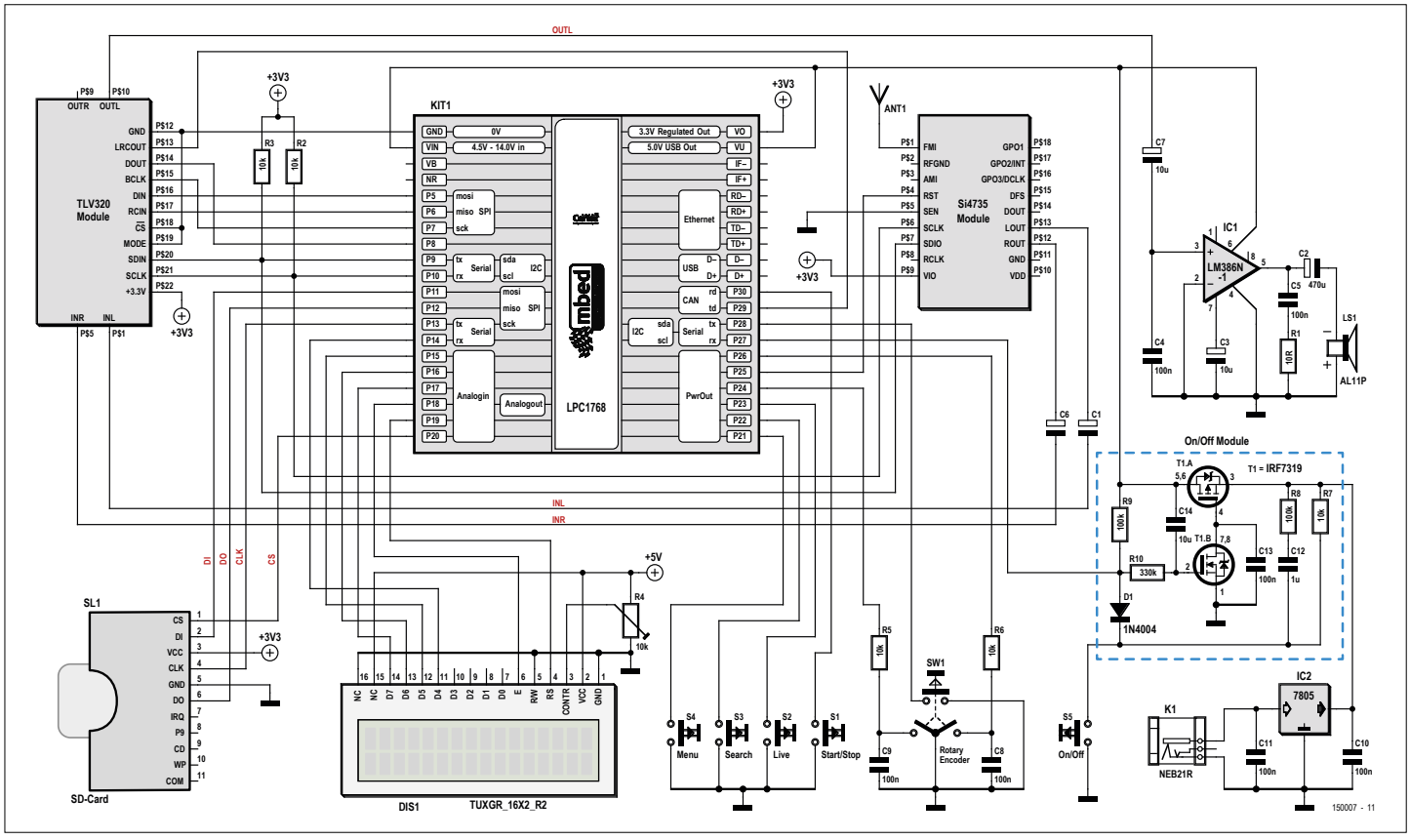


Figure 3. The schematic shows the crucial connections between the two modules.

devoted exclusively to the audio amplifier, built on the board using discrete components. Yes, discrete, because in comparison with the other components, we can also consider the LM386 as 'discrete'. What else is there on the board? The rotary encoder with capacitors C8 and C9 plus resistors R5 and R6 soldered direct onto it close to the audio amplifier, lower down a preset resistor for setting the contrast of the LCD, next to this the pull-up resistors for the I^2C bus below the mbed module. Then above the on/off module we have the 7805 voltage regulator with its decoupling capacitors and last but not least, on the right next to the receiver, the coupling electrolytics for the audio CODEC module. Forgotten anything? Keep the schematic (**Figure 3**) to hand and localize the individual components. They are all present there!

The wiring is carried out intricately and very painstakingly with a Verowire (or Road Runner) pen and 0.2-mm diameter (#30 AWG) enameled copper wire. The bewildering choice of an enclosure I solved with a black project box equipped with a sloping lid, as seen in the header photo. It may not look artistic but it works!

Apart from the project's external appearance, I can see scope for improvement in the sensitivity of the receiver. Although several descriptions of the Si4735 module attest to its good sensitivity, I'm unable to endorse this here. This probably has to do with my construction techniques (or with the project box). But my circuit is definitely not tied down to using the Si4735 for the receiver. It's entirely feasible to provide an audio input for an existing, better quality receiver (communications receiver or scanner). Also more effort could be applied to the debouncing of the press buttons. But ultimately we are dealing here with a draft design that is still a bit rough around the edges, so any further improvements are left to you and your imagination. ⏮

(150007)

Web Links

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

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

[3] https://developer.mbed.org/cookbook/TLV320AIC23B

[4] http://uk.rs-online.com/web/p/processor-microcontroller-development-kits/7541974/? (RS # 754-1974)
http://docs-europe.electrocomponents.com/webdocs/10fc/0900766b810fcde6.pdf (data sheet)

[5] www.ak-modul-bus.de/cat/documentation/Si4735.Documentation.zip

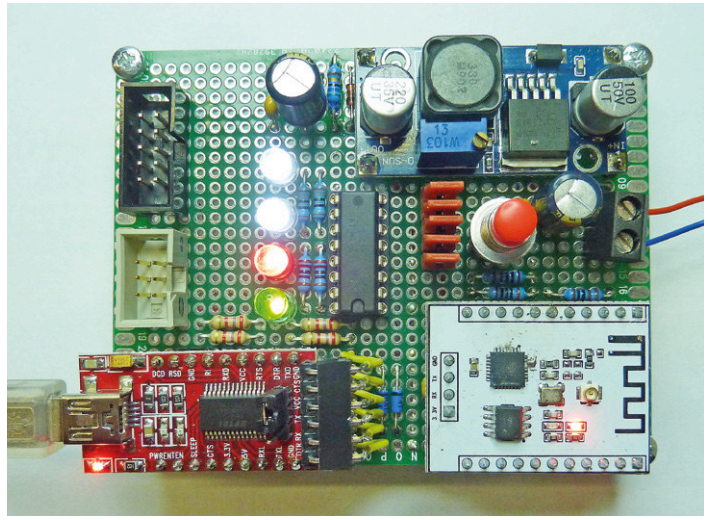
[6] www.mosaic-industries.com/embedded-systems/microcontroller-projects/electronic-circuits/push-button-switch-turn-on/microcontroller-latching-on-off

[7] www.elektormagazine.com/150007

[8] https://developer.mbed.org/handbook/mbed-Compiler

Compact and Self-contained WLAN

Using the ESP8266 without a host microcontroller



By **Walter Trojan** (Germany)

It's easy enough these days to connect a microcontroller to a WLAN and make it part of the 'Internet of Things' (IoT), and a wide range of chips and add-on boards is available to help. But is it possible to replace these two-part designs with something more compact, where the WLAN chip itself takes over the job of the microcontroller?

In Elektor edition 1/2016 (January & February) [1] we showed how the ESP8266 can be embedded in a microcontroller design and controlled using AT commands. However, the device is capable of more than that. As well as its dedicated WLAN interface circuit it also contains a powerful 32-bit processor that can be clocked at 80 MHz or at 160 MHz. According to the manufacturer the WLAN functions only consume 20 % of its processing power, which naturally leads to the idea of using the remaining 80 % for other purposes.

A look at the block diagram (**Figure 1**) shows the clear separation between the analog high-frequency circuitry, which includes a 2.4 GHz transceiver with antenna matching and the necessary frequency synthesizers, and the digital part including a Tensilica L106 processor core. As well as 32+80 Kbyte of RAM there is a useful peripheral module that can be accessed via 17 external connections. Depending on the application, these can be configured as GPIOs, as a UART, SPI or I²C interface, or for PWM generation. The firmware is stored in an external flash ROM connected using a high-speed SDIO interface (four data lines, two control lines). Current consumption in normal operation is around 70 mA peaking at 170 mA, and the power management unit can be used to put the device in a sleep mode where the consumption is reduced to about 10 µA. This set of features means that the ESP8266 is suitable for stand-alone use in small- to medium-complexity WLAN projects.

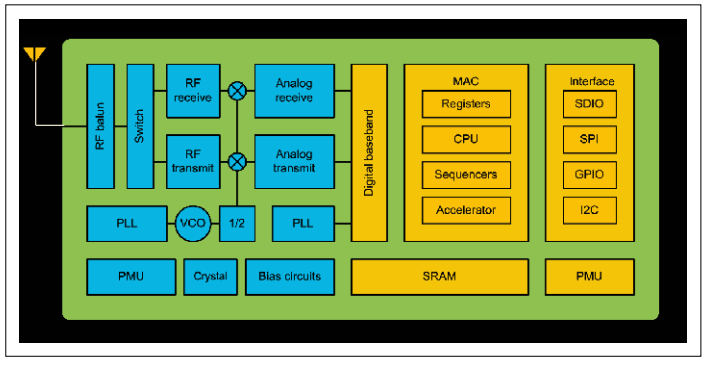


Figure 1. Innards of the ESP8266.

All roads lead to ROM

Espressif, the manufacturer of the device, provides a development system in the form of a virtual machine image running Lubuntu Linux. Since this might not be everyone's cup of tea the highly active community of ESP8266 enthusiasts has developed a number of alternative approaches. We will look at the three most popular.

Lua: a scripting language from South America

Lua (Portuguese for 'moon') is an imperative and extensible scripting language [2]. It is platform-independent, being implemented using an interpreter with a small memory footprint (120 Kbyte). It can call functions in C libraries, and likewise C functions can call Lua routines.

The ESP8266 can be flashed with the 'NodeMCU' Lua interpreter [3], overwriting the AT command processor. The chip will then instead accept Lua commands and execute them. Dedicated boards are available carrying an ESP8266 (preloaded with the interpreter) and a USB-to-serial interface. An easy-to-use IDE called 'ESPlorer' [4] is available for developing and executing Lua scripts.

Figure 2 shows a screenshot in which the left-hand pane displays a Lua script, while the right-hand pane shows the dialog with the ESP8266. The 'LED blink' example shown is an effec-

tive illustration of the simplicity and clear structure of a Lua script. One disadvantage of the Lua approach is the need to learn a new language and that the NodeMCU firmware takes a slice out of the resources available on the device.

Arduino IDE

The Arduino contingent has also been active and has integrated support for the ESP8266 in the Arduino IDE. Adding the extension to the IDE [5] is straightforward enough: simply add the address of the extension [6] under 'Settings' and 'Additional Board Manager URLs'. Then the 'Boards Manager' will offer the installation of the 'Generic ESP8266 module'. Once this has been installed the ESP8266 module will appear in the list of available boards.

If you now select 'Generic ESP8266 module' you can begin to develop firmware. If you use the most recent version of the IDE you may find that the IDE extension libraries have not yet been suitably updated: in this case it is best to install the previous version (for example, version 1.6.5 instead of 1.6.6). The integration is very clearly explained in a free e-book, some 300 pages long, by ESP enthusiast Neil Kolban [7]. The book also includes a good basic description and specification of the ESP8266: highly recommended!

The Arduino environment supports a WLAN shield through a powerful library with over 100 functions, and these functions are now also available to control the ESP8266. **Figure 3** shows a screenshot of the familiar Arduino IDE where the main window is displaying part of the code ('sketch') for a web server. A web server controlling some LEDs can be implemented in only around 100 lines of code. The browser window when accessing the server is shown in **Figure 4**.

Eclipse and C

Fans of the Eclipse development environment will be comfortable writing sketches in the Arduino IDE with the help of a plug-in (see Neil Korban's book). However, those who want to have complete control over their firmware and to exploit the ESP8266 SDK to the full will need to use the standard version of Eclipse and program in C. Eclipse is an IDE developed originally by IBM and released by them as open-source software in 2001. It was originally conceived as a Java development platform, but a large number of 'perspectives' (specially-designed window layouts) have been developed optimized for a range of programming languages.

For C and C++ work we have the 'C Development Tools' (CDT). Advantages of Eclipse include the fact that it runs under both Linux and Windows, and that it is under continuous development. Since 2006 ten versions have been released; the most widely used is called 'Luna', while the most recent, released in 2015, is called 'Mars'.

Well-known development platforms such as Atollic TrueSTUDIO and OpenSTM use Eclipse as their IDE. User CHERTS has put together a package for Windows that contains the following components:

- ESP8266 System Development Kit with documentation and many examples;
- Eclipse Mars IDE;
- MinGW compiler and package management system;
- a script to automate the installation of extra packages.

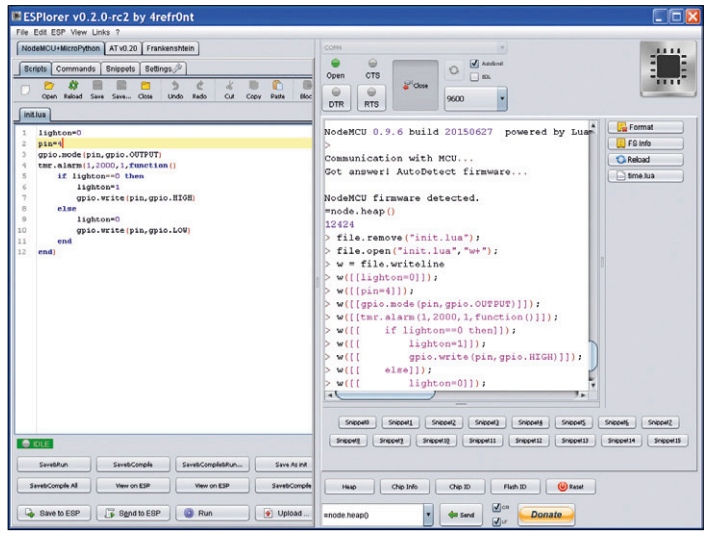


Figure 2. ESPlorer: the left pane shows the Lua script, while the right pane shows the dialog with the ESP8266.

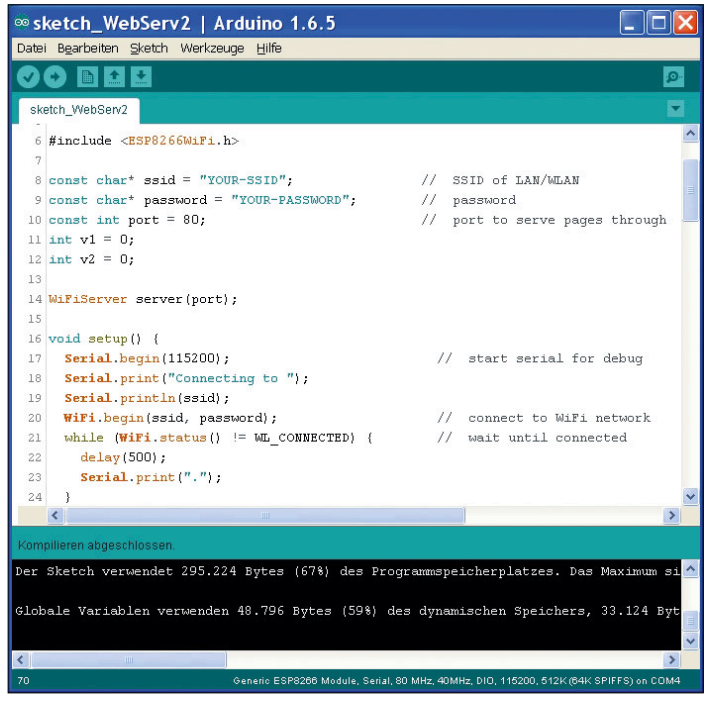


Figure 3. A simple web server in the Arduino IDE.



Figure 4. A page served up by the simple web server.

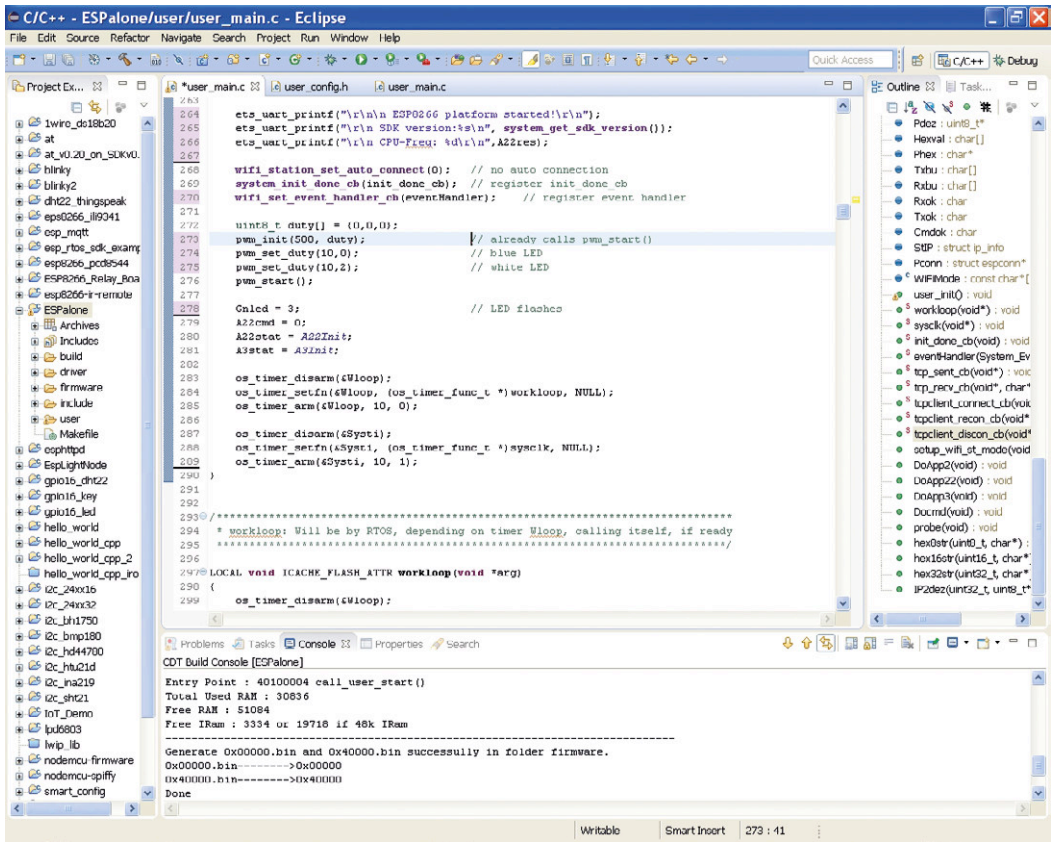


Figure 5. A densely-populated Eclipse window with lots of information about the project and potential actions.

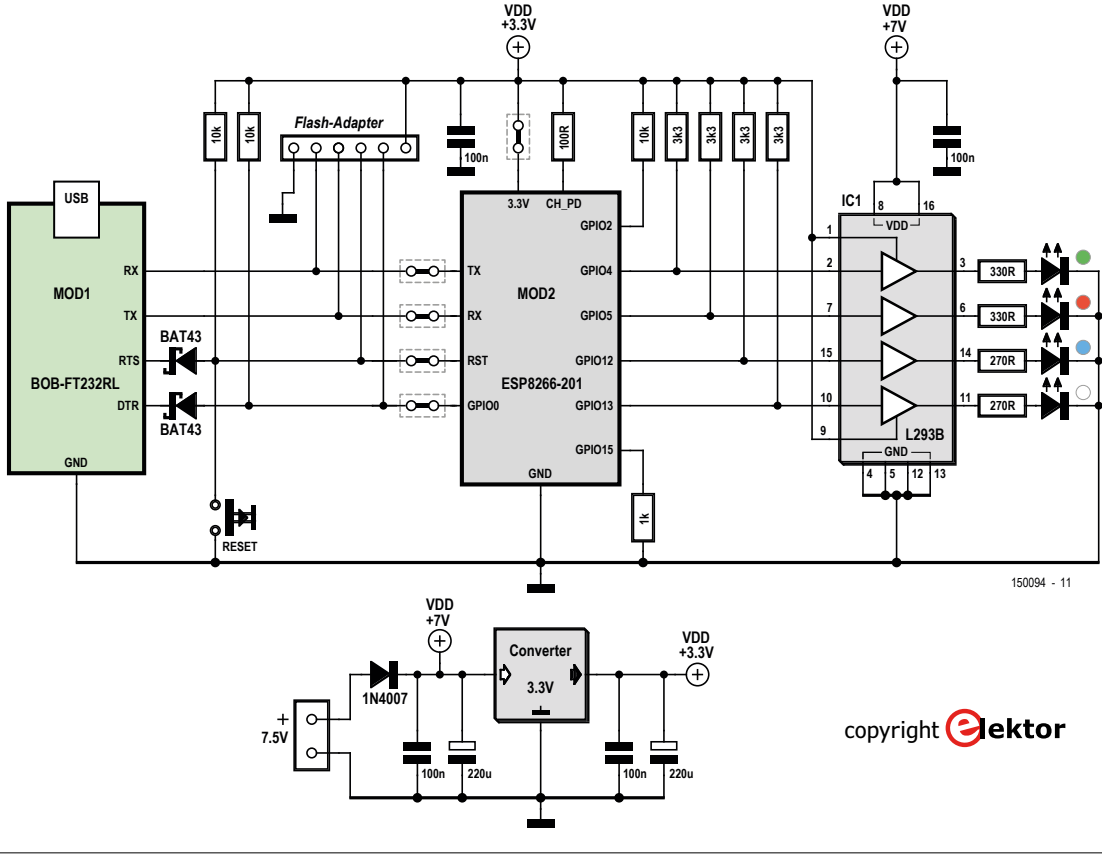


Figure 6. Hardware for the LED blink project.

Instructions and download addresses can be found at [8]. After installation, which should be trouble-free, the Eclipse window will look something like **Figure 5**. On the left in the project window are the available example programs and user projects, while the main pane in the center shows the C code. Under 'Outline' on the right you can quickly access all the variables and functions of the program. The pane at the bottom includes options to display console messages, error reports and search results. Additional utility functions such as flashing the ESP from Eclipse are supported by makefiles included in the projects.
Because I use Eclipse already for developing code to run on STM32-series microcontrollers, I personally find this approach the most convenient for programming the ESP8266 as well, and chose to use it when developing the example application described below.

Now things get tricky!

Now that we have seen a few different ways to approach the development process, we can turn to a concrete example project. The requirement is to control three LEDs as we did in the article in the edition 1/2016 [1], but without using a host microcontroller: all the work is to be done by the 32-bit processor in the ESP8266 alone. The device is to be operated over the WLAN using the same clients for PC, iOS and Android devices.
Take a look at the compact circuit in **Figure 6**. At its heart is an ESP8266-201 module, which allows access to a number of GPIOs. Four LEDs are connected using GPIOs 4, 5, 12 and 13. The green LED indicates the communication status:

- Off: no connection to the WLAN;
- flashing: ESP8266 is connected to the router;
- On: a client has connected.

Since the maximum current load on a GPIO pin is 12 mA the LEDs are connected via an L293B driver IC. This device will also provide a sufficiently high output voltage to drive white and blue LEDs. The driver is indeed over-specified for this use, but I wanted the board also to be useful in higher-current applications such as motor control. Pull-down resistors at the inputs to the drivers ensure that they see a defined level while the ESP8266 is booting. The level on GPIOs 0, 2 and 15 controls how the device boots up when reset: with GPIO 0 and GPIO 2 high and GPIO 15 low the ESP8266 will boot from external flash memory as desired. Pull-up and pull-down resistors are used to set the required levels.
A USB-to-serial converter based around an FT232RL is provided on the board to make it easy to load new firmware. Software is transferred to the ESP8266 using the TX and RX signals. First the device must be put into flash mode, which is achieved by holding GPIO 0 and RST low using the FT232RL's RTS and DTR outputs. The BAT43 diodes allow these pins to be used also in normal operation. The data transfer rate is 115200 bit/s, and everything is controlled by the Eclipse IDE. The USB-to-serial converter is powered from the USB connector. The ESP8266 can be reset manually to execute the new firmware by using the pushbutton to pull the RST line low.
It is important when choosing a USB-to-serial converter that its **outputs operate at 3.3 V**. A higher voltage is a sure way to destroy your ESP8266!
In order to allow other boards to be programmed, I decided to bring out the programming connections to a six-way header and to connect the module via jumpers. To program an external device remove the jumpers and connect the board using a short cable.
The power supply is not critical. An input voltage of 7 V to 9 V DC is fed directly to the driver (which can operate at up to 45 V) via a diode to protect against reverse polarity, as well as to the input of a switching regulator. The output of the switching regulator is set at 3.3 V. The supply needs to be rated to handle the average current draw of the ESP8266 (70 mA to 80 mA) and peaks of up to 170 mA. These current peaks are smoothed by the 220 µF capacitors. The construction of the board is shown in the **photograph at the head of this article**.

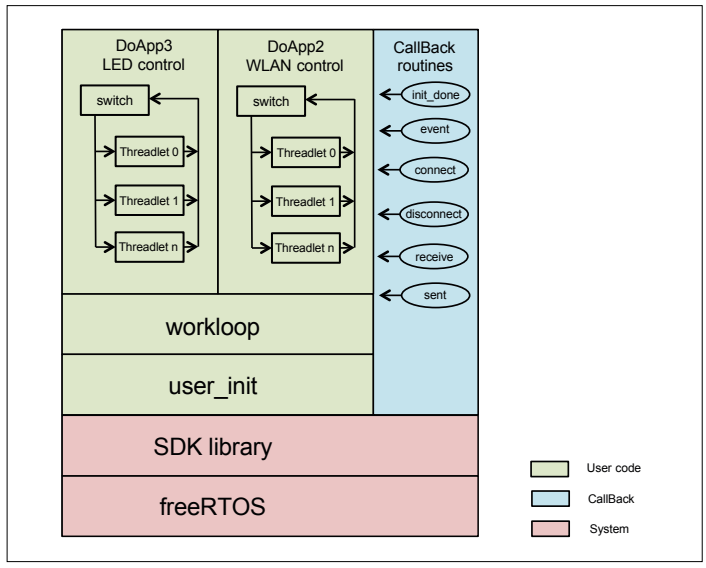


Figure 7. How the firmware is organized.

FreeRTOS versus 'bare metal'

Espressif supplies SDKs with or without a real-time operating system. The interfaces to the system and WLAN functions are the same in both cases. If, like me, you have only previously implemented 'bare metal' projects with firmware running directly on the microcontroller hardware, you will find working in a real-time operating system quite a change: what you lose in the level of control, you gain in convenience.
The well-known FreeRTOS operating system can be run on the ESP8266. It coordinates the WLAN functions with the other activities of the firmware. The following advice should be borne in mind when programming a FreeRTOS system.

- Avoid the use of delay loops which load the processor unnecessarily. Use (software) timers instead.
- Always define timer variables as static rather than local, for reasons that should be obvious.
- A user function should complete execution within 15 ms, as otherwise it will interfere with the concurrent execution of WLAN functions and may cause the watchdog timer to expire and reset the system.

Given these cautions, we can now proceed to the nitty-gritty of the firmware, whose structure is shown in **Figure 7**. The basis of the system is the FreeRTOS operating system itself, and the many utility functions offered by the SDK. Overall control, as

Listing 1. An 'infinite loop' under FreeRTOS.

```
LOCAL void ICACHE_FLASH_ATTR workloop(void *arg)
{
  os_timer_disarm(&Wloop);     // switch Wloop timer off

  DoApp2();                    // WiFi state machine
  DoApp3();                    // LED state machine
  os_timer_setfn(&Wloop, (os_timer_func_t *)workloop, NULL);
                               // call workloop again...
  os_timer_arm(&Wloop, 1, 0); // ... after a 1 ms delay
}
```

Listing 2. The function tcpclient_connect_cb(), which is called when a client connects.

```
static void ICACHE_FLASH_ATTR tcpclient_connect_cb(void *arg)
{
  Pconn = arg;                 // copy argument
  espconn_regist_sentcb(Pconn, tcp_sent_cb);
                               // initialize tcp_sent_cb()
  espconn_regist_recvcb(Pconn, tcp_recv_cb);
                               // initialize tcp_recv_cb()
  ConnState = TCP_CONNECTED; // signal status
}
```

in any C program, is handled by a main() function. However, this is not visible to the user: it runs in the background and manages (among other things) all the WLAN functions. After reset the ESP8266 calls the function user_init() once.

This function

- sets up the configuration of the peripherals, in this case the UART and the GPIOs;
- sets up an init callback and an event handler;
- enables a systick function for software timers;
- calls the main program loop, here called workloop().

Listing 3. Print commands document the program flow.

```
ESP8266 platform started!  // Start in user_init
SDK version:1.3.0          // version of SDK used
CPU-Freq: 80               // CPU clock is 80 MHz
Init done.                 // WLAN module is initialized
A2ConnStation              // DoApp2() takes over and...
ESP8266 in STA mode configured.
             // switches to 'station mode'
Event: EVENT_STAMODE_CONNECTED
             // ESP8266 successfully connected to router
Event: EVENT_STAMODE_GOT_IP
             // ESP8266 has received an IP address
A2CheckIP            // the IP address is checked and...
IP OK 192.168.178.28 // printed out
A2StartServer        // the TCP server is launched
                     // ... waiting for a client...
tcpclient_connect_cb // a client has connected
IP OK 192.168.178.20 // the IP address of the client
                     // ... waiting for a command...
tcp_recv_cb B2,Lon   // client sends a command: LED on
A2ProcMsg            // command has been processed
A2SendMsg Data sent, Txbu: ack
                     // acknowledgement to client
tcp_sent_cb          // successfully sent
                     // ... waiting for a command...
tcp_recv_cb B1,Lof   // client sends a command: LED off
A2ProcMsg            // command has been processed
A2SendMsg Data sent, Txbu: ack
                     // acknowledgement to client
tcp_sent_cb          // successfully sent
tcpclient_discon_cb  // client has disconnected
```

We can now clearly see how things are structured. Heavy use is made of callback routines: for example, the init callback is called when the ESP8266 is ready for normal operation, and in particular when the WLAN interface is operating correctly. The event handler is also a callback. Available events include the following which are of interest to us.

- `EVENT_STAMODE_CONNECTED`: the ESP8266 is connected to the router in 'station mode';
- `EVENT_STAMODE_DISCONNECTED`: the connection to the router has been broken;
- `EVENT_STAMODE_GOT_IP`: the ESP8266 has received an IP address from the router.

In this callback, as in all the other callbacks we shall look at, there is no significant processing done in the routine: all the callback code does is set a global flag variable.

The real processing work is done in the state machines `DoApp2()` and `DoApp3()`, called from within `workloop()`. Here `DoApp2()` carries out the WLAN-specific functions while `DoApp3()` is concerned with controlling the LEDs. The function `workloop()` is not the conventional infinite loop. Instead, after calling the two state machine functions in turn, it terminates; but not before asking the operating system to call it again after a short delay. This means that its execution will never block and that control is returned to the operating system. **Listing 1** shows the structure of this 'infinite loop' in FreeRTOS.
In the course of operation the following further callback functions, defined in our code, are set up.

- `tcpclient_connect_cb()`: a client has connected;
- `tcpclient_disconnect_cb()`: a client has disconnected;
- `tcp_recv_cb()`: a TCP message has been received;
- `tcp_sent_cb()`: a TCP message has been successfully sent.

Again, these functions simply set a flag variable to indicate that the event has occurred, and the real work is deferred to `DoApp2()` and `DoApp3()`. **Listing 2** shows by way of example the function `tcpclient_connect_cb()`, which is called when a client connects. It simply enables the message transmit and receive callback routines and records the new event. The `espconn_*()` functions are implemented in the library supplied with the SDK. `DoApp2()` and `DoApp3()` operate using cooperative multithreading. After global initialization in `user_init()` is complete the main loop function `workloop()` is started, where the threads

are called in turn. They are structured as state machines, where the code sequences, which we will call 'threadlets', are executed depending on the values of the status variables. If a threadlet completes execution successfully it updates the switch variable so that on the next call a different threadlet will be executed, depending on the tasks that need to be done. Print commands are used to log the execution of the code: see **Listing 3**.

The client commands defined for this application are listed in **Table 1**, along with the action taken by the ESP8266 on receiving each. The client UI includes buttons and sliders, and <value> can be any number from 0 (representing LED off) to 255 (representing the LED at full brightness). Further details can be gleaned from the source code, which is available for download as a complete Eclipse workspace from the Elektor project web pages [10].

The stand-alone approach works just as reliably as the combination approach described at [1] using an STM32 microcontroller.

Conclusion and outlook

In the form of the ESP8266 the maker community has access to a device that can be used to build small- to medium-complexity IoT projects without the need for a host microcontroller. Furthermore, development tools are available to suit (almost) every taste. But this is by no means the end of the story, as the ESP8266's bigger brother has already been announced: watch this space!

(150094)

Table 1. Commands and corresponding actions taken by the ESP8266.

Operation	Command from client	Resulting action
Button 1	B1, Lof	Red LED off
Button 2	B2, Lon	Red LED on
Button 3	B3, Lbl	Red LED flashes
Button 4	B4, Lzz	Red LED flashes quickly
Slider 1	S1,<value>	Set brightness of blue LED according to <value>
Slider 2	S2,<value>	Set brightness of white LED according to <value>

Web Links

[1] WLAN for microcontrollers (Elektor 1/2016): www.elektormagazine.com/150093

[2] The Lua scripting language: www.lua.org

[3] NodeMCU: https://github.com/nodemcu/nodemcu-firmware

[4] ESPlorer: http://esp8266.ru/esplorer/

[5] Arduino IDE: https://www.arduino.cc/en/Main/Software

[6] Arduino IDE extension: http://arduino.esp8266.com/versions/2.0.0/package_esp8266com_index.json

[7] Neil Kolban's e-book: http://neilkolban.com/tech/esp8266/

[8] Eclipse-based development system: www.esp8266.com/viewtopic.php?f=9&t=820

[9] FreeRTOS: http://www.freertos.org/

[10] Project page: www.elektormagazine.com/150094

Advertisement

The Newest Products for Your Newest Designs®

MOUSER ELECTRONICS

More new products in stock than any other distributor.



Order now at mouser.co.uk

Master / Slave LED Striplight Dimmer Unit

With optional microcontroller drive

LED striplights can be bought in various lengths at ridiculously low prices and provide bling & glitz. Anyone can hang up a few of these and make them light, but you should feel challenged to add proper dimming or the ability to control a few strings at large distances like 300 feet away. Or hook up an Arduino. How big is your party?

By **Barry Sandeman** (South Africa)

This project was created from a need to dim multiple LED strip lights at the same time. The unit triples from being a single 10-m LED striplight dimmer to a Master and Slaves network, and finally it's controllable from a little computer that has PWM output having 5–12V swing. Referring to the circuit diagram in **Figure 1** there are four sections that afford these functionalities:

- The onboard controlled PWM manual (i.e. pot operated) controller comprising U1.B and surrounding components;
- The remote input consisting of components SK1, U2 and U1.D;
- The output to a remote unit via U1.C, U1.A and Q2 with surrounding components and SK2 as the output connector;
- The driver section comprised of the MOSFET and SK2.

Powered off the +12 to 15 V supply rail, LED D10 acts as the Power On indicator.

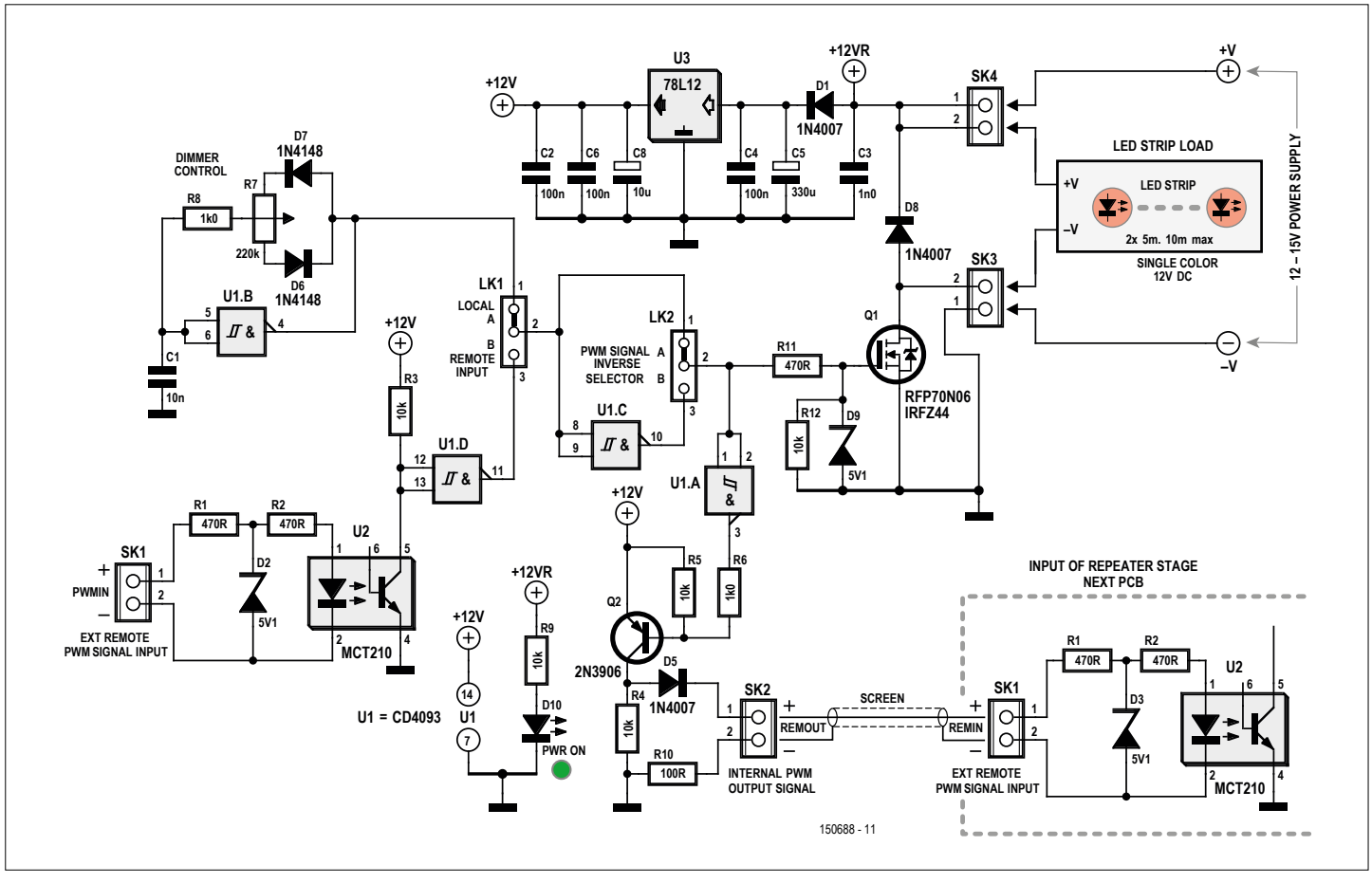


Figure 1. Schematic of the dimmer/repeater for 12-V mono color LED stringlights. Through wire links at various places the circuit permits a choice of local/remote control, master/slave configuration, and optional microcontroller driving.

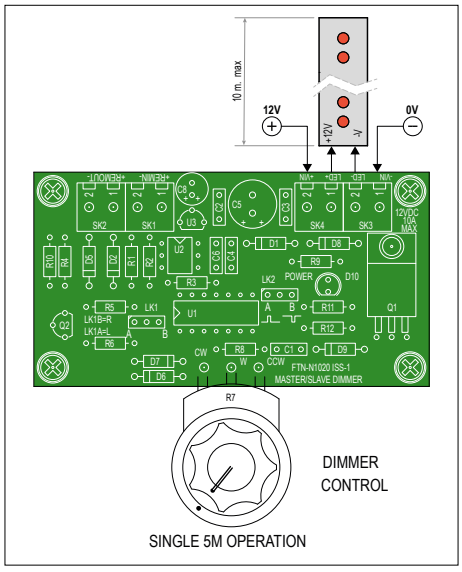


Figure 2. Connecting up for local control.

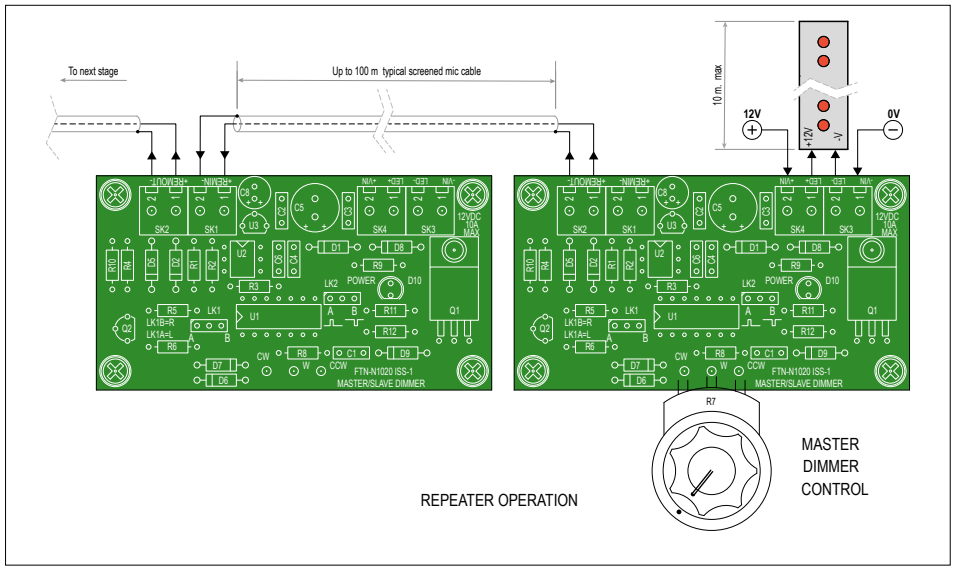


Figure 3. Board "interwiring" for repeater (Master/Slave) operation. The boards can be up to 100 m (300 ft) apart so how big is your gig?

Configurations and options

The circuit permits a number of different modes of operation, selectable with wire links (jumpers).

- Manual control is enabled with potentiometer R7 fitted, and controls the LEDs in the string(s) connected from 1% to 99% brightness approximately. This 'local' mode (**Figure 2**) is selected with link LK1 in position A, and LK2 in position A also. The signal is then passed on to MOSFET Q1 and onto the LED striplight(s) (10 m max., or 2 × 5 m, 12 VDC, single color).
- Extended control to a second dimmer unit is with link LK1 in position B (**Figure 3**). This has a single core shielded microphone cable (up to 100 m length) connected between the Master and the Slave.
- Wire link LK2 allows the dimming to be reversed between Master and Slave. Each Node must have its own local power supply up to 10 amps at 12-15 volts DC.
- Connecting a Master and Slave is done via Sk2 on the Master unit to Sk1 on the Slave unit. As each node is a repeater, no extra power supply is needed.

Note that the modules are also electrically isolated via the opto isolator U2.

Printed circuit board

The CAD files for the author-designed printed circuit board can be downloaded from the Elektor Magazine website [1]. The PCB design produced using the vintage 'Pcad' photo plotting program is provided as guidance only, respecting that the PCB CAD wizards and tweakers among our readers might wish to work on the design — which BTW is the aim of all Reader's Projects in this here the *Design* section of Elektor Magazine.

An early prototype of the board is shown in **Figure 4**. Having since tweaked the design in a few places, the author is having the PCB made professionally as a single sided solder masked and silk-screened board as pictured in **Figure 5**, with 10 PCBs per panel.

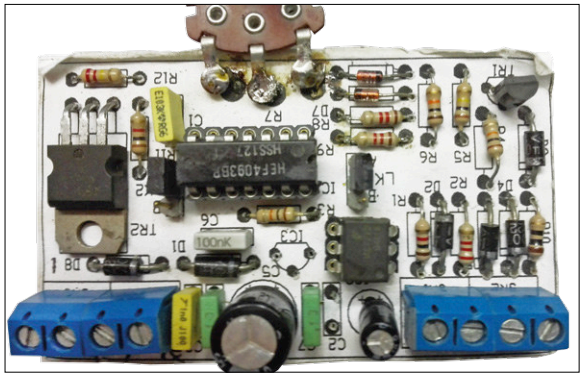


Figure 4. An early, well-used, prototype of the dimmer board with a white decal used for the component map.

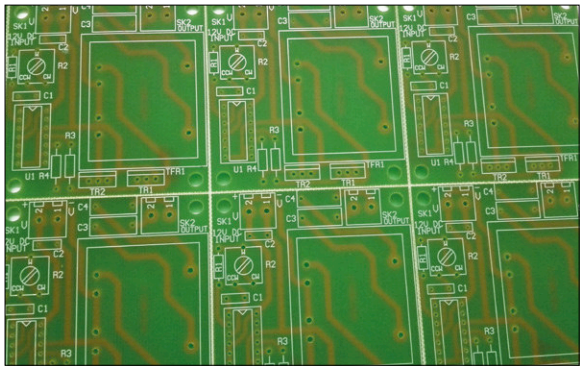


Figure 5. Panel with 10 dimmer boards.

The challenge: Arduino'd control

This project does not need a computer or Bluetooth type units for remote control, as this will really add to the costs. Although aimed at beginners, the project actually has real world applications like controlled emergency lights, cinema stair lighting and many more.

Computer control can be used as the PWM output signal is then connected to SK1 with LK1 in position B. The lights can then really be controlled from an Arduino or similar processor, and has the potential to control six separate lines. The possibilities are then endless. ⏮

(150688)

Web Link

[1] Author's PCB design files (Pcad): www.elektormagazine.com/150688

Baroduino



Arduino'd 48-hour recording barograph feat. storm alert

By **Daniel Coppens** (Belgium)

Possibly one of the most important environmental properties to know while sailing is atmospheric pressure. This project tracks the measured air pressure over a 48-hour (max.) period and shows it on a 2.2" TFT. This helps you keep track of sudden changes, so you are able to navigate to a safe haven before the storm starts.

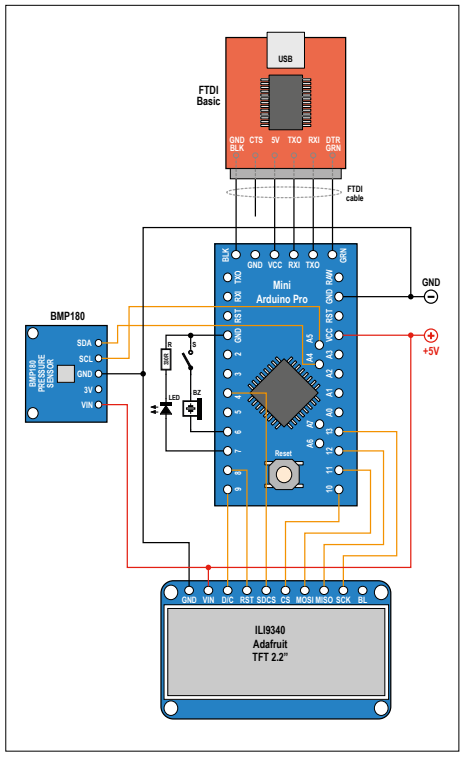


Figure 1. The basic components and how to interconnect them. Mind the connection to the optional FTDI Basic board for programming the firmware.

This project results in an electronic barograph displaying the instantaneous atmospheric pressure and the graph of its evolution over the past 48 hours, on a 2.2" TFT screen. Since it is primarily aimed at sailors, it also displays the pressure evolution over the past hours. A sound (buzzer) and visual alarm (flashing LED) flag a pressure drop exceeding 140 Pa h^{-1}, the latter being a sign of upcoming strong winds.

How it's built

Baroduino consists of a mere three major components. At the heart is the Arduino Pro Mini 5V; its interface consists of a 2.2-inch TFT screen with Micro SD card holder from Adafruit, and completing the setup is a BMP180 breakout board pressure sensor from Sparkfun. **Figure 1** shows the setup.

For best flexibility, the power source can be supplemented with a stable 5-V power supply based on the LM2574, as shown in **Figure 2**. This allows for any DC power source between 5 V and 40 V to be used. Alternatively, a power source of up to 12 V can be directly connected to the Arduino RAW pin. The Arduino Pro Mini is equipped with a voltage regulator that steps down and stabilizes this voltage to the microcontroller's operating voltage. This voltage is also put on the Arduino's 5V pin, available for powering small loads and ideal for supplying the pressure sensor and TFT screen in our circuit with power.

The alarm function is optional and only requires a piezo type buzzer, an LED with its appropriate current limiting resistor, and a switch to turn the alarm off. It is not necessary to adapt the software if you do not connect the LED and/or the buzzer.

The BMP180 pressure sensor also measures the temperature, so why not display it on the screen as well? The added value is limited though, since it measures the temperature right where the sensor is installed, which may not necessarily represent an accurate figure.

And that's all there is to it! ...except for the software of course.

The (soft)core of the circuit

As you may have noticed, the hardware side of this project is very simple. But

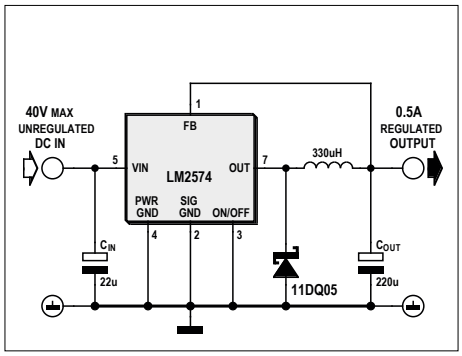


Figure 2. If the available board voltage exceeds 12 V, this circuit is an adequate addition to the circuit, allowing Baroduino to operate from supplies up to 40 volts.

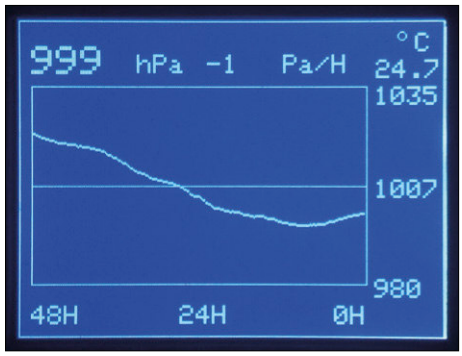


Figure 3. The full 48-hour graph is of course only shown after 48 hours of logging!

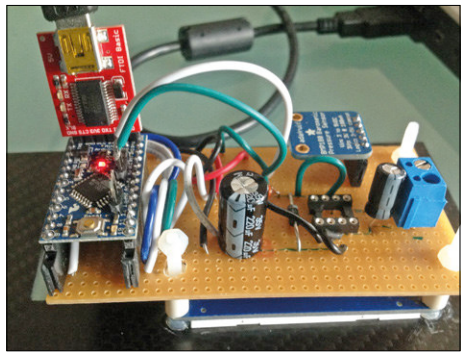


Figure 4. The author's circuit was built on a piece of perfboard. The TFT screen is mounted on the bottom side for easy assembling the circuit in a case.

since there is a microcontroller involved, software — or better, firmware — is mandatory. The Arduino Sketch, as the code for Arduino is named, is slightly more complicated than the hardware.

What happens basically is that the atmospheric pressure and temperature are read from the sensor every 5 seconds. This raw data is then processed and calibrated. The routine to do this is based on the work of Jim Lindblom, whom I like to thank here for his efforts. The calibrated pressure is then displayed on the TFT screen as well as saved on the Micro SD card. A 2-GB Micro SD card is sufficient and should be FAT16-formatted before use.

All further calculations of pressure changes and displaying the graph are started by reading the data from the Micro SD card. The resulting graph is refreshed every 60 seconds. Obviously the graph will only be displayed over the total width of the screen after 48 hours of data capturing, see **Figure 3**.

Software libraries

The Sketch developed for the project uses a certain number of 'standard' libraries specified at the beginning of the code. Besides these, two libraries from Adafruit are needed: Adafruit_GFX.h and Adafruit_ILI9340.h. These can be downloaded from GitHub, see [1] and [2]. It is necessary to install them before they can be used in the Arduino development environment or 'IDE' [3]. The installation process is described in detail at [4].
Make sure to have all libraries available before loading the Sketch into the Arduino development environment.

Setting it all up

After wiring up the hardware, downloading the Sketch [5] and opening it in the Arduino IDE, the Sketch can be uploaded to the Arduino board and run. The Pro Mini has no USB connection on board, so an 'FTDI cable' is needed to connect it to your PC. If you use an Arduino with onboard USB, you can connect it directly to your computer and start programming the microcontroller on the board.

After loading the sketch, the instantaneous pressure (and temperature) will promptly appear on the TFT screen and a graph will slowly cover the total width of the screen over a period of 48 hours. If you're stuck for some reason or would like to tech-discuss this project, please visit our Labs website and join the discussion [6].

A suggestion for assembly is shown in **Figure 4**. The author's circuit was built on a piece of perfboard. The TFT screen is mounted on the bottom side for easy assembling the circuit in a case. ⏮

(150425)

Web Links

[1] Libraries on Github: https://github.com/adafruit/Adafruit-GFX-Library
[2] Libraries on Github: https://github.com/adafruit/Adafruit_ILI9340
[3] Arduino Development environment: www.arduino.cc/en/Main/Software
[4] How to install a library: www.arduino.cc/en/Guide/Libraries
[5] Project web page: www.elektormagazine.com/150425
[6] The project on our labs website: www.elektor-labs.com/node/4449

What you need:

Mandatory:

- Arduino Pro Mini 5V 16 MHz. (or equivalent, but watch the pinout)
- FTDI breakout board to load the Sketch into the Arduino (for example SparkFun FTDI Basic Breakout - 5V)
- Adafruit 2.2" TFT ILI9340 screen with Micro SD card holder
- Micro SD card 2GB, formatted in FAT16
- Sparkfun BMP180 sensor
- 2 switches to turn on the barograph and turn off the alarm
- 2 LEDs, green for power and red for flashing alarm
- 2 330-ohm resistors for LEDs
- 1 piezo buzzer

Optional:

- Regulated 5-V power supply using:
- LM2574
- 22-µF and 220-µF capacitors
- 11DQ05 diode
- 330-µH Inductor

A Simple Current Sensor Probe

Make floating current measurements with ground-referenced 'scopes



By **Alfred Rosenkränzer** (Germany)

If the multimeter is the most important test and measurement device for electronicists, there can be no dispute that the oscilloscope comes next. Its application is not only universal but also highly flexible, its optical display revealing exactly what's going on in dynamic signals or voltage curves. But herein lies the problem: unlike a multimeter, an oscilloscope just not innately equipped for measuring current flows. With this probe you can change that!

When you use a multimeter for measuring currents, as is well known, you are not directly measuring the actual current flow but rather the voltage drop across a (low-value) resistor that is located inside the multimeter and can be switched according to the desired measurement range. This works fine, as a multimeter is practically always battery-powered and is therefore not only floating (voltage-free) but also well insulated thanks to its plastic case. An even better-insulated variant uses so-called current clamps, in which you apply a folding metallic ring-core around one conductor and measure the resulting magnetic field by means of a Hall sensor (or if dealing with purely AC currents, alternatively via a winding).
Both methods have their drawbacks, however. Unfortunately they do not capture the dynamic aspect of currents that change, and straight current clamps are best suited to fairly large currents flowing through cables or their screening. Furthermore their useful frequency range is not particularly great. Capturing dynamic signals within circuitry is the domain for oscilloscopes — yet they are not really designed for this purpose. Here we need seek some extra help...

Shunts and oscilloscopes

Oscilloscopes are notoriously shy of strangers when it comes to measuring currents. With dynamic current flows we're generally interested in which selectable amps (or lower) range they fall, and above all the current-flows in the circuits. As a rule you wish to see how the current passing through a component behaves in relation, for instance, to a voltage at another location — or vice versa. Understandably current clamps are not particularly easy to use in this situation. More likely you will look for a pre-existing resistor to use as a shunt and simply measure the voltage drop across this in order to calculate the current, or else you unsolder one end of a component or disconnect part of the circuit and insert here a resistor of appropriate ohmage and wattage. Perfectly simple then — or what?

Unfortunately oscilloscopes have a few limitations. Some of them are not fully floating, which rules out the trick of simply measuring the current in a resistor as the difference between the signals of two normal oscilloscope probes. Far more problematic is that if you wish to measure the voltage referenced to the ground of the circuit under investigation using an oscilloscope with 'only' two channels simultaneously, then one end of the shunt or resistor through which the current flows must be connected to this ground. So you cannot avoid the need for potential-free (floating) and differential measurement of small voltages, which is precisely what you can do with the upgrade that the following circuit provides.

Differential and floating

The principle of a feasible solution will be obvious to the observant electronicist: you use a so-called instrumentation amplifier. This measures (usually in a high-impedance manner) voltage

Technical data

- Current sensor probe for oscilloscopes
- Common mode voltages:
  - min. ±200 V at ±5 V for IC1
  - up to ±600 V at ±15 V for IC1
- Common mode suppression:
  - min. 90 dB
- Measured voltages:
  - up to ±4.7 V at ±5 V for IC1
  - up to ±14.7 V at ±15 V for IC1
- Input impedance: 2 MΩ for voltage across shunt
  - 500 kΩ for common mode voltage
- Bandwidth: 0 up to 130 kHz (–3 dB)
- Amplification factor: x 1
- Amplification:
  - max. 0.001 % errors
  - max. 5 ppm nonlinearity
  - max. 5 ppm drift
- Offset voltage: max. 1 mV
- Noise:
  - max. 35 µVpp at 0.01 to 10 Hz
  - typ. 1.6 µV/√Hz at ≥100 Hz



differentials independent of the ground potential. Generally available circuits admittedly have a limited common mode voltage range. However, since the measurements we make on electronic circuits involve higher supply voltages than the customary ±15 V of op-amps we must come up with something different.

Analog Devices, the well-known supplier of analog ICs, provides a perfect solution in the form of a special operational amplifier with the type number AD8479. This special type has integral, laser-trimmed resistors within the chip. Figure 1 shows the functional diagram of its inner workings. The device operates as follows: 1 MΩ resistors on each of the negative and positive entry points to the device are connected to the corresponding inverting and non-inverting inputs of the internal op-amp. Unlike normal instrumentation amplifiers, the resistance of the non-inverting input versus ground (Pin 5) is very low and divides the input voltage (and thereby also the common mode signal) by 1/60. The resistance between the inverting input and ground (Pin 1) is equally low and trimmed in such a way that the common mode signal is suppressed extremely well — by 90 dB according to the data sheet [1]. The outcome is that the differential voltage is amplified simply and accurately, and the common mode range rises theoretically to a 60-fold value.
From this we acquire an extended common mode range, as visualized in the graphic in Figure 2. A power supply of ±5 V enables a common mode range of noticeably greater than ±200 V (area outlined in red). That should be fully sufficient for practical measurement purposes, as probes and oscilloscopes are normally equipped for high voltages of this order. Using a supply voltage of ±15 V the permissible common mode range is extended up to ±600 V (area outlined in blue) and is at this voltage limited not by the voltage divider in the IC but by the distance between pins and the IC housing.

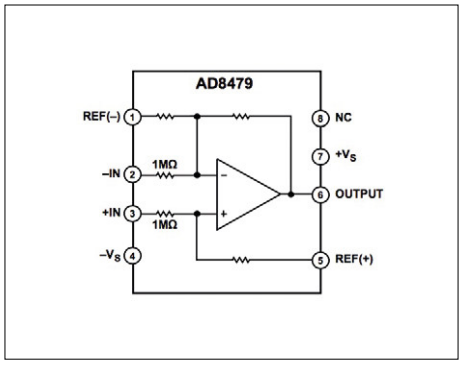


Figure 1. Block diagram of the AD8479 with integrated resistors.

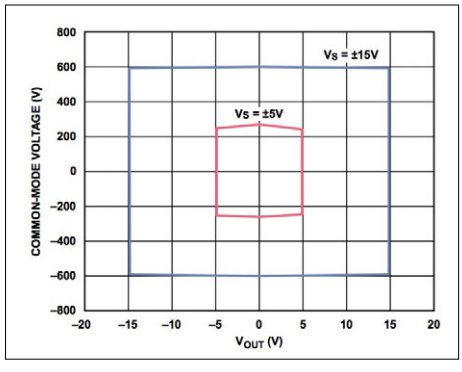


Figure 2. Common mode range using supplies of ±5 V (red) and ±15 V (blue).

Probe circuit

The fact that (in principle) almost everything is already included in the AD8479 makes the circuit of Figure 3 very simple. IC1 needs to be provided with merely the capacitors necessary for decoupling (C1 to C4). After inserting a series resistor of 50 Ω (R1) on the output for connection to a 50 Ω c coaxial cable, our work is done. In order to maintain the excellent characteristics of the op-amp in practice as well as in theory you should provide a clean and interference-free supply voltage to the stabilizing ICs in the probe — for understandable reasons as close to the AD8479 as possible. The two voltage regulator ICs ensure that you can then run the circuit from unstabilized ±9 to

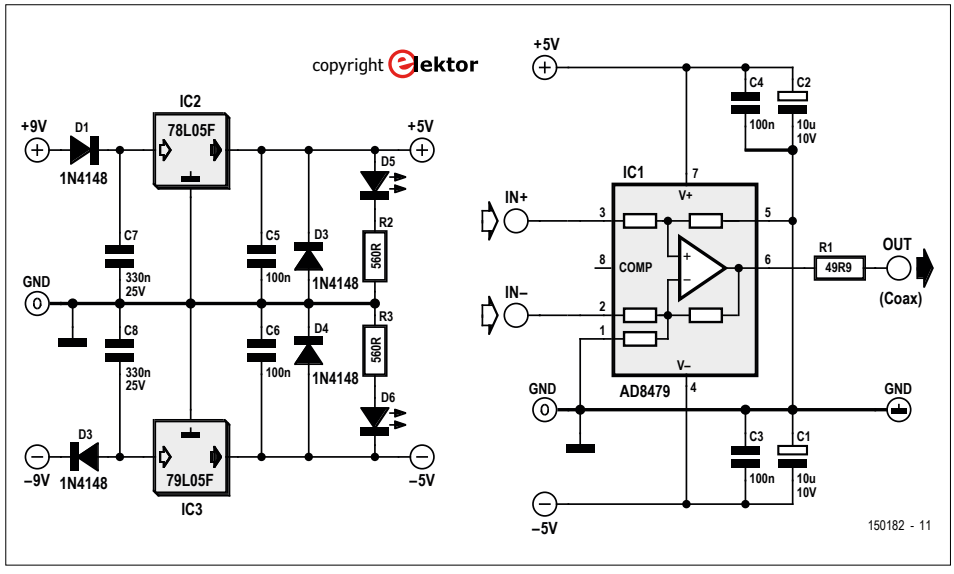


Figure 3. Circuit of the current probe including voltage stabilizing.

25 V. The current consumption of IC1 is minimal, which is why SMD versions are adequate for IC2 and IC3. D1/D2 serve for protection against polarity reversal whilst D3/D4 prevent any so-called latch-up of the voltage regulators, in case the two voltages drop out unequally and thereby risk causing polarity reversal of one output.

If the slightly over ±200 V common mode range versus ground is insufficient, you can also power IC1 with higher voltages. With ±9 V you already have a common mode range of ±400 V and at ±15 V you reach the maximum range of ±600 V. It is vital to ensure that the unstabilized supply voltage is in each case always 4 V higher/lower than the output voltage of the voltage regulator. In this case C1 and C2 need to be replaced by specimens with a voltage rating of 25 V too. With unstabilized voltages above 25 V the voltage rating of C7 and C8 should be raised to 35 V.

Component List

Resistors
All SMD 0603
R1 = 50Ω
R2,R3 = 560Ω

Capacitors
All ceramic
C1,C2 = 10µ 10V*, SMD 1206
C3..C6 = 100n 50V, SMD 0603
C7,C8 = 330n 25V*, SMD 1206

Semiconductors
D1–D4 = 1N4148, DO214AA
D5 = LED, red, SMD 0805
D6 = LED, blau, SMD 0805
IC1 = AD8479, SOIC-8
IC2 = 78L05F*, SOT89
IC3 = 79L05F*, SOT89

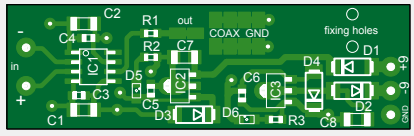


Figure 4. Component layout plan for the PCB. The two pairs of holes on the right-hand side are for securing the connecting cable using cable ties.

Miscellaneous
Coaxial cable*
Mikrophone cable*, 2-wire, shielded
Pin probes*
PCB

* see text



Figure 5. The PCB end of the coaxial cable looks like this when correctly stripped and tinned.

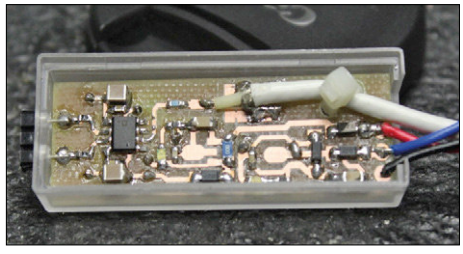


Figure 6. An early prototype of the current probe, houses in a small plastic enclosure. The coaxial cable is fixed only with cable ties here.

Construction

Mounting SMD ICs on a scrap of perf board is not going to be easy, even if the circuit itself is designed with laudable simplicity. Accordingly the author designed a suitable printed circuit board (see Figure 4), the layout file of which is available for free download at the web page of this article [2]. And given that the op-amp comes in SMD format, SMDs can also be used for the other components, which is all to the benefit of the miniaturization of the PCB and consequently its practicality as a shunt probe. Even if the circuit is far from complex and not very many components require to be placed, you should still give yourself time for the soldering task. A magnifying lens is extremely helpful as a couple of the capacitors and resistors belong in the 'diminutive' category 0603.

When you examine the PCB the two circles on the upper right-hand side will be conspicuous. These are guide markers for making openings through which you will restrain the output coaxial cable using narrow cable ties. Although the current probe has a restricted bandwidth of only 130 kHz, on account of the AD8479 IC, the choice of coaxial cable is completely rational for the purpose of good signal quality and serviceability. On account of the restricted bandwidth the oscilloscope input must not be terminated with 50 Ω, but can remain high-impedance with the usual 1 MΩ, meaning the signal is then not even reduced by half.

For soldering the coaxial cable directly onto the PCB we proceed as follows. At its free end we first remove 11 mm of the outer insulation, after which the screening/braiding layer now visible is shortened to 6 mm and tinned. The inner insulation is removed far enough for the bare-metal inner conductor to extend out for 3 mm. Figure 5 shows what is intended. Next we lay the now prepared cable end onto the

matching position on the PCB and first solder the inner conductor. Afterwards comes the braided screen with plenty of solder on the two elongated ground pads. For the power lead a flexible microphone cable with two inner leads and screening serves well. This is soldered to the three pads at lower right. To the input we connect standard off-the-shelf test prods; it is not absolutely necessary for their short leads to be screened, since the source impedances are customarily low. You can also make the prods pluggable, for instance using scraps of pin and receptacle header strip with a 5.08 mm (0.2 inch) pitch.

After the PCB has been fitted into a small plastic lidded box and the cable passed through the corresponding holes, the free end of the coaxial cable can be fitted with a BNC plug. The connector for the supply voltage depends on the power source employed. It doesn't need to handle much power, as only a few mA are necessary.

And finally

Figure 6 shows an early prototype of the current probe. Once constructed and put into an enclosure nothing else stands in the way of putting it to use. There's nothing you need adjust nor anything to configure. All you should note is that only voltages across shunts of up to ± 4.7 V can be measured when IC1 is powered with ±5 V for simple amplification, as the output voltage is exactly the same size. In special situations, if you require a greater voltage swing (which is hardly to be expected in current measurements), then you must raise the supply voltage of IC1 as described earlier and at the same time keep an eye on the voltage stability of C1 and C2.

All in all, this probe is an enhancement for every oscilloscope, extending its usability for problem-free measurement of current. For its relatively minor cost you gain comparatively great usability.

(150182)

About the author

Alfred Rosenkränzer has worked for 32 years as a development engineer, initially in the field of professional television technology. Since the end of the 1990s he has developed digital high-speed and analog circuitry for IC testers.

Web Links

[1] AD8479 datasheet:
www.analog.com/static/imported-files/data_sheets/AD8479.PDF

[2] Elektor Magazineweb page with PCB layout files:
www.elektormagazine.com/150182

Low-Battery Indicator

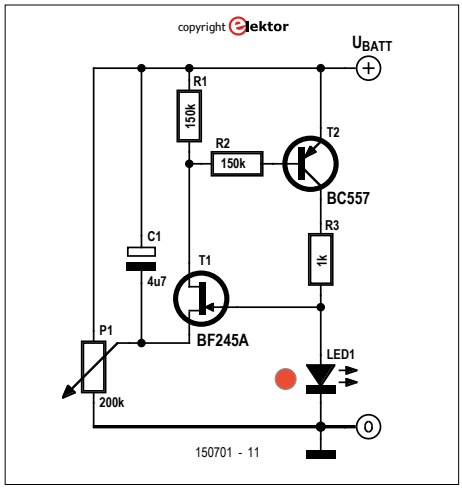
for 9 V Batteries



By **Dr. Rainer Giedigkeit** (Germany)

For many years now I have been using the electret microphone amplifier described in Elektor Magazine in the distant past as part of my loudspeaker measurement system. So that I don't need to guess the state of charge of the battery for reproducibility of test results, I have retro-fitted the amplifier circuit with a low-battery indicator. As you can see from the schematic in **Figure 1** there isn't a microcontroller or even an opamp in sight.

The red LED has a dual function in this circuit; not only does it act as an indicator to let you know that the battery voltage is low but the forward voltage drop across the LED is used as a reference voltage level. The J-FET T1 effectively operates as a comparator, comparing this reference voltage with the divided-down battery voltage level at the wiper of P1. T1 will conduct when the voltage falls below the threshold level set on P1. T1's drain current passing through resistor R1 lowers the voltage at the base of the PNP transistor T2 and brings the transistor into conduction. R2 limits T2 base current. The resulting emitter-collector current through T2, limited to 7 mA by resistor R3 is sufficient to light LED1. C1 causes the LED to blink when the battery voltage falls below the threshold level. The blink frequency increases as the battery voltage falls further. At about 1 V below the voltage threshold, LED1 is lit continuously. If it falls to 3 V below the threshold, it starts to dim.

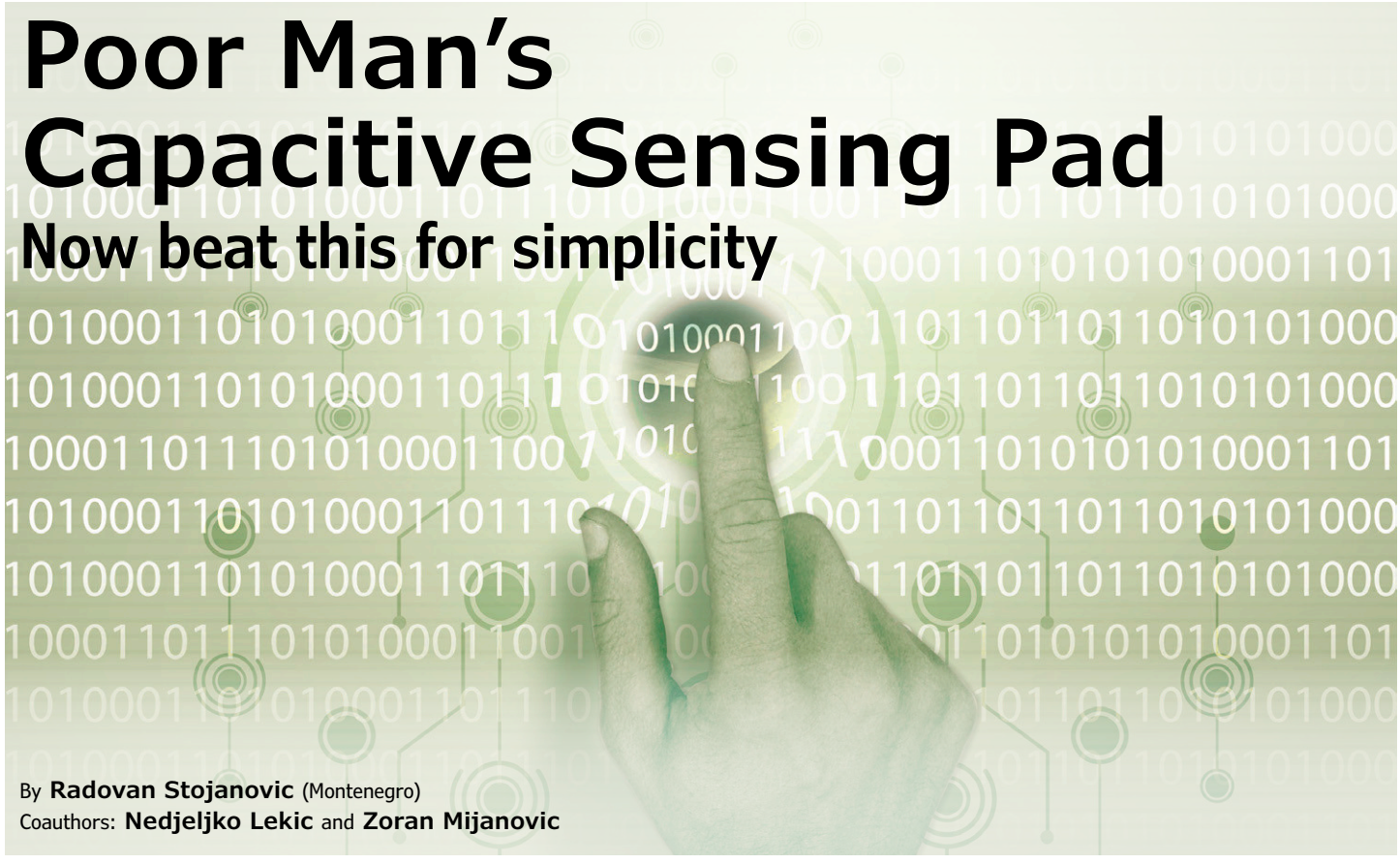


Using a 100 kΩ pot for P1 alters the circuit behavior. Now when the battery voltage drops below the threshold, the LED lights continuously rather than blinks. The switching threshold also now exhibits hysteresis because as LED1 starts to shine the increased current raises its forward voltage drop and therefore the voltage reference level. ⏮

(150701)

Poor Man's Capacitive Sensing Pad

Now beat this for simplicity



By **Radovan Stojanovic** (Montenegro)
Coauthors: **Nedjeljko Lekic** and **Zoran Mijanovic**

This design note is likely to specify the simplest ever capacitive touch pad for use with microcontrollers or FPGAs. The resulting design is cheap, extremely practical, and robust enough for many applications.

In single-touch configuration the sensing pad is usually connected to a bidirectional microcontroller ('MC') or FPGA pin P_n via resistance R, as shown in **Figure 1**. The logic direction of the pin, i.e. input or output (I or O), is determined by the DDR register, while the input state is read by the IN register. Here, C_p is a pin capacitance of about 5 pF and 4 pF, for microcontrollers and FPGAs respectively. Element C_b represents the body-to-ground capacitance — it can be as much as 400 pF but typically is in the range 100-150 pF. Elements C_{x0} and C_{x1} are the capacitances of the untouched and touched pad respectively.

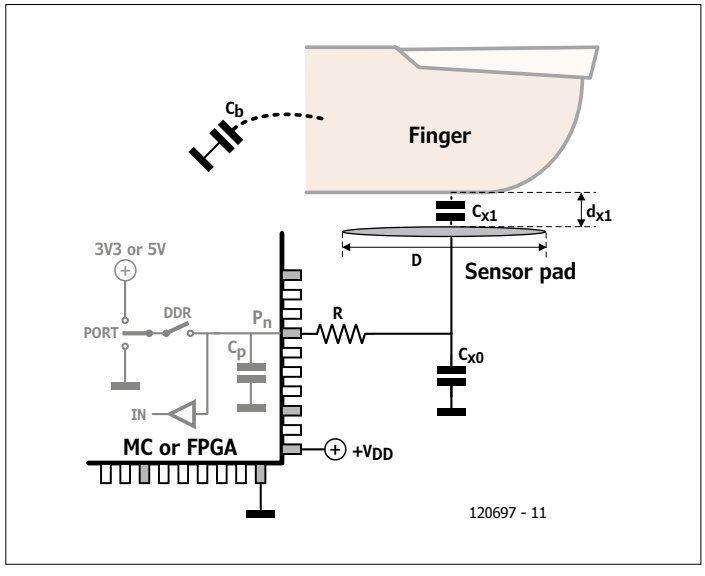


Figure 1. Adding a capacitive pad to a microcontroller or an FPGA.

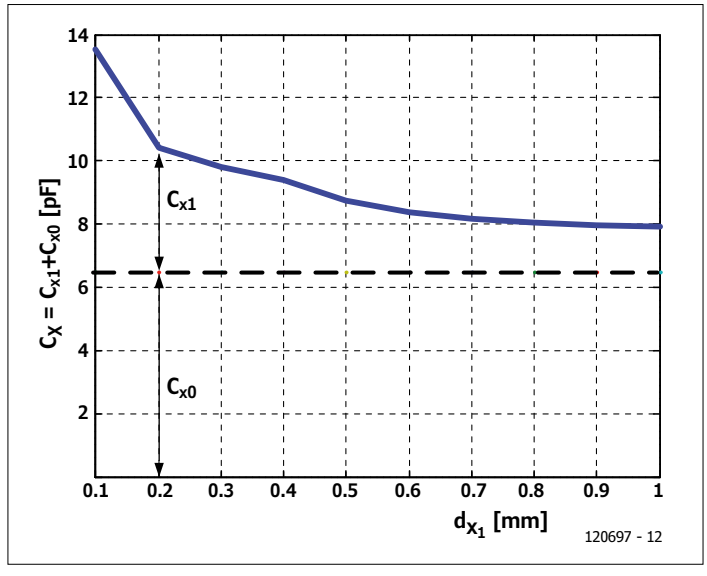


Figure 2. Equivalent capacitance of finger and pad as a function of distance d_{x1}, for D = 10 mm.

Usually, $C_b >> C_{x1}$, and $C_x \approx C_{x0} + C_{x1}$. With C_{xo} constant at 6.5 pF for D=10 m, the graph in **Figure 2** shows C_{x1} decreasing from 7 pF down to 1.5 pF for 0.1 mm<d_{x1}<1 mm.

Principle of capacitive sensing

The sensing process may be compared to the pouring of an amount of water from one bowl to another, as illustrated in **Figure 3**, where arguably the water level corresponds to voltage (V), the size of the bowl to capacitance (C), and finally the amount of water to electrical charge Q.

The CHARGE phase starts by setting a logic 1 (i.e. 3.3 V or 5 V) to the relevant pin P_n for a time t_c=0.75-2 µs, i.e. long enough to charge C_p and C_x to logic High voltage levels V_1=V_2=3.3 V. During the DISCHARGE phase, for t_d between 200 ns and 750 ns, C_p is fully discharged, while V_2 drops by time constant RC_x, eventually reaching the value $V_{2(td)}$ with remaining charge $QC_{x(td)}$. During the SHARE phase, for the duration of t_s, pin Pn switches to the high impedance mode (Z mode) and $QC_{x(td)}$ shares (or 'distributes') proportionally to C_p and C_x. At the end of the sharing, $V_1 \approx V_2$, $i \approx 0$, when the MEASURE phase begins, voltage V_1 reaches the level below or above the pin's threshold V_{TR} depending on whether the pad is touched or not. In digital speak, this equals logic 1 or 0 for the corresponding bit within the IN register.

Finding an optimal value for R is an additional issue. According to our experiments, R=68 kΩ is the optimal value for V_{DD}=3.3 V, while 47 kΩ is best for 5 V. Additionally, R protects the FPGA pin from static discharges up to about 1,000 V.

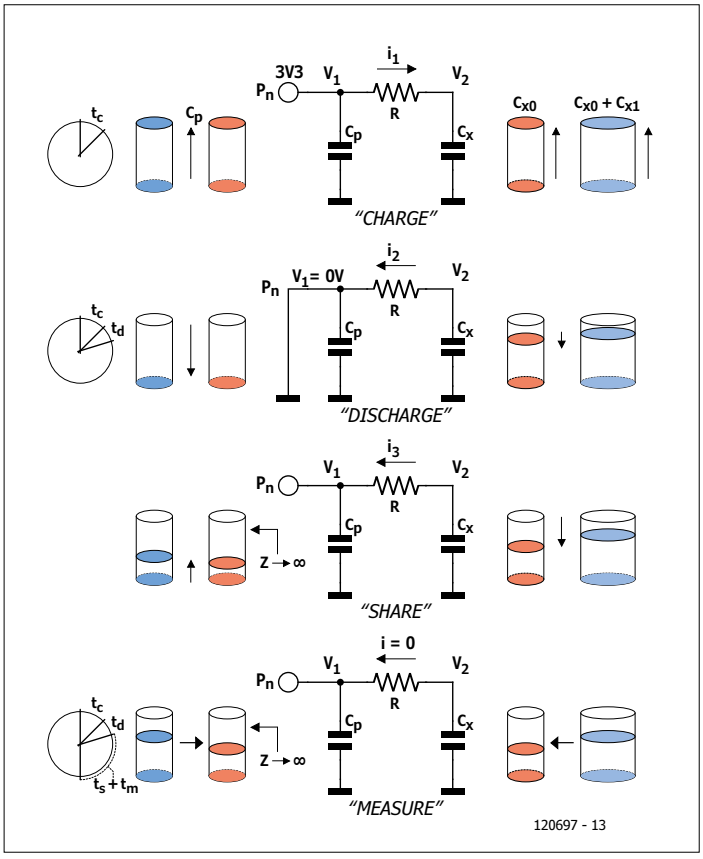


Figure 3. The four phases of capacitive sensing.

Software support

The overall sensing process is driven by software routines. Here, the codes for single pad configurations are given. In case of a microcontroller, the pseudo code in WinAVR GNU GCC compiler format is shown in **Listing 1**. For the target chip we've selected an ATMega16 from Atmel's family of 8-bit RISC micros. In the case of an FPGA, the single-pad cap interface is a 3-terminal port with clock (CLK), sensor input (S) and sensor status (SS) signals. In essence, we're looking at a counter with a comparator. The generic constants CHARGE_TIME_NS, DISCHARGE_TIME_NS, START_READ_NS, correspond to the t_c, t_d and t_s+t_m periods respectively, while PERIOD_NS marks a read cycle. The VHDL code is given in **Listing 2**.

Experimental results

The proposed approach was verified in practice on an Atmel ATMega16 microcontroller and an EP1C6Q240C8 FPGA chip

Listing 1: Pseudo code for microcontroller-driven implementation of capacitive pad

```
// MC CAP PAD PSEUDOCODE
//"DISCHARGE" phase.
// Initially PORT=1.
PORTC &= ~TOUCH;  // PORT=0
short_delay();  // Delay 750ns

//"SHARE" phase.
DDRC &= ~TOUCH; // DDR=open. Touch port set as input
short_delay();  // Delay of 750ns

//"MEASURE" phase
if(PINC & TOUCH) PORTD |= LED; // If PIN is high turn on LED
else PORTD &= ~LED; // else turn off LED

// "CHARGE" phase.
PORTC |= TOUCH; //Touch port=1
  DDRC |= TOUCH;  // DDR=close. Touch port set as output
```

from Altera's Cyclone series. After the code writing, compilation and debugging, the micro and the FPGA chips got programmed/configured in single-pad configuration. The pad of 10 mm diameter is isolated with a 0.1 mm thick paper layer. As shown in **Figure 4a**, in real measurements the oscilloscope probes were connected to the ends of resistor R, introducing an additional capacitance C_{op} of 15 pF in parallel to C_x and C_p. We used an Agilent Tech type DSO3102A 'scope and probes type N2862A. Considering the 7 pF or so capacitance of C_{x1} it significantly reduces the sensing effects. To have more pronounced effects on the 'scope screen, the charging, discharging and sharing times are increased to t_c=24 µs, t_d=2.5 µs and t_s+t_m=15 µs, respectively. The traces in **Figure 4b** present waveforms equivalent of the equivalent circuits in Figure 3. The presence of a finger is reliably detected. In real life situations the effect should be even more pronounced.

A microcontroller can excite and read several pads, for example, 8 or 16 pads connected to PORTB and/or PORTC. The algorithm remains almost the same, because the capacitances are read in parallel in one go as the 'state' of the port. But in terms of parallel processing, an FPGA is a better solution. A single pad typically occupies only 60 LEs, (Logical Elements) of an FPGA. In an Altera EP1C6Q240C8 Cyclone that's a mere 1 % of total number of available LEs (5,980 LEs available). Sets of 2, 4, 8, 16, 32 and 64 caps occupy 140, 151, 168, 213, 297 and 462 LEs. It can be noted that the design is very silicon-effective and, as an example, a 64-pad configuration occupies only 7.7 % of the EP1C6Q240C8's resources and 72 pins out of 240. In practice, an *N*-input capacitive interface may be embedded in a low-capacity FPGA, the only limitation being the number of available FPGA pins.

The design is also noise immune to a good extent. Because of the parallel processing the sampling period does not increase with the number of pads, and can be adjusted from 5 µs to 20 µs to match real life situations. The input impedance of pin P_n has a low value during the CHARGE" and DISCHARGE phases, and is subject to noise effects to a degree during the SHARE and MEASURE phase only (High *Z*). This makes the sensor pretty resilient to low-frequency noise such as 50 Hz or 60 Hz AC powerline noise, which in practical terms is only

Listing 2: VHDL code for capacitive sensing.

```
-- FPGA CAP SENSOR, VHDL CODE
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY FPGA_tri_state IS
generic (CHARGE_TIME_NS       : integer := 144405;
         DISCHARGE_TIME_NS    : integer := 4545;
         START_READ_NS        : integer := 10000;
         PERIOD_NS            : integer := 400000
         );
    PORT(
            CLK    : in std_logic;
            S  : inout std_logic;
            SS : out std_logic
        );
END FPGA_tri_state;

ARCHITECTURE arh OF FPGA_tri_state IS

constant clock_cycle  : integer := 21;
constant one_time: integer := CHARGE_TIME_NS / clock_cycle;
constant zero_time    : integer := (CHARGE_TIME_NS + DISCHARGE_TIME_NS) / clock_cycle;
constant read_time    : integer := (START_READ_NS + CHARGE_TIME_NS + DISCHARGE_TIME_NS) / clock_cycle;
constant max_time     : integer := PERIOD_NS / clock_cycle;

SIGNAL  b  : STD_LOGIC; -- DFF that stores feedback value.
shared variable counter : integer range 0 to PERIOD_NS/clock_cycle :=0;
signal dir      : std_logic;

BEGIN
    PROCESS(CLK)
```

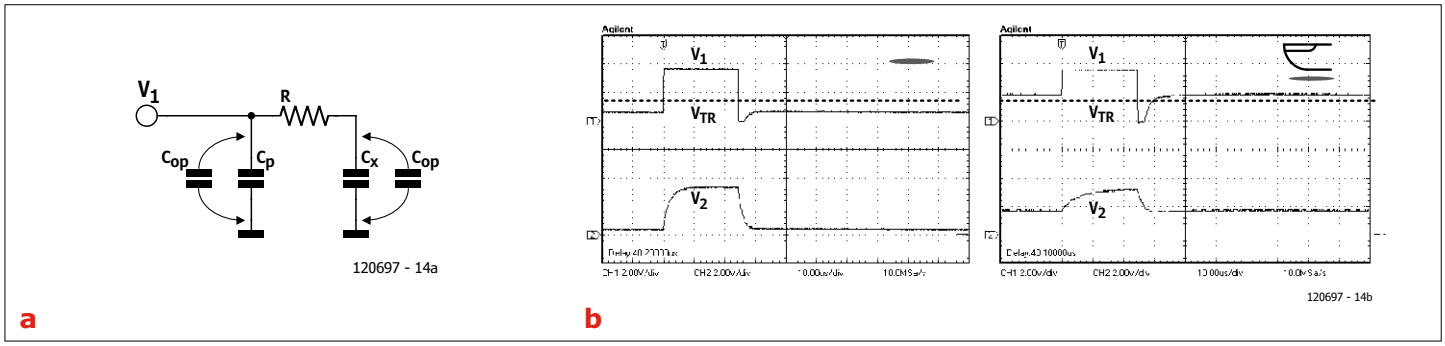


Figure 4. Real waveforms for untouched pad (upper trace) and touched pad (lower trace), with D = 10 mm and d_{x1} = 0.1 mm (paper insulator).

problematic during the SHARE and MEASURE phases, whose total duration is limited to 'just' 1 µs. For example, AC noise with an induced amplitude of 100 V has an increment of $V_{AC\text{-}max}$=22 mV during the sensing cycle. So, AC powerline voltages may be considered as DC signal which does not affect the sensing mechanism. The influence of radiated noise (high frequency) can be reduced by implementing software filters in VHDL or C, while environmental noise usually has a slow increment over time. ⏮

(120697)

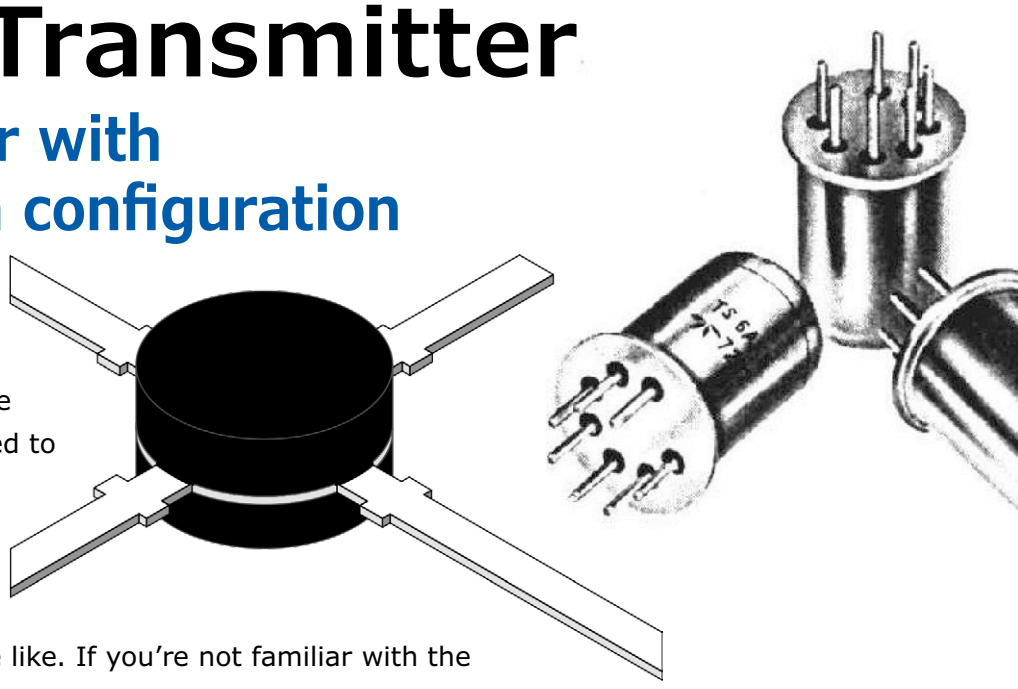
```
    BEGIN
        IF rising_edge(CLK) THEN  -- Creates the flipflops
            if counter < max_time then
                counter := counter + 1;
                if counter < zero_time then
                    dir <= '1';
                else
                    dir <= '0';
                end if;
            else
                dir <= '0';
                counter := 0;
            end if;
        END IF;

        SS <= b;

        IF( dir = '0') THEN
            if counter = zero_time then
                S <= 'Z';
            else
                if counter = read_time then
                    b <= S;
                end if;
            end if;
        ELSE
            if counter < one_time then
                S <= '1';
            elsif counter >= one_time and counter < zero_time then
                S <= '0';
            end if;
        END IF;
    END PROCESS;
END arh;
```

Picowatt Transmitter

An FM transmitter with a BF961 in Fetron configuration



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

The BF961 is a plain vanilla dual-gate MOSFET, which can also be configured to act as a Fetron. And with this Fetron you can build a simple FM radio transmitter with low power consumption and low output power — ideal for testing FM radios and the like. If you're not familiar with the term Fetron, read on...

A small FM radio transmitter is basic equipment on every workbench or in every electronics lab. This sort of transmitter is very useful for testing a variety of receivers, but it should have very low output power (practically undetectable at arm's length) to avoid interference with broadcasters or official radio services and conflicts with regulations and statutory rules. This can be achieved very easily with the circuit described here.

Fetron

A long time ago in 1972 — from the present perspective, shortly after the invention of the transistor — Teledyne Semiconductor unveiled a special semiconductor device intended to be a direct replacement for pentode tubes. After all, vacuum tubes with filaments were threatened with extinction in the early 1970s, and the Fetron was supposed to be a replacement for pentodes. That would give various types of treasured equipment and tried-and-true circuits a longer lease on life.

Figure 1 shows this revolutionary invention in its full glory: a pair of n-channel high-voltage FETs connected in cascade to form a three-terminal semiconductor device with characteristic curves similar to those of a pentode. With a maximum operating voltage of 350 V, high maximum frequency and other similar properties, it was indeed suitable as a direct replacement for tubes – and with no need for filament heating. An original datasheet (in German) from the distributor Omni Ray can be found at [1]. They offered this phenomenal semiconductor component at an equally impressive price of 46 Swiss francs (each).

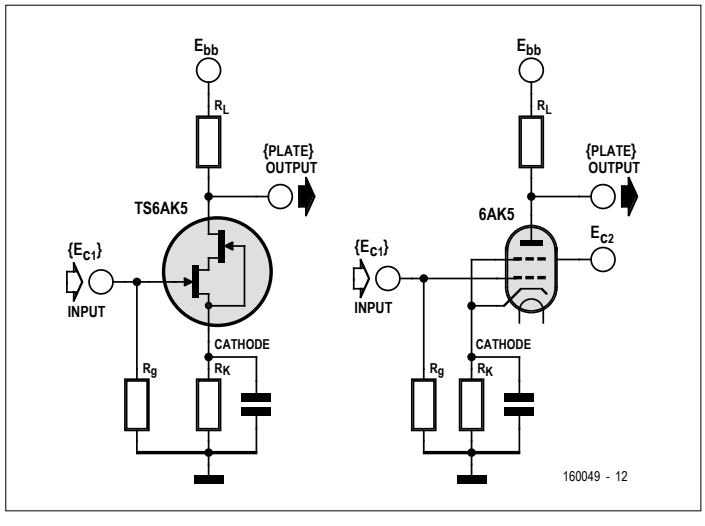


Figure 1. A 6AK5 (= EF95, *Ed.*) pentode tube (right) and a Teledyne Fetron replacement device (left) with the alliterative type number TS6AK5.

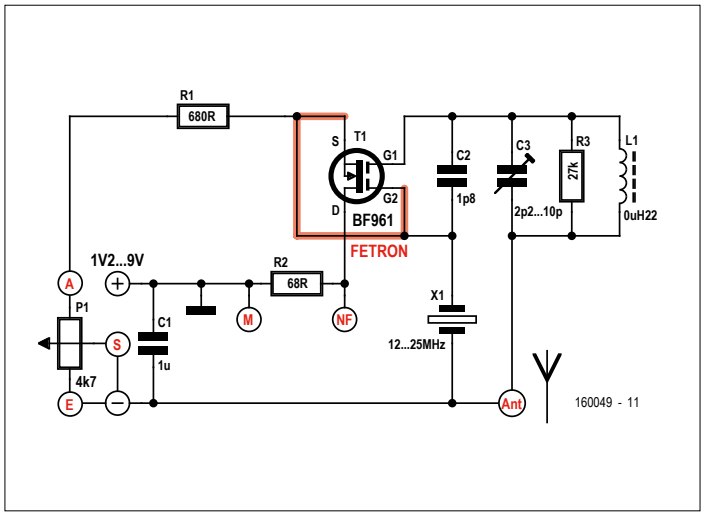


Figure 2. Circuit diagram of the picowatt transmitter with the Fetron substitute in common-source mode.

FM transmitter

The author thought to himself: with high frequency capability and pentode characteristics, you could build some nice RF circuits. However, it would take extraordinary luck to get your hands on a working Fetron now. The only solution is to find a replacement for the tube replacement. And that turns out to be possible: you can certainly configure a modern dual-gate MOSFET to obtain characteristic curves similar to those of a pentode. If you don't need high voltage capability that gives you what you want with a BF961 MOSFET. As can be seen from its datasheet [2], this device is also fast. Now the key question is: how you can you make a Fetron-like device from a dual-gate MOSFET? The answer is simple: connect gate 2 to the source.



Figure 2 shows a very low-power FM radio transmitter built with the "Fetron substitute" and a few additional components, which needs amazingly little operating current thanks to its inherently high internal impedance. Here the pseudo-Fetron is wired in common-source mode. The modulating signal from a low-impedance source is injected via the NF input terminal (referenced to the ground terminal M) to vary the supply voltage. The neat thing about this is that the oscillator continues operating at its resonant frequency with constant amplitude. The tracking range of the resonant frequency exactly corresponds to the audio modulation signal, but the amplitude modulation components to the left and right of the tracking range are effectively cut off. The result is exactly what you want: frequency modulation.

How it works: The oscillator feedback path is from the junction of the source and gate 2 to the quartz crystal in series with capacitor C2, which are wired in parallel with the resonant circuit. Trimmer capacitor C3 allows the oscillator to be tuned to resonate at an overtone of the crystal frequency. The modulating audio signal must be fed in from a low-impedance source. A small audio amplifier with low output power or the headphone or speaker output of a radio or other device is suitable for this purpose. Capacitor C1 suppresses induced hum. The low current consumption and wide operating voltage range of 1.2 to 9 V are amazing. The oscillator works with a current as low as 0.3 mA, and a standard setting of 0.6 mA is enough for reliable operation over the full supply voltage range. All this means that the test transmitter can easily be powered from a battery.

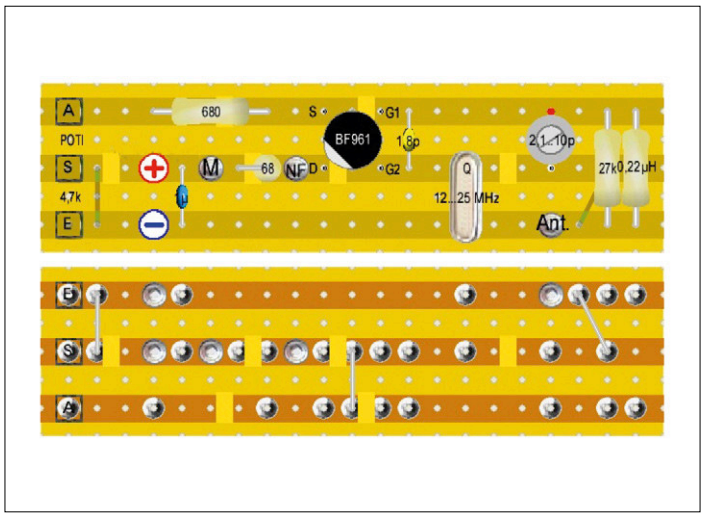


Figure 3. The transmitter can be built on a small piece of stripboard with just three strips.

Build and test

To make assembly really easy, the author designed a PCB layout, and the artwork files can be downloaded for free from the Elektor web page for this project [3]. The files are for Sprint Layout, which is not a widely used program, but that is not a major problem because the circuit is so simple that it can easily be built on a piece of stripboard despite its RF operation. A suitable layout for this is shown in **Figure 3**. Mounting the handful of components is quick and easy. Instead of connecting a full-blown potentiometer to terminals A, S and E, you can extend the board a bit and solder on a trimpot. For this you should use a trimpot with a relatively large footprint. The only thing you need to careful with is the MOSFET, which like all components of its kind is sensitive to electrostatic discharge. That means you should wear a ground strap while handling the device.

After assembly and checking, you can connect a suitable power source and an audio signal. Adjust P1 to the recommended supply current of 0.6 mA, and then tune an FM radio next to the transmitter through its receiving range. If you don't pick up a signal from the transmitter, turn trimmer C3 a bit and try again. Once you have found the transmitter signal, you can adjust the amplitude of the audio signal to a level that avoids audible distortion in the receiver. That's it.
By the way, raising the supply voltage increases the RF power. Connecting an antenna is not necessary if the transmitter is right next to the receiver — it can radiate a few picowatts even without an antenna. ⏮

(160049)

Web Links

[1] Fetron datasheet: http://goo.gl/RrIOW8
[2] BF961 datasheet: http://goo.gl/xNha5N
[3] www.elektormagazine.com/160049

Component List

Resistors
Default: carbon film, 5%
R1 = 680 Ω
R2 = 68Ω
R3 = 27kΩ
P1 = 4.7kΩ, pot or trimpot*

Capacitors
C1 = 1µF, ceramic or film*
C2 = 1.8pF ceramic
C3 = 10pF trimmer*

Inductor
L1 = 0.22µH

Semiconductor
T1 = BF961, e.g. Vishay

Miscellaneous
X1 = quartz crystal, 12-25MHz, overtone, series resonance
Antenna*
PCB or stripboard, 57.5 x 17.5mm*

* see text

The combination of RTLS and IPS technologies

Discover the Hungarian developed cutting-edge system

spad

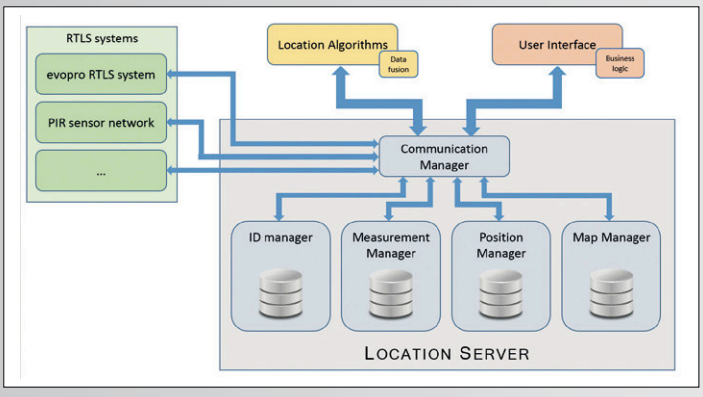
Market Trends

With the spread of the location-based services of smartphones and GPS, the importance of outdoor positioning became obvious. Civil society's growing demand for safety (care of children and the elderly, access control to dangerous areas etc.) and the pursuit of higher efficiency in industrial systems fuels the need for indoor real-time locating.

In the industrial scene, the supply chain's efficiency can be highly increased by tracking materials, tools and products, and the tracking of the staff inside the building or factory aids work organizing. The need is real for real-time tracking of people and objects in civil society and the industrial scene.

Hybrid Technologies

Hybrid technologies represent the new direction in the development of RTLS (Real-time Location Systems) and IPS (Indoor Positioning Systems). These systems use at least two types of technologies in order to get most accurate results possible. Many market players use this principle, and the most common combination, because of its low costs and the capability of using an already existing WiFi infrastructure, is the mix of WiFi and BLE (Bluetooth Low Energy).



The architecture of RTLS system developed by Spad Ltd.

The Solution Developed by SPAD Ltd.

SPAD Ltd. developed a cutting-edge system which combines RTLS and IPS technologies for an R&D tender.

Characteristics and Benefits

The solution is capable of localizing people and assets in real-time in a changing environment. The system is made up by active and passive tags, by the wireless network created by the tags and the communication hubs, by a cloud-based service center and by the software layer making location-based services possible.

The novelty of the system consists of the energy-optimized active tags, the integrated evaluation of the passive tags' signals (sensor fusion), the security solutions of the network and the generic service-based software relying on the raw data of tagged objects. The solution combines the benefits of IPS and RTLS for industrial applications by using RFID, BLE, WiFi, passive motion-sensing and the sensors of smart devices (i.e. accelerometer, gyroscope). Furthermore, the solution has multiple characteristics that make it an ideal choice for a variety of use cases:

- Its modular structure makes it a flexible and scalable system that can be adjusted to specific needs cost effectively and expanded easily;

- By integrating numerous technologies, fusing sensors and using different location algorithms, it can be used in various industrial and civil environments.

Civil society's growing demand for security systems' accountability and transparency means that a real-time positioning solution needs to be created in a way that no entity can use it for its own, self-serving purpose. That's why the solution detailed here is devised in a way that, while maintaining the application-specific usability, it can randomize data, so that no specific ID-, time-, location- or route data can be obtained and combined. The system is also capable of using a mix zone model, which doesn't pair route sub-sections, and a k-anonymity model, which combines multiple outcomes to a certain input.

The system is already proving itself in multiple locations: the sites of Ikarus Egyedi Ltd., SPAD Ltd. and evopro Innovation Ltd.

Project leader: *István Szatmári PhD*

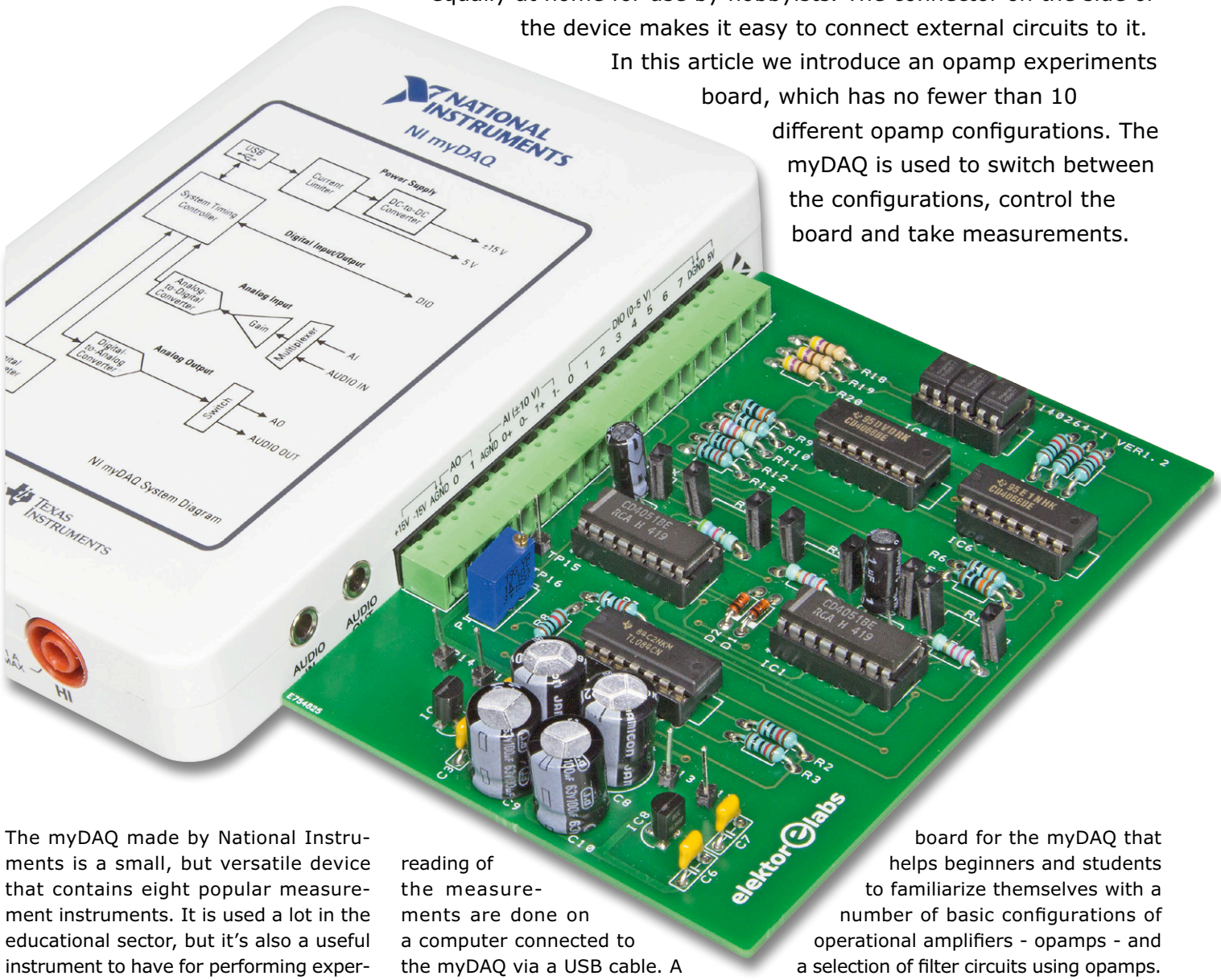
For more information: *spad@spad.hu*

Opamp Experimenting Kit for myDAQ

Tinker with 10 different configurations

By **Sunil Malekar** (Elektor Labs, India) and **Roy Aarts** (Elektor Labs trainee)

The myDAQ measurement instrument made by NI is a versatile device for educational purposes, which is equally at home for use by hobbyists. The connector on the side of the device makes it easy to connect external circuits to it. In this article we introduce an opamp experiments board, which has no fewer than 10 different opamp configurations. The myDAQ is used to switch between the configurations, control the board and take measurements.



The myDAQ made by National Instruments is a small, but versatile device that contains eight popular measurement instruments. It is used a lot in the educational sector, but it's also a useful instrument to have for performing experiments and measurements in your lab at home. Some of the features of the myDAQ include a dual-channel oscilloscope, a DMM, a two-channel function generator and a number of digital inputs and outputs. The control of the device and the reading of the measurements are done on a computer connected to the myDAQ via a USB cable. A large connector on the side makes it easy to connect your own circuits to the board. In Elektor we have previously had a look at myDAQ expansions (July & August 2014; September 2014).

This project concerns an expansion board for the myDAQ that helps beginners and students to familiarize themselves with a number of basic configurations of operational amplifiers - opamps - and a selection of filter circuits using opamps. The board therefore contains a number of opamps, along with a large number of electronic switches. Via the myDAQ you can switch between the different configurations and subsequently take real-time measurements of the signals for the

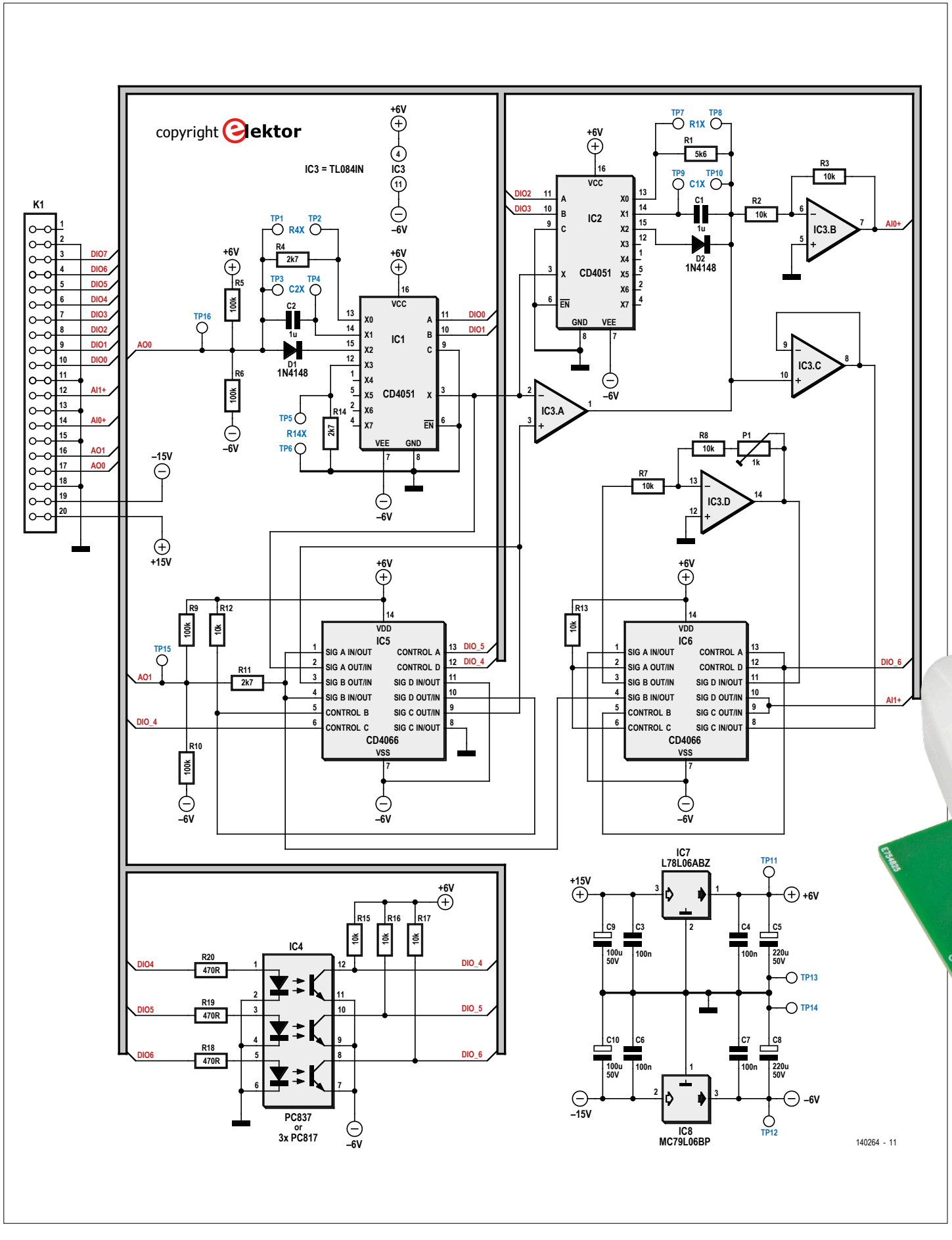


Figure 1. The circuit appears to be very complex, but it consists of just a few opamps that can be configured in a variety of ways using electronic switches.

chosen opamp configuration. A particular configuration can be selected on the screen. A number of virtual dials on the screen can then be used to adjust the frequency, waveform and amplitude. There are four scope screens, which show the inputs and outputs of both channels. The software for this project is — naturally — made with LabVIEW. This combination gives you the opportunity to learn how the output signals have been programmed and how the input signals of the myDAQ have been visualized on the screen with the help of LabVIEW.

The myDAQ hardware

The NI myDAQ is provided with two analog inputs (AI) with a maximum sample rate of 200 ksamples/s and a resolution of 16 bits, 2 analog outputs (AO) with the same sample rate en resolution, eight digital inputs/outputs (DIO), one 32-bit general purpose counter/timer, one 3.5-digit digital multimeter (DMM) and a power supply section that supplies +5 V at 100 mA and ±15 V at 32 mA. On the side is a 20-way connector where an expansion board can be plugged in.

The communications with the computer happen via a USB 2.0 Hi-Speed interface. A number of software instruments for standard functions is provided with the myDAQ (ELVISmx Software Instruments), which can be run independently. When you want to use it for different applications, you will also need a copy of LabVIEW.

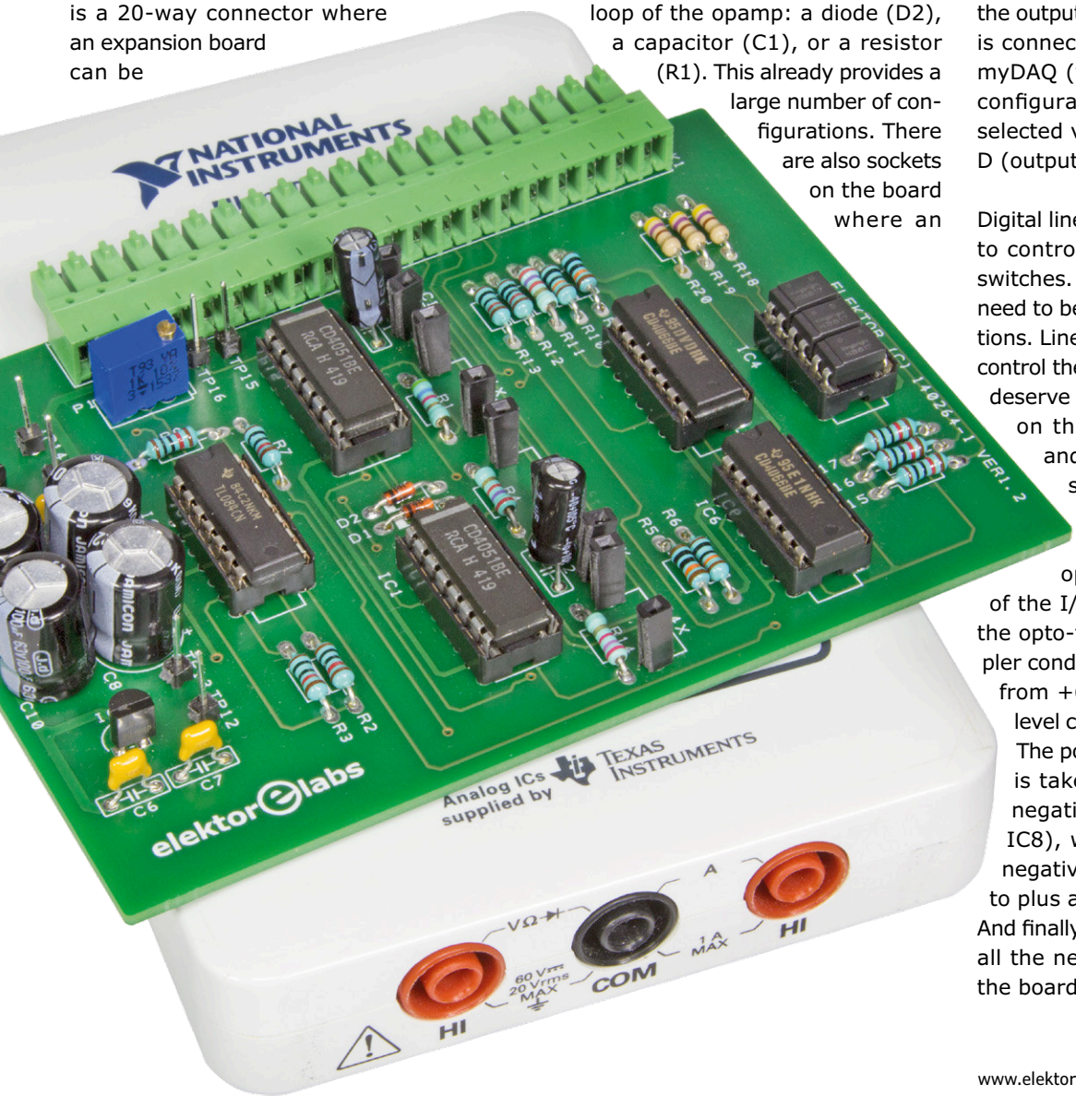
The circuit

The circuit shown in **Figure 1** may appear very complex at first, but it's not that bad really. At the heart of the circuit is opamp IC3.A, which can be switched into a number of different configurations using two 8-channel analog multiplexers/demultiplexers, IC1 and IC2. IC1 is used to select the component at the input of the opamp. This can be a diode (D1), a capacitor (C2), or a resistor (R4) in series with the inverting input of the opamp, or a resistor to ground (R14) in the non-inverting configuration. IC2 is used to select the component in the feedback loop of the opamp: a diode (D2), a capacitor (C1), or a resistor (R1). This already provides a large number of configurations. There are also sockets on the board where an extra resistor or capacitor can be plugged in (R1X, C1X, R4X, C2X, R14X), which will be in parallel with the component already on the board. At the output of IC3.A is an inverting (IC3.B) and a non-inverting stage (IC3.C). The output of the de non-inverting stage always goes to the first analog input of the myDAQ. Depending on the settings on the control panel, either the inverted output signal or the output of the configurable amplifier (IC3.D) goes to the second input of the myDAQ.

There is also a pair of 4066 ICs (IC5 and IC6) on board, which each contain four analog switches. With IC5 the second output signal from the myDAQ can be connected via R11 to either the inverting or the non-inverting input of IC3.A, resulting in a differential or summing circuit. It is also possible to send the signal after R1 to the configurable amplifier built around IC3.D. Switch C in IC5 is used to connect the non-inverting input of IC3.A to ground for some of the configurations. The switches in IC6 determine whether the output signal of IC3.B or that of IC3.C is connected to the second input of the myDAQ (via switch C). And finally, the configurable amplifier around IC3.D is selected via switch B (input) and switch D (output) in IC6.

Digital lines DIO0 through DIO6 are used to control the multiplexers and analog switches. **Table 1** shows which I/O lines need to be high for each of the configurations. Lines DIO4, DIO5 and DIO6, which control the analog switches in IC5 en IC6, deserve a closer look. Since the voltage on the I/O lines is 3.3 V, and IC5 and IC6 have a symmetrical ±6 V supply, we need to have a level conversion between them. For this we've used three opto-couplers (IC4). When one of the I/O signals at the input is high, the opto-transistor inside the opto-coupler conducts and pulls the output signal from +6 V to −6 V. In other words, a level converter with a built-in inverter! The power supply for the whole circuit is taken care of by a positive and a negative voltage regulator (IC7 and IC8), which reduce the positive and negative 15 V supplied by the myDAQ to plus and minus 6 V.

And finally, 20-way connector K1 provides all the necessary connections between the board and the myDAQ.



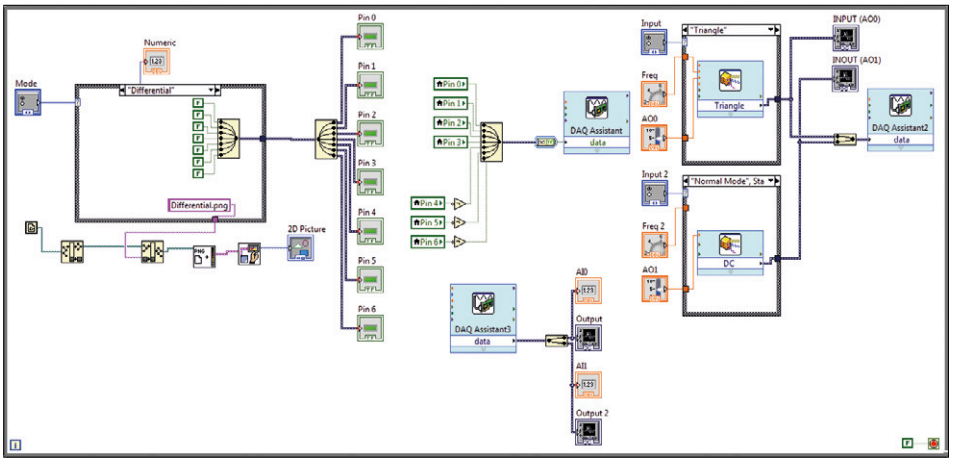


Figure 3. A block diagram of the LabVIEW program developed for the opamp experiments board.

Construction

Naturally, a printed circuit board has been designed for this circuit, as shown in **Figure 2**. It has a spacious layout and contains only through-hole components, which make the construction very easy. If necessary, you can put the ICs in socket. This can be useful with the opamps in particular, in case you want to experiment with other (pin-compatible) types. Only low-frequency signals are used on the board, so the possibility of external interference is therefore negligible. For pins TP1–TP16 you could cut 10-way male and female headers into single connectors. It is also possible to buy single pins, but these aren't widely available.

Once you have populated the board and mounted the 20-way connector, you can plug the expansion board into the socket on the side of the myDAQ. It's recommended to provide the board with some extra supports on the underside, so that it's not supported by just the connector. This makes it safer for you to put some pressure on the board when you insert extra components or when you adjust the potentiometer.

For those of you who want to build the circuit, we're providing a kit of parts that contains the 20-way connector (K1) and the MC79L06BP-AP voltage regulator, which aren't always easy to find, and the printed circuit board (Elektor Store order number: 140264-71).

The software

A software module in LabVIEW has been developed for use with this circuit, which lets you select a number of opamp-configurations, 'play' with the input signals and see what happens to the output signal. This vi-module can be downloaded freely from the Elektor Magazine website [1], but you first need to install LabVIEW on your PC before you can use it.

The block diagram for the program is shown in **Figure 3**. On the left the desired opamp configuration is selected, which results in setting the required digital outputs of the myDAQ. On the right hand side, the signals are generated for the analog output of the myDAQ. In the center at the bottom, the signals on the analog inputs are shown on four graphical windows. The block diagram can therefore be separated into a digital logic section, a function generator, and an oscilloscope.

When you start the program (**Figure 4**), you can experiment with all the different opamp circuits. On the left of the screen are 10 check boxes. These are used to select one of the opamp configurations on the board. Below that is the circuit diagram of the selected configuration. To the right of the check boxes are seven green boxes that represent the logical levels of

Component List

Resistors
Default: 5%, 250mW
R1 = 5.6kΩ
R2,R3,R7,R8,R12,R13,R15,R16,R17 = 10kΩ
R4,R11,R14 = 2.7kΩ
R5,R6,R9,R10 = 100kΩ
R18,R19,R20 = 470Ω
P1 = 1kΩ preset

Capacitors
C1,C2 = 1µF 50V, radial
C3,C4,C6,C7 = 100nF
C5,C8 = 220µF 50V, radial, 10mm pitch
C9,C10 = 100µF 50V, radial, 8mm pitch

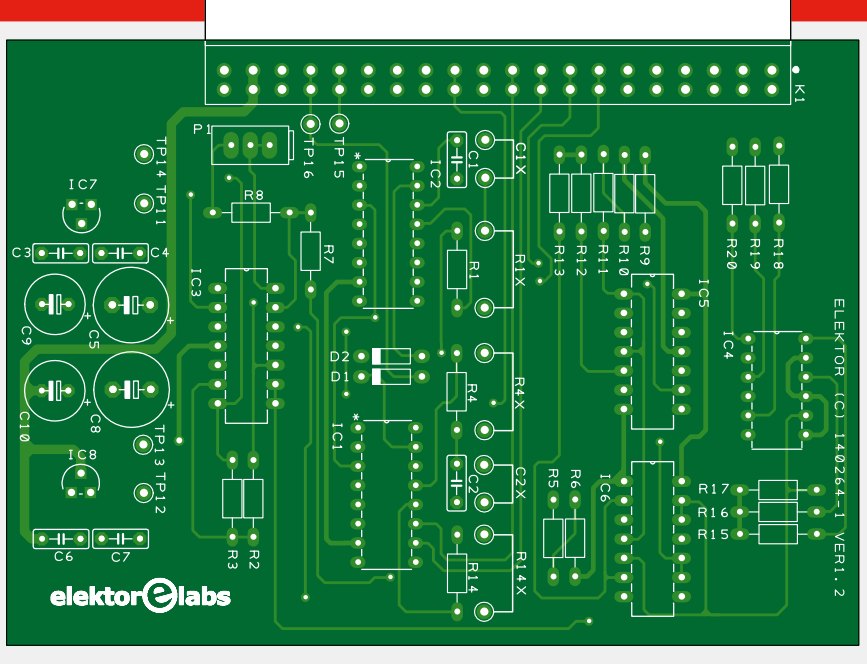


Figure 2. The board has a clear layout and is easy to populate because only through-hole components have been used.

the digital outputs of the myDAQ. These will be the same as the levels shown in Table 1. These signals are then used to control the analog multiplexers on the board. Next to these are a number of settings that can be applied to the two output signals of the myDAQ. For the waveform you can choose between a DC voltage, a sine-wave, a triangle-wave and a square-wave. The vertical sliders are for setting the level and polarity of the DC-signal, or the amplitude of the other signals. The 'Freq' dials are used to set the frequency. This has purposely been kept very low, so the signals are easy to see on the scope screen. On the right are the four scope screens, with the left two showing the input signals, and the right two showing the output signals. You are now ready to experiment to your heart's content and observe what the selected circuit does with the input signals. Happy experimenting!

(140264)

Web Link

[1] www.elektormagazine.com/140264

Table 1. I/O line functions

Configuration	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
Inverting	X	0	0	1	0	0	0	0
Non Inverting	X	0	0	0	0	0	1	1
Summing	X	0	1	1	0	0	0	0
Differential	X	0	0	0	0	0	0	0
Differentiator	X	0	0	1	0	0	0	1
Integrator	X	0	0	1	0	1	0	0
Logarithmic	X	0	0	1	0	1	0	0
Anti Logarithmic	X	0	0	1	0	0	1	0
Comparator	X	0	0	1	1	1	0	0
Gain	X	1	0	1	0	0	0	0

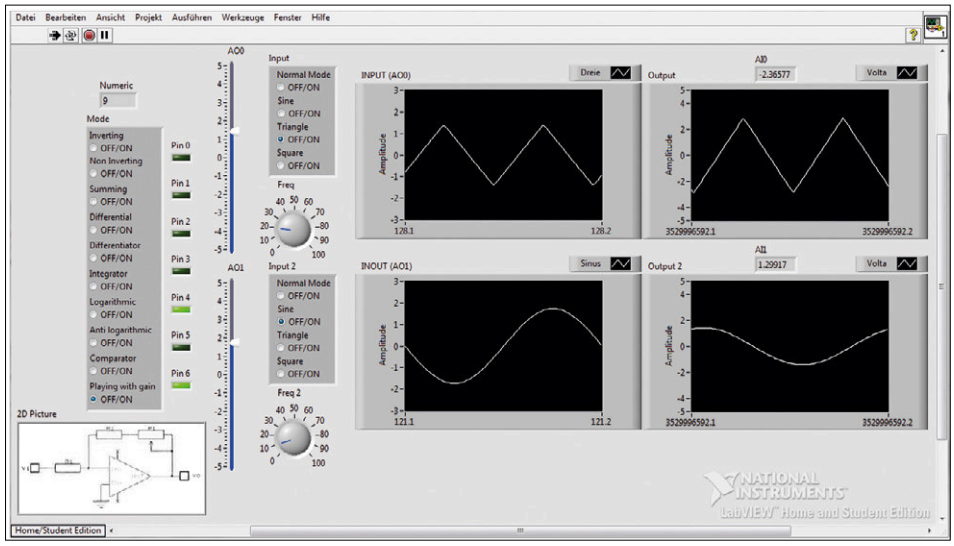
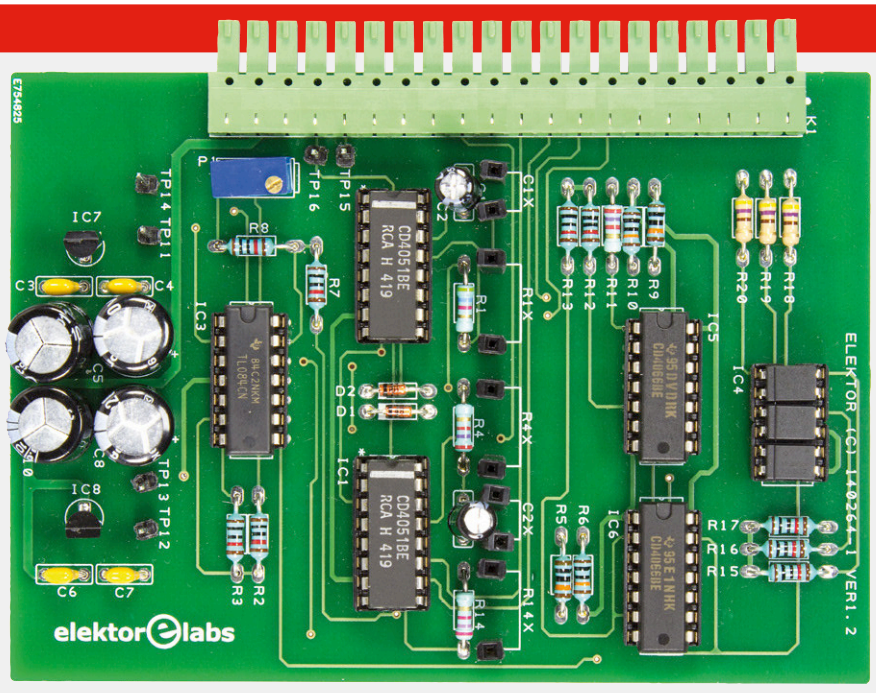


Figure 4. This is how the program appears on your screen.
On the left you can select the configurations and input signals, on the right are the scope screens showing all the input and output signals.



Semiconductors

IC1,IC2 = CD4051BE
IC3 = TL084
IC4a,IC4b,IC4c = PC837 or 3 pcs PC817
IC5,IC6 = CD4066BE
IC7 = L78L06
IC8 = MC79L06
D1,D2 = 1N4148

Miscellaneous

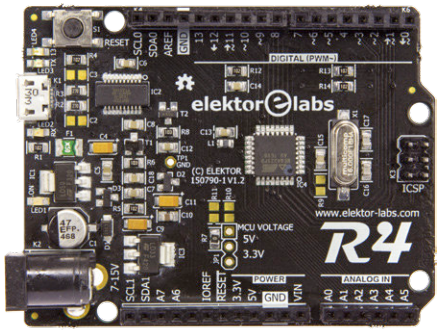
DIP-6 IC socket
DIP-14 IC socket
DIP-16 IC socket
K1 = 20-way connector, 3.81mm pitch, for linking to myDAQ
TP1-TP10 = single-pin receptacle, 0.1'' pitch
TP11–TP16 = single-pin pinheader, 0.1'' pitch
PCB package # 140264-71, contains PCB, 20-pin myDAQ connector and MC79L06 voltage regulator

welcome in your ONLINE STORE

EDITOR'S CHOICE



The Elektor Uno R4 is an evolution of the Arduino Uno R3. The R4 has an ATmega328PB processor, which is the supercharged 'B'-rother of the ATmega328P. Compared to the Uno R3, the Elektor Uno R4 features two USARTs, four SPI peripherals, two I²C peripherals, five timers and nine PWM channels. It also has an extra 4-bit GPIO port E, 8 analog inputs, an analog comparator output, a unique ID and some more extra stuff. The R4 is an improved drop-in replacement for the R3 that can also run on 3.3V (no more level shifters!). Integration in the Arduino IDE is easy thanks to a Boards Package that can be downloaded from GitHub (https://github.com/ElektorLabs/Arduino/). The R4 is compatible with Windows, Linux and OS X. Now that I have the R4, I don't use the R3 anymore.

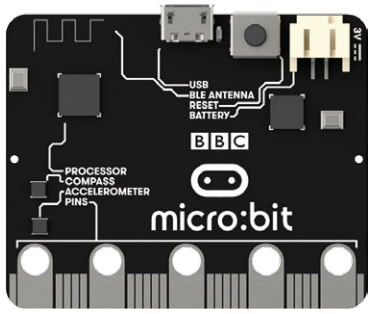


Clemens Valens
Elektor Labs

www.elektor.com/elektor-uno-r4

Elektor Bestsellers

1. BBC micro:bit
   www.elektor.com/microbit



2. Elektor Uno R4
   www.elektor.com/elektor-uno-r4
3. CAN Projects with ARM and Arduino
   www.elektor.com/can-projects
4. Make: Arduino Special
   www.elektor.com/make-arduino-special
5. Elektor SDR Reloaded
   www.elektor.com/sdr-reloaded
6. ARM Microcontroller Projects
   www.elektor.com/arm-projects
7. DVD Elektor Magazine 2010 - 2014
   www.elektor.com/elektor-2010-2014

Arduino Uno - 45 Projects for Beginners and Experts

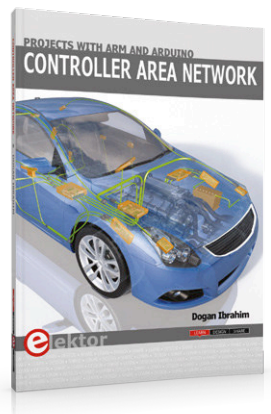


This book covers a series of exciting and fun projects for the Arduino, such as a silent alarm, people sensor, light sensor, motor control, internet and wireless control (using a radio link). Contrary to many free projects on the internet all projects in this book have been extensively tested and are guaranteed to work!

member price: £25.95 • €31.46 • US $35.00

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: £27.95 • €33.75 • US $38.00

www.elektor.com/can-projects

DVD Elektor Magazine 2010 - 2014

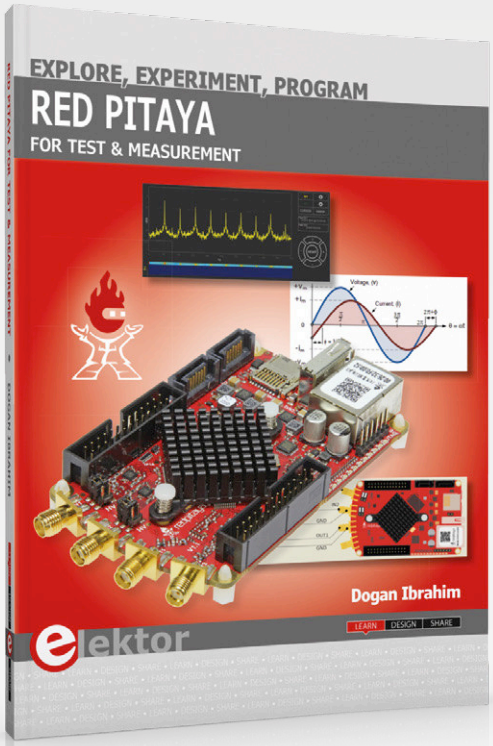


This DVD-ROM contains all circuits and projects published in Elektor Magazine's year volumes 2010 through 2014. Since 1975, specialist magazine 'Elektor' has been in the forefront of electronics and computer technology through the publication of repeatable, professionally designed circuits and projects. 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

Red Pitaya for Test en 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. The book includes many chapters on basic electronics and teaches the theory and use of electronic components including resistors, capacitors, inductors, diodes, transistors, and operational amplifiers in electronic circuits. Many fun and interesting Red Pitaya experiments are included in the book. The book makes an introduction to visual programming environment.

MEMBER PRICE: £24.95 • €29.95 • US $34.00

www.elektor.com/red-pitaya-book

Finally: A Red Pitaya Book

Limited time offer for GREEN and GOLD Members: 15% Discount plus free shipping!

Elektor SDR Reloaded

Software Defined Radio shield for the Arduino!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

Raspberry Pi 3 Starter Kit (Deluxe)



This special Raspberry Pi 3 Starter Kit includes everything to get started right away with the world's most popular mini computer! The kit consists of a Raspberry Pi 3 (Model B), a high-quality ABS case for Raspberry Pi, an original micro USB power supply for Raspberry Pi 3 (5.1 V, 2.5 A), a high-speed HDMI Cable (1 m), a patch cable cat.5e (2 m) and a transcend (16 GB, Class 10) microSD Card with SD Adapter (pre-installed with NOOBS).

member price: £66.95 • €80.96 • US $91.00

www.elektor.com/rpi- starter-kit-deluxe

IoT-GET-U-GOING



In 35 fun projects, this book will show you how to build your own Internet of Things system. We'll cover the hardware (primarily the Raspberry Pi and Arduino) and the software that makes control via Internet possible. We employ Wi-Fi and radio links so no requirement any longer to install cabling crisscross through your home. In this unique book, Raspberry Pi, Arduino and HTML webpages with stylesheets and JavaScript come together in clearly-described, easy-to-build projects.

member price: £29.95 • €35.96 • US $40.00

www.elektor.com//iot-get-u-going

Logic Analyzer ScanaQuad



ScanaQuad (SQ) is a series of high performance 4 channels logic analyzers and digital pattern generators. They are designed to be your best companion when working on serial protocols like UART, SPI, I²C, 1-Wire, USB, I²S, CAN, LIN, RS232, RS485, and more. With ScanaQuad Logic Analyzers, you can capture signals, you can play them back, and you can even build genuine test signals and generate them!

member price: £106.95 • €128.66 • US $144.00

www.elektor.com/scanaquad

Join the Elektor Community

Take out a GOLD Membership now!





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

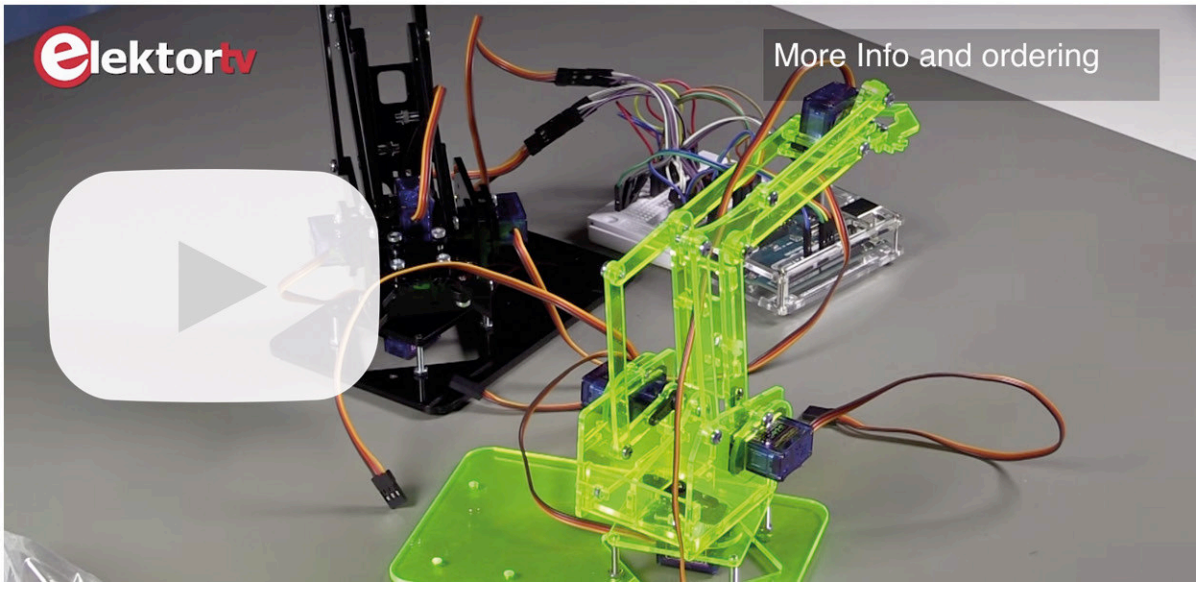
layar INTERACTIVE PRINT

SCAN THIS PAGE AND WATCH THE VIDEO

1 Download the free Layar App

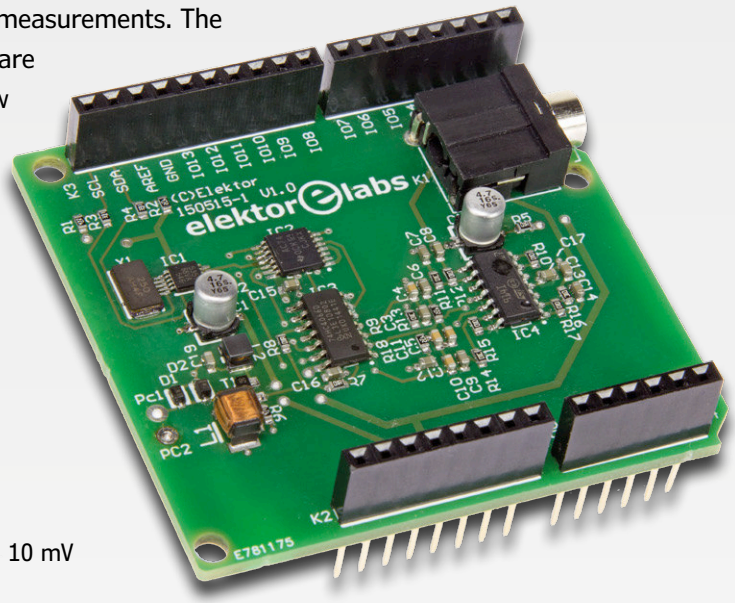
2 Scan this page

3 Discover interactive content



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.



Technical Characteristics:

- Supply voltage: 5 V and 3.3 V as for Arduino
- Frequency range: 150 kHz up to 30 MHz
- Sensitivity: 1 µV
- Total amplification: 40 dB
- Maximum antenna signal level: 10 mV
- Dynamic range: 80 dB

MEMBER PRICE: £24.95 • €26.96 • US $30.00

www.elektor.com/sdr-reloaded

Finally: A Red Pitaya Book

Limited time offer for GREEN and GOLD Members: 15% Discount plus free shipping!

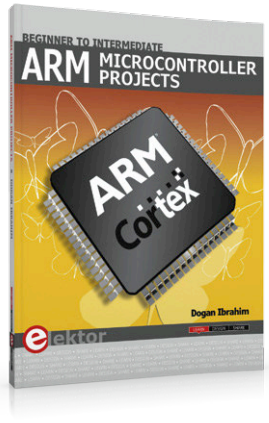
Elektor SDR Reloaded

Software Defined Radio shield for the Arduino!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

ARM Microcontroller Projects

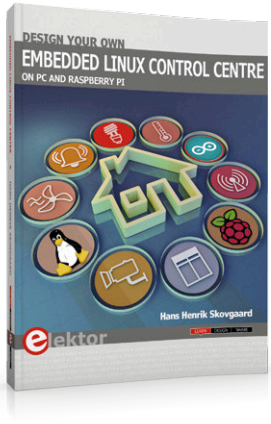


This book makes use of the ARM Cortex-M family of processors in easy-to-follow, practical projects. It gives a detailed introduction to the architecture of the Cortex-M family. The architecture of the highly popular ARM Cortex-M processor STM32F107VCT6 is described at a high level, taking into consideration its clock mechanisms, general input/output ports, interrupt sources, ADC and DAC converters, timer facilities, and more.

member price: £29.95 • €35.96 • US $40.00

www.elektor.com/arm-projects

Design Your Own Embedded Linux Control Centre 3

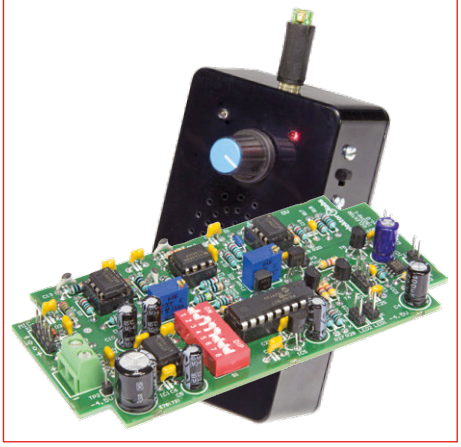


This book is all about building your own DIY home control system. It presents two innovative ways to assemble such a system: By recycling old PC hardware - possibly extending the life of an old PC, or by using Raspberry Pi. In both cases, the main system outlined in this book will consist of a computer platform, a wireless mains outlet, a controller and a USB webcam - All linked together by Linux.

member price: £31.95 • €38.66 • US $44.00

www.elektor.com/elcc3

Bat Detector PLUS



Unlike many other projects this bat receiver employs both frequency division and amplitude recovery! This bat detector uses a MEMS microphone for its high sensitivity in the ultra-sonic range. The detector produces a sine wave for the speaker which can be re-modulated with the original amplitude. Four low-pass filter cutoff frequencies, four frequency division ratios and more can be set with DIP switches.

member price: £.51.95 • €62.96 • US $71.00

www.elektor.com/bat-detector-plus

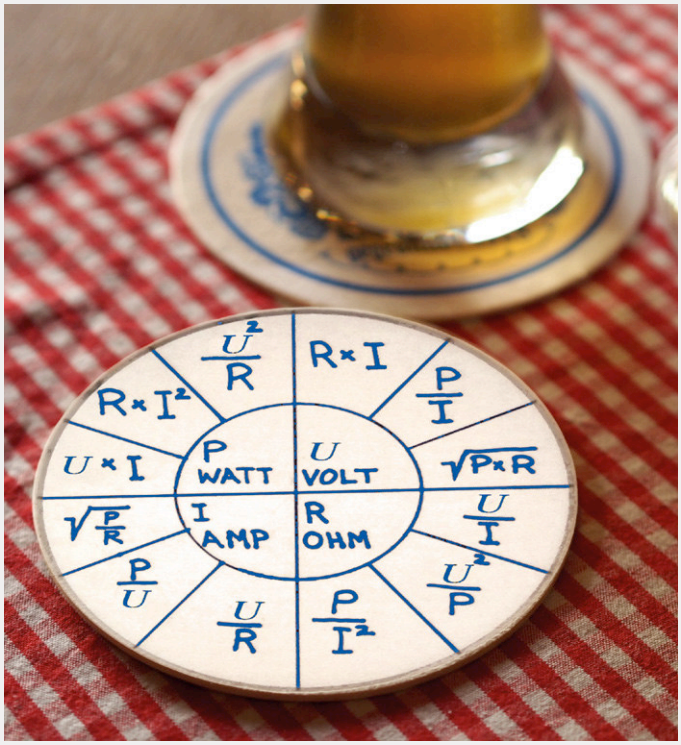
Welcome to the SHARE section



By **Thijs Beckers** (Elektor Netherlands)

Coaster

For a while now, our Elektor e-zine (weekly newsletter) has opened with an small editorial piece. In *E-zine 169 | Ohm's Law revisited* colleague Clemens Valens wrote a piece about the history of Ohm's Law (I therefore am not referring to Welcome to the Design Section earlier in this magazine), which, if things had been only slightly different, could have been called *Cavendish' Law* instead.
Prompted by this piece, Mr Van der Giessen, a faithful reader of Elektor, shared with me a diagram that I hadn't seen for a while: Ohm's Law on a coaster!
For many years this handy diagram has aided students and pupils who have found it difficult to digest this fundamental law of electronics by applying it to various calculations. I felt I had to share it with you.



Nixie but not Nixie

Another cute project was shared with me by translator Kees de Groot. He got the idea of making a Nixie clock, but one without Nixie tubes! These he wanted to replace with 'digit plates' laser-cut from a sheet of acrylic, which are lit from the side using LEDs. The rough sides of the acrylic will then radiate the light and hopefully display the appropriate digit.
I find this a great idea myself. However, the only difficulty that Kees has been struggling with, is to make the device bright enough. He has already thought about a number of variations: a round hole for the LED, an adaptor part to better capture the light, additional SMD LEDs along the edges, edges beveled at 45°, perhaps digits that were cut using another method would work better… However, 'life' made other demands and the project is now in cold storage for the time being. But if someone would like to take the idea further, then please share it with us!

(160078)

If you haven't read the editorial by Clemens or would like to refresh your memory then you can find the article here:
www.elektormagazine.com/news/ohm-s-law-revisited

Elektor Labs Pipeline

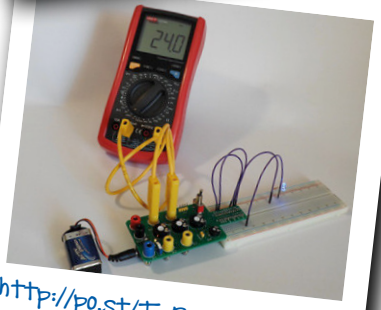


http://po.st/FastForward

Fast-Forward Your Project to Success

Posting a project on Elektor Labs allows you to get exposure for your work. The most interesting projects are selected for publication in Elektor Magazine and the ones highlighted in our e-zine are read by thousands of people. We now go a step further by organizing a contest culminating in a big event on the electronica trade show to be held in Munich at the beginning of November 2016. Send in your project and compete for the great prizes that can be won. Entries can be as simple as an idea, or as advanced as a start-up company.

Do you have a great idea that should be developed further? Do you have a project that you would like to take to the next level? Are you a start-up company wondering how to conquer the market? Head over to the Fast Forward Start-Up Award website, fill in the Application Form and hopefully you will be fast-forwarded into a successful future.



http://po.st/T-Board-Adapter

Improving breadboarding

Loved for their versatility and ease of use, hated for their bad contacts and ever lack of space, breadboards remain a great source of inspiration. Many circuits need some form of power supply, but breadboards do not have good connections for them. Few things are more irritating than a circuit that behaves erratically because its power supply is badly connected. The Elektor T-board concept was initially developed for microcontroller experiments on breadboards by keeping as much breadboard surface free as possible, but it can usefully be extended to other domains like power supply and measuring. That is what this project is about.

Build an electronic wind instrument

Based on a PIC32 this device lets you control a music synthesizer over MIDI by blowing in it, just like a flute. Unlike a flute, or any other wind instrument for that matter, this one can also be played without blowing at all. Also the fingering has been simplified because there are no physical constraints on the position of the keys. For the moment this Rudimentary Electronic Musical Instrument, or REMI as it is called, is not so much an instrument as it is a MIDI controller, but the addition of a full-blown (pun intended) synthesizer is planned. A PWM-controlled filter and amplifier plus audio output to support a single audio channel are already there. The design is completed with an air pressure sensor and touch control.



http://po.st/REMI

Make your doors (look) smarter

The creator of this project lives in a house where doors stay mostly ajar, i.e. almost closed but never completely. A closed door means that someone is behind it and that he/she closed it on purpose, usually to secure some privacy. However, it can be hard to see from a distance if a door is closed or slightly ajar and so this Poster started thinking of an elegant system capable of signaling the state of the door. The Smart Door was born. The system is simple: a magnet and a reed relay create a classic door open/closed sensor. A microcontroller reads the sensor and makes an LED blink if the door is closed. Of course a microcontroller is not really necessary for this, but it has the advantage that the circuit can remain very simple and small. This is especially important because of the project's Elegancy requirement: it must be small enough to be hidden in a standard door without damaging it.



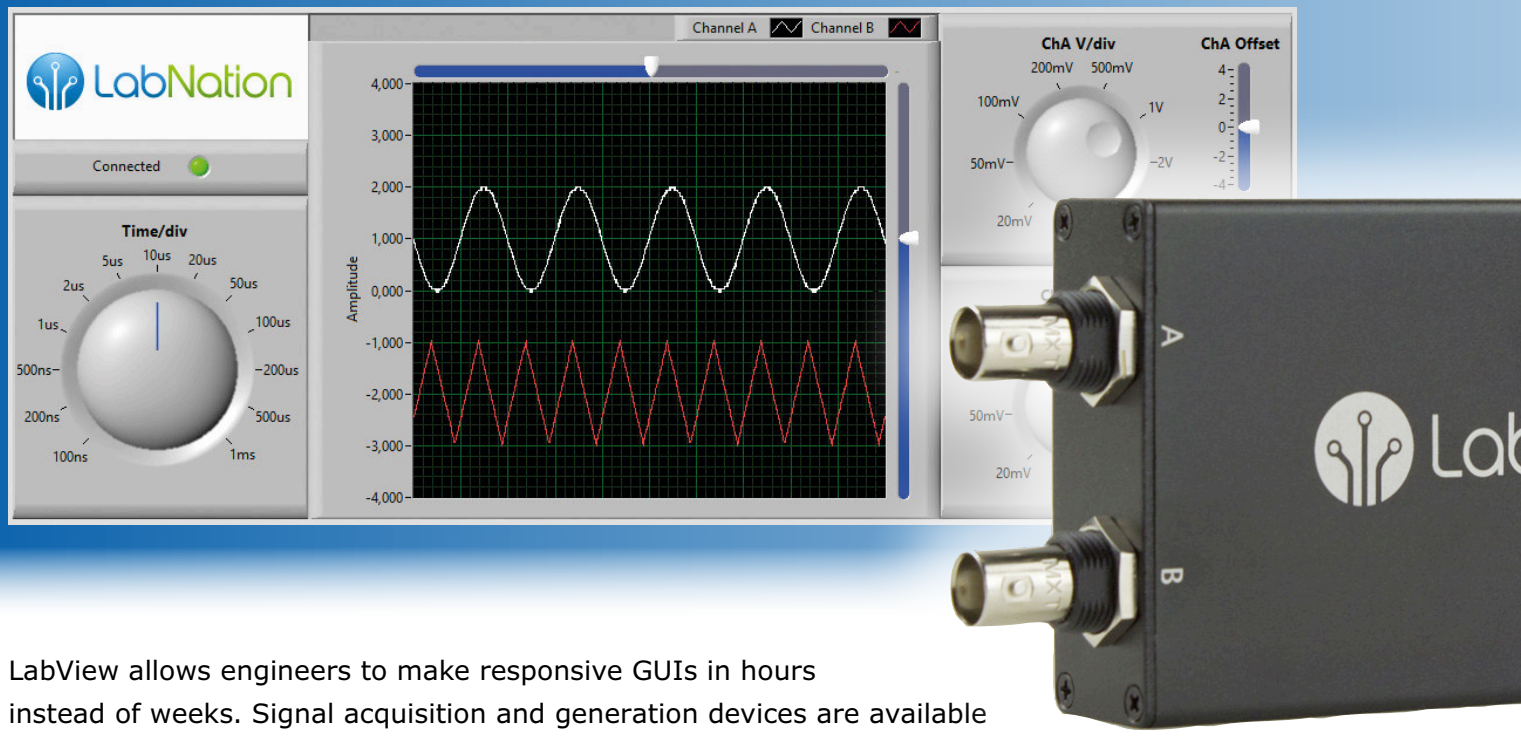
http://po.st/SmartDoor

(160079)

SmartScope Meets LabVIEW

With a dedicated set of VI blocks

By **Riemer Grootjans**, LabNation (Belgium)



LabView allows engineers to make responsive GUIs in hours instead of weeks. Signal acquisition and generation devices are available from National Instruments, but they come at a cost. The SmartScope with its dual 100 MS/s digitizers and 4 MS of on-board RAM is an exciting alternative. A set of VI blocks developed by LabNation allows you to configure and read out the SmartScope with ease.

This article aims at explaining all steps necessary to create the GUI shown in the head illustration. Moreover, LabNation's interface software is 100% open-source and this document explains how to access more advanced features of this underlying library. To be able to trace the steps, you will need a SmartScope and a PC with a recent version of LabVIEW. We also assume that you already have some experience with programming in LabVIEW.
Behind this GUI is the block diagram shown in **Figure 1**. Notice that the bottom part is only needed to update the GUI — all control and acquisition steps are done by the topmost part. All steps and VI blocks are described below.

Using the SmartScope in a LabVIEW application

When using the SmartScope in LabVIEW, there are three phases to go through. This section lists them, together with all VIs for each phase.

Phase 1: Initializing the SmartScope

This phase searches for a physical SmartScope, and returns a reference to the SmartScope which is needed by all other LabNation VIs.

INIT: Initialize.vi
When executed, this VI polls every 500 ms whether a SmartScope is detected. This can be a SmartScope connected locally on the USB port, or a SmartScope shared over the network using the SmartScopeServer (this will be presented shortly). Once detected, this VI will upload a basic configuration to the SmartScope and cause it to enter Running mode.

The output of this VI is a reference to the SmartScope and also needed for all other LabNation VI blocks. In practice, this means the dataflow in your block diagram will be blocked until a SmartScope becomes available.

> The underlying software is 100% open source

Phase 2: Configuring the SmartScope

At some point in your application, you will want to configure the acquisition settings of the SmartScope. This includes voltage ranges of the analog inputs, or the depth of the on-board RAM. It is important to note that all changes will only take effect after you've executed the CommitSettings.vi.

VERT: SetVertical.vi

This VI allows you to configure the analog input channels of the SmartScope. These are the expected inputs:

Voltage range. Simply set the largest amplitude you expect to measure and the SmartScope will automatically change its divider and multiplier stages in order to optimize its range for this amplitude.



Note that in order to get the finest resolution for your signal, it is best to set the voltage range as small as possible, while still making sure your entire signal can fit within that range.

Offset. The value you specify here will be physically added to or subtracted from the input voltage. This allows you to zoom in on a certain voltage range not centered around 0 V. For example, if you want to measure signals between 0 V and 12 V, setting the voltage range to 12 V and the offset to 6 V will result in the highest measurement resolution.

Coupling. This input terminal allows you to specify whether you want to have AC or DC coupling. DC coupling results in absolute voltages, while AC coupling first subtracts the mean value from the signal before being digitized. AC coupling allows you to zoom in on signals with small amplitude but large offsets.

Channel. last but not least, you should specify whether you're configuring Channel A or Channel B.

HOR: SetHorizontal.vi

This VI is a simplified version which allows you to define very basic timing settings. If you want to have more control over the timing settings including the on-board RAM, see the last section. Here are the required inputs:

ViewportLength: allows you to define the length in seconds of the sequence to acquire and adjusts the sampling rate, so the Viewport buffer of 2048 samples corresponds to this length.

TriggerHoldoff: specifies the position of the trigger in seconds, relative to the centre of the Viewport.

TRIGGER: AnalogTrigger.vi

This VI allows you to configure a basic analog trigger, using the following inputs:

Channel: The analog channel you want to trigger on.
EdgeType: This VI only supports Rising, Falling or Any edge.
TriggerLevel: The voltage, when being crossed, causing a trigger event to occur.

COMMIT: CommitSettings.vi

An understanding of this VI is quite important. Any changes you make through the previous VIs will have no effect until the CommitSettings VI is called. Using the previously described VIs, all changes you make are made in shadow registers which

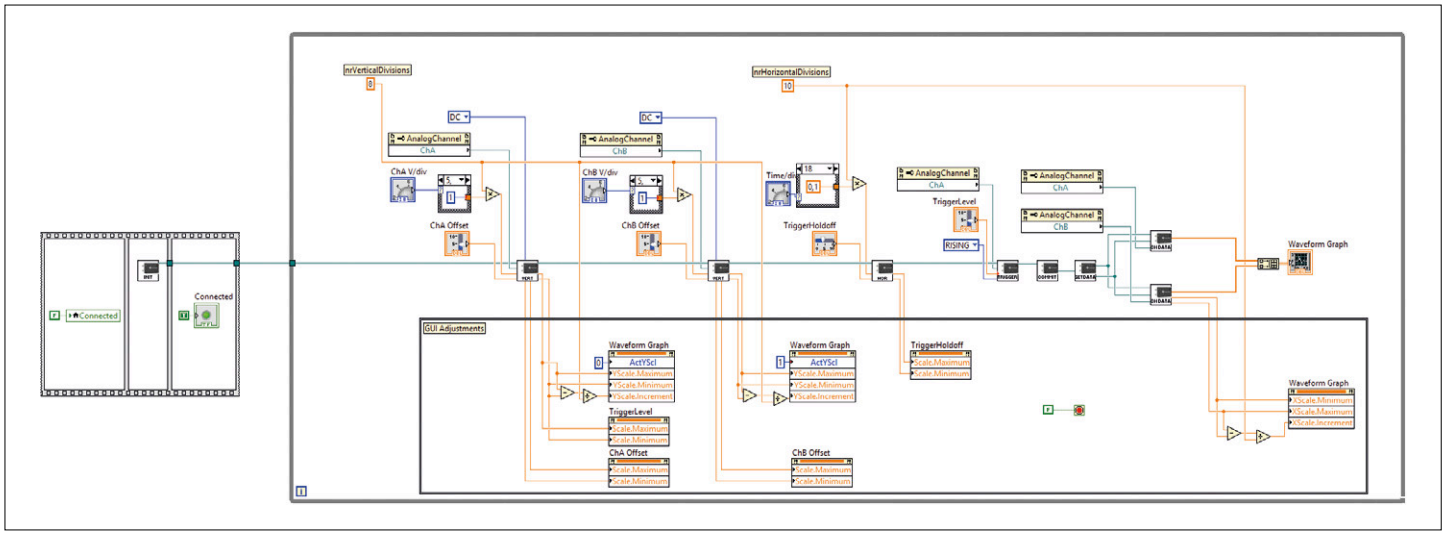


Figure 1. The block diagram in LabVIEW for the GUI shown in the head illustration.

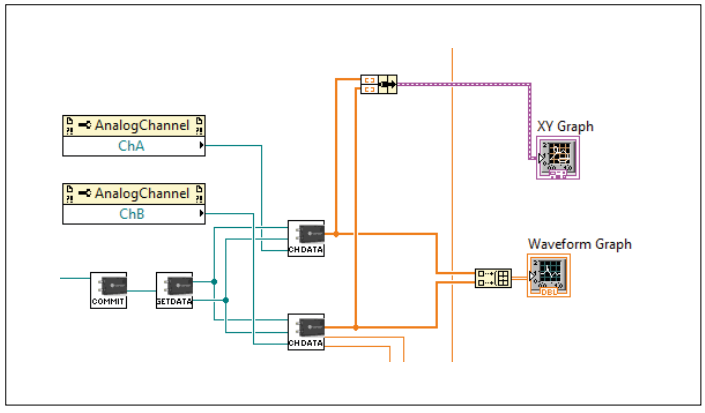


Figure 2. Adding an XY plot is simply a matter of wiring.

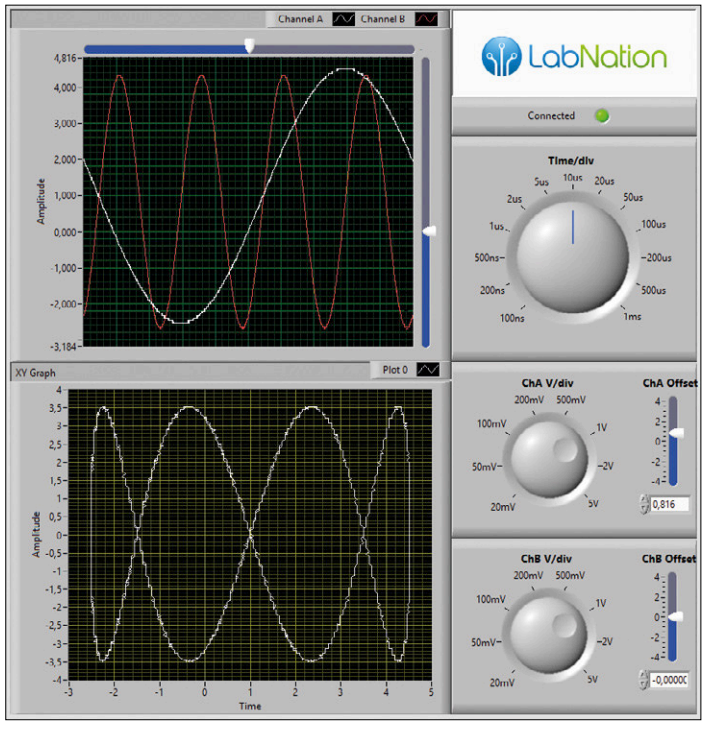


Figure 3. The GUI is rearranged and shows an XY plot.

are copied in one shot to the real registers at the moment you call the CommitSettings VI. This ensures acquisitions are made with a complete configuration.

Phase3: Reading out SmartScope data

You can fetch the latest data package from your SmartScope at any moment. A datapackage is a coherent set of information, containing the acquired data for all channels as well as information regarding the configuration used during the acquisition. It is important to have all of this data in one package, to ensure all data from different channels was acquired at the same instant.

GETDATA: GetLatestDataPackage.vi

This VI simply fetches a reference to the latest datapackage received by the SmartScope driver. In case of acquisitions with a very long ViewportLength, it can happen that this VI is called multiple times during the same acquisition. In such cases, the same reference to the last datapackage will be returned.

CHDATA: GetChannelData.vi

As explained above, a datapackage contains all data acquired at the same time for all channels. The GetChannelData VI simply returns the viewport data for the channel you specify, presented as a 1D array of floats.

The Viewport data fetched by this VI are up to 2048 samples which span the duration you specified in the SetHorizontal VI. This VI requires a reference from the GetLatestDataPackage VI, as well as the analog channel you want to get the data from.

Adding an XY plot

With the SmartScope data available to your LabView program, it is incredibly easy to add new representations fast. For example, adding a XY plot is simply a matter of wiring in LabVIEW's XY Chart VI (**Figure 2**).

This allows you to re-arrange the GUI easily, as shown in **Figure 3**.

Accessing more advanced DeviceInterface functionality

The software which interfaces with the SmartScope is 100% open source. This DeviceInterface library is the only piece of

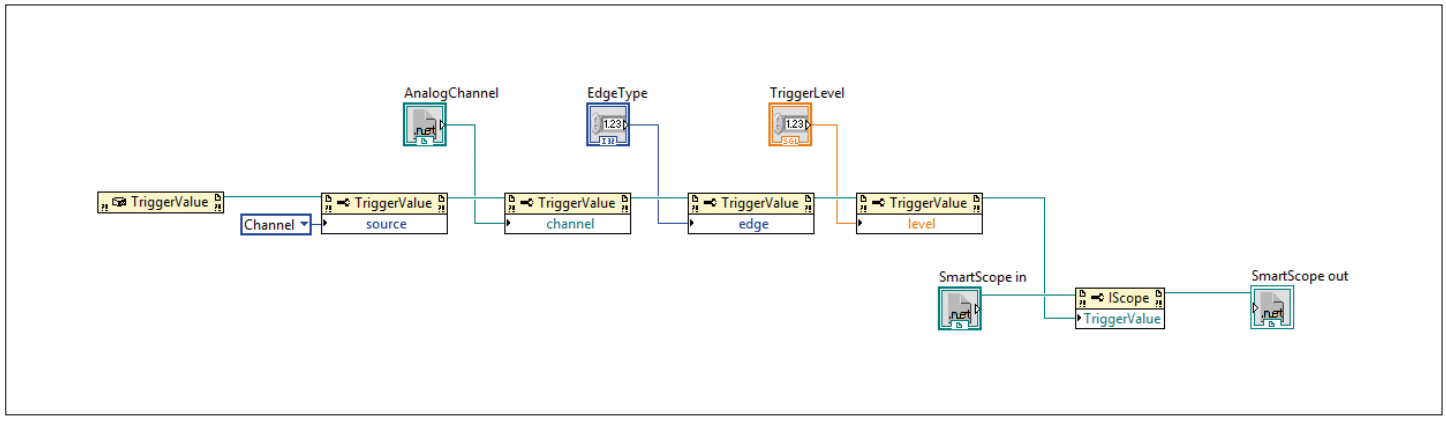


Figure 4. The diagram of the AnalogTrigger VI.

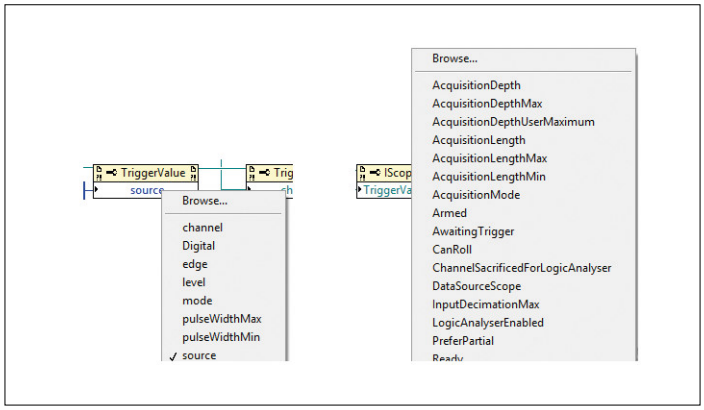


Figure 5. Properties of a TriggerValue block and an Iscope block.

code between your GUI and your SmartScope. As such, all functionality used by the official SmartScope app is accessible, also to LabView. Even though the VIs presented allow only basic operations, it is easy to use these more powerful features. As an example, let's set up a slightly more advanced pulse-width trigger. The trigger should only fire when a pulse width between 10 µs and 100 µs is detected. While this would not be possible using the VIs described above, the SmartScope supports this through the DeviceInterface, and hence we can use it from LabView. And it's not all that difficult.
To start, open up the AnalogTrigger VI and copy-save it to AnalogTriggerPulse. Have a look at the block diagram in **Figure 4**.

You see this diagram contains new blocks. LabView takes the DeviceInterface library and automatically creates these blocks for all functions inside the library. To give you an idea, simply click on any Property of a TriggerValue block, e.g. on 'source' (**Figure 5**). You will see a nice drop-down list of all properties you can set. Click also on the 'TriggerValue' Property of the IScope object on the bottom-right: you'll get a list of all Properties of the SmartScope you can set. These are explained on the DeviceInterface section of the LabNation wiki [1] and 100% of the source can be found on LabNation's GitHub account [2]. If we want to change the trigger from simple edge-detection to pulse-detection, add the 3 blocks shown in **Figure 6**. You can simply copy-paste the blocks from Figure X and change their properties to 'mode', 'pulseWidthMin' and 'pulseWidthMax'.
To create the 'Pulse' constant, right-click on the 'mode' terminal, select Create->Constant and LabVIEW will automatically present you a list of the various possibilities. The other 2 constants define the minimum and maximum pulse width, which are now fixed, but you can easily make them adjustable by using a Control instead of a Constant.
Voila, when you use this new subVI in your main VI, your SmartScope will reject all triggers which are shorter than 10 µs or longer than 100 µs.

With this new set of special VI blocks you can use the SmartScope in many LabVIEW applications.

(160045)

Web Links

[1] http://wiki.lab-nation.com

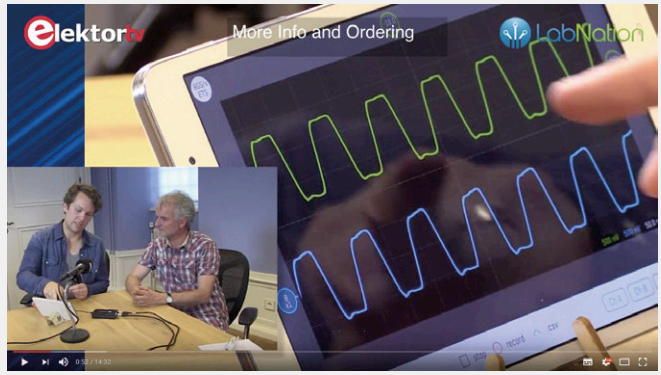
[2] https://github.com/labnation

On Elektor TV:

watch Riemer Grootjans explaining basic and advanced functions of the Smartscope:

https://youtu.be/SJsNiJBbs0w and

https://youtu.be/JgHt5zTEZF4



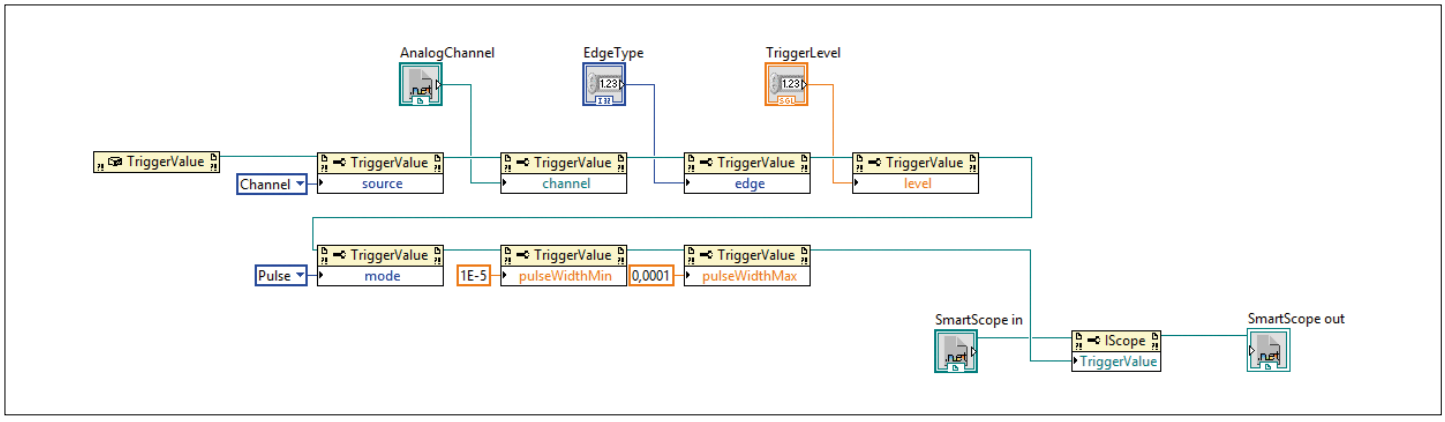


Figure 6. Three blocks are added for pulse detection: mode, pulseWidthMin and pulseWidthMax.

Fragile LED Filaments

Many LEDs on a glass carrier

By **Harry Baggen** (Elektor Labs)

LED-lamps are nothing new anymore — they are available in a wide variety of types and sizes. Most of these try to imitate the shape of ordinary incandescent lamps. But unfortunately the illumination pattern from many cheap LED lamps is not all that good. The manufacturers try to solve this with the aid of elaborate LED arrangements and reflectors inside the lamp. Also, while the shape may approach that of an incandescent lamp, it nevertheless doesn't look all that attractive when screwed into an open light fitting, because of the alternative construction.

Recently a new variant has made an appearance, the so-called LED filament lamp. In these, instead of individual LEDs, a so-called LED filament is used, which is a small rod supporting a large number of LED chips in series. The forward voltage is so high that a series connection of a few filaments can, via a rectifier, be connected directly to the line voltage. The filaments resemble the old-fashioned incandescent filaments somewhat and foremost have a major advantage over ordinary LEDs in that they radiate light in all directions (at least with the better implementations).

When these filaments are manufactured a so-called chip-on-glass technique is used, where the semiconductor material for the LED chips is directly applied to a small glass-like rod. This small rod is then coated with a layer of phosphor, where the spe-



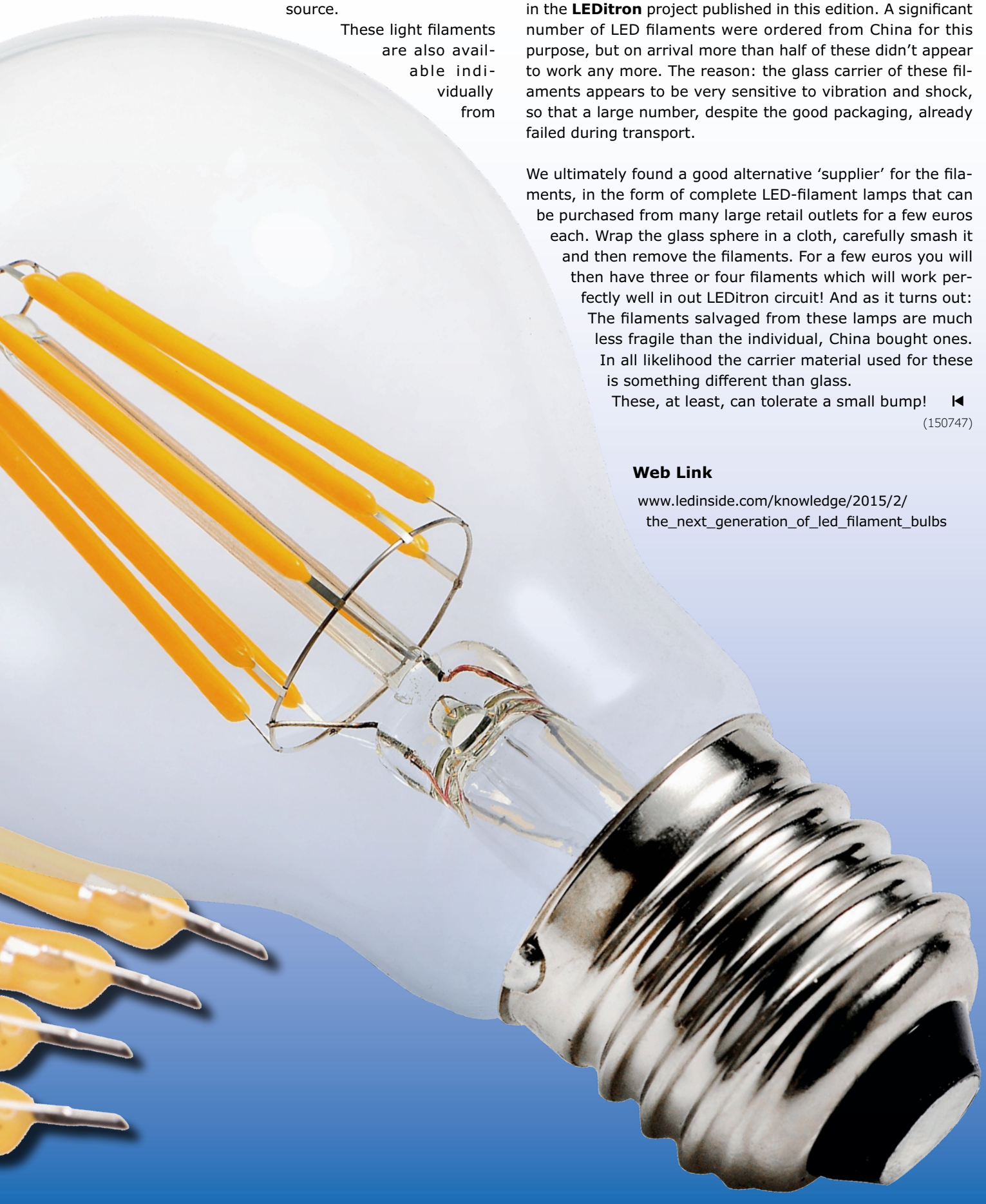
cific phosphor mixture and the color of the LEDs together determine the final light spectrum. At each end of the rods a small metal contact is attached, for connecting it to a voltage source.

These light filaments are also available individually from Chinese suppliers and that gave us the idea to 'abuse' them for other purposes. In this case, the idea was to make a large seven-segment display from them and this ultimately resulted in the **LEDitron** project published in this edition. A significant number of LED filaments were ordered from China for this purpose, but on arrival more than half of these didn't appear to work any more. The reason: the glass carrier of these filaments appears to be very sensitive to vibration and shock, so that a large number, despite the good packaging, already failed during transport.

We ultimately found a good alternative 'supplier' for the filaments, in the form of complete LED-filament lamps that can be purchased from many large retail outlets for a few euros each. Wrap the glass sphere in a cloth, carefully smash it and then remove the filaments. For a few euros you will then have three or four filaments which will work perfectly well in out LEDitron circuit! And as it turns out: The filaments salvaged from these lamps are much less fragile than the individual, China bought ones. In all likelihood the carrier material used for these is something different than glass.

These, at least, can tolerate a small bump! ⏮

(150747)



Web Link

www.ledinside.com/knowledge/2015/2/the_next_generation_of_led_filament_bulbs

MOS Clock 5314 (1974)

With digital precision and seconds readout!



mos~clock

Elektor's "mos clock 5314" (*sic*) was a raving success even many years after publication and the number of boards sold for it reached K levels. Can lessons be learnt from this blockbuster publication originally from 1974, the decline-of-TTL era? Fortunately, no.

By **Jan Buiting**, Editor-in-Chief

My reasons for doing reverence to the MOS Clock 5314 here in *Retronics* are twofold. First, and portentously, it's the mention of "your MOS Clock" as a friendly aside in correspondence I keep getting from readers writing: "a copy the plans please Jan of Elector board number [*1234*] from year [*<1980*] which I just found but never finished". Some call it nostalgia; I prefer "the attic-and-retirement effect".

In any case, if more than one reader remembers the Elektor MOS Clock to the degree of mentioning it to me to get a photocopy of another article, then it must have been a good one at the time (like *Junior*, *SC/MP*, *Crescendo*, *Formant*, *Mondrian*, *Filmnet Decoder*). The second reason I admit is trivial at best — I stumbled across the prototype of the MOS Clock 5314 in the storage space for Retronics gear and other edebris while looking for something else. MOS Clock in hand I tested the *déja-vu* effect on a non-random selection of colleagues here at Elektor HQ in Holland, and two responded with a nod of approval: Harry Baggen and Jan Visser, both Elektorians with deservedly heritage status. "Yes the MOS clock, good one! Man, where did you find it?"

Time warp, time obsession

As we discovered with our 2015 "Propeller Clock" and other publications in this the microcontroller-dominated era, Elektor readers are crave for circuits of clocks with an unusual readout or face (the RhineTower Clock comes to mind). Back in 1975, "digital" was all the rage on anything clock. I personally remember, at the age of 17 and on a student budget, my insistence on having a wristwatch with a seconds hand which in my mind rivalled a Digital watch for accuracy.

It was the time when sticklers for precision started their clocks to the final, elongated beep signal heard on the radio (rather than by radio or satellite as we have today and everyone's still running late). Digital (alarm) clocks with a seconds display must have existed but were in the realms of MIT scientists *et al*.

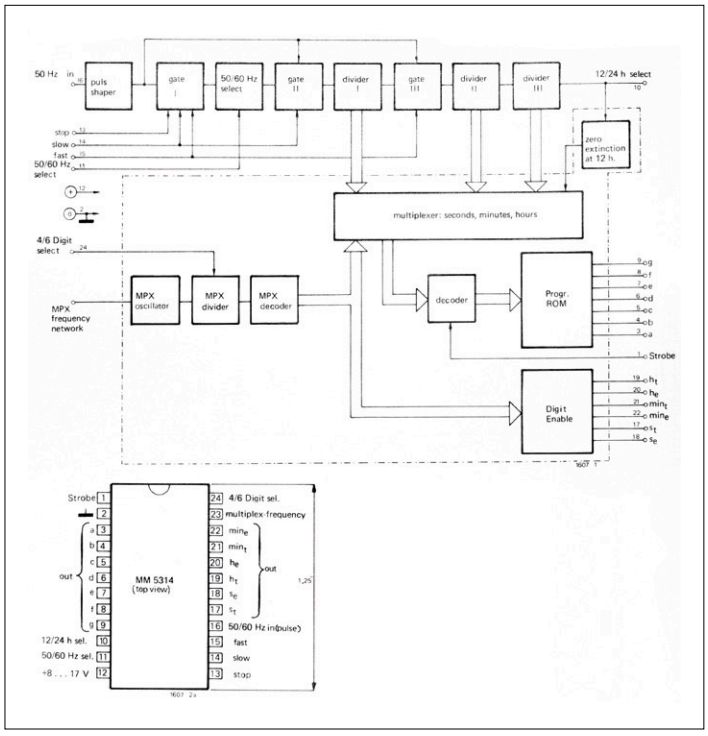


Figure 1. Elektor's 1974 rendering of the architecture and pinout of a classic chip: National Semiconductor's type MM5314 clock IC.

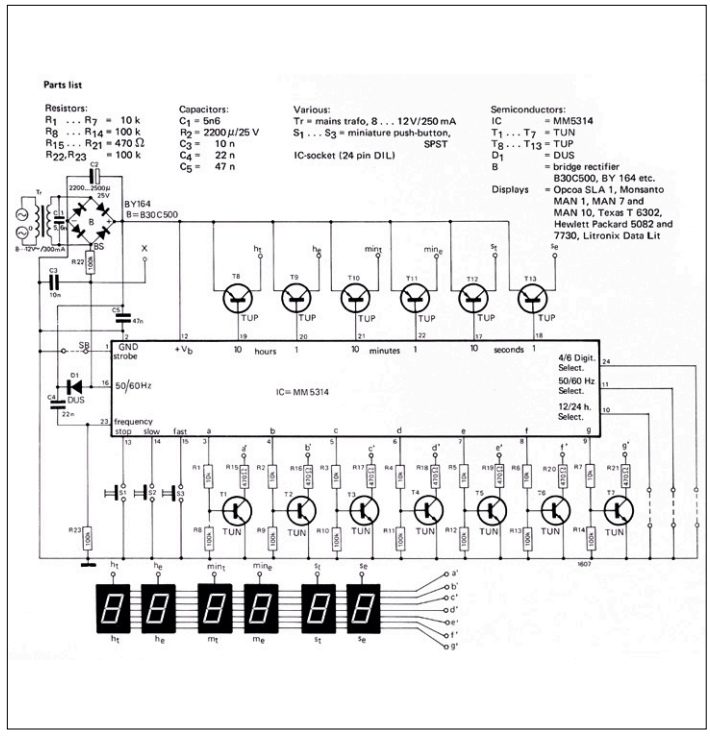


Figure 2. Schematic and enclosed parts list of the MOS Clock 5314 from 1974.

MM5314 bombshell

Besides the 00-59 seconds readout (suggesting Digital Precision!) the use of a single integrated circuit (IC) was sensational to the budding electronicists in 1974. And M-O-S, man, that's metal oxide semiconductor! In the years before, clocks using dozens of power hungry, noisy TTL ICs had been published by many magazines in such quantities and variations as to reach the *ad nauseam* level with pros and hobbyists alike.

National Semiconductor's MM5313 turned out a glorious candidate for killing all that TTL in a clock but it was *expensive*, a word that infuriated Elektor's predominantly Dutch lab staff at the time, and still does. Happily the MM5314 is the BCD-less, 24-pin, smaller brother of the 5313. As the 1974 Elektor article extols, the '14 is *cheaper* and that BCD output stuff on the '13 to drive 7-segment displays straight — man, multiplex is cleverer and *cheaper* and with a few of our *cheap* TUNs and TUPs we *save* a *guilder or two* to put in the *bank*.

Applying more Dutch economies, we can now fathom why the article as early as in the third paragraph states that you can actually save money by omitting the two digits for the seconds display if you don't need them ("digital accuracy" down the drain).

That MM5314(N) is a product from National Semiconductor, which curiously are nowhere mentioned in the 1974 article, while the IC's block diagram and pinout appear to be redrawn by Elektor (**Figure 1**) rather than copied from the NS databooks I am sure the Elektor designers had at that time (blue! they were). A no-fake MM5314N is both scarce and *expensive* today (try EBay) making it a veritable collector's item. It derives its clock accuracy from the 50-Hz or 60-Hz (selectable) AC powerline frequency, which seems a good laugh with all that

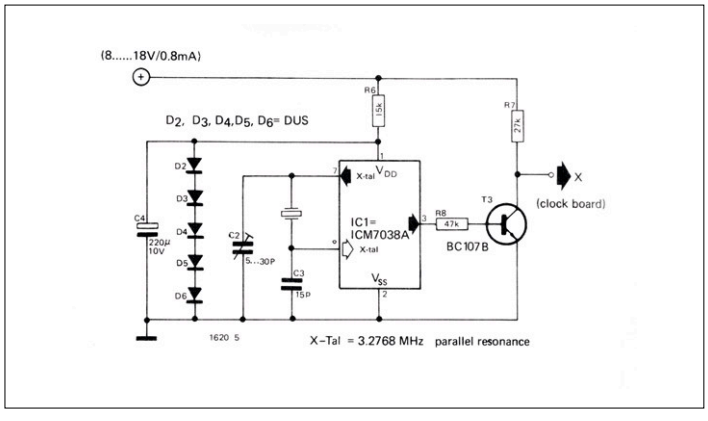


Figure 3. This quartz crystal controlled reference timebase prevents the MOS Clock 5314 from losing sync during power outages.

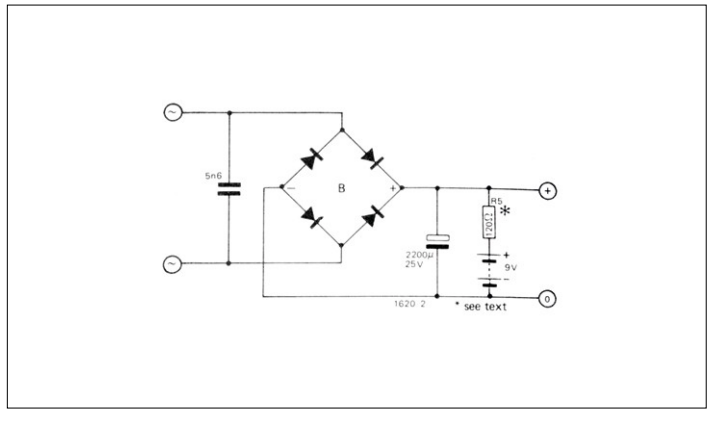


Figure 4. Add this fantastic battery backup circuit to keep the MOS clock ticking when unplugged or without AC power.

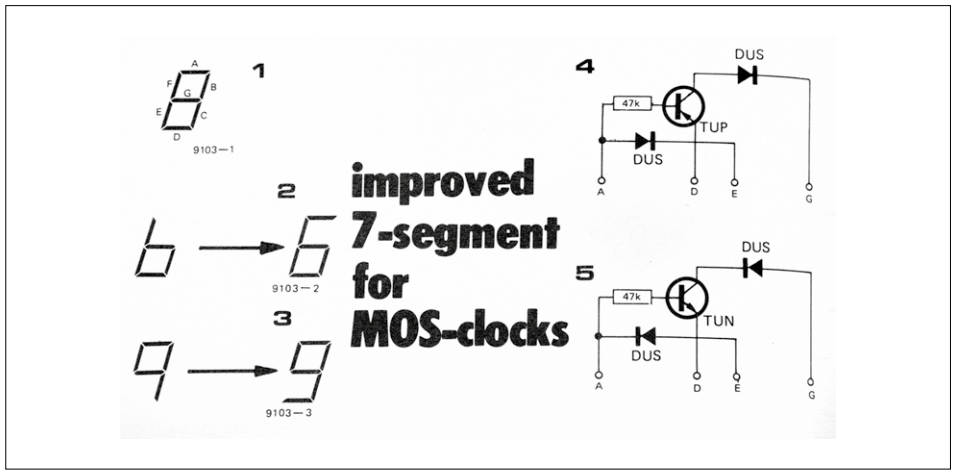


Figure 5. 1975-style aestheticizing of the '6' and '9' renderings on the clock display, using TUN, TUP and DUS semiconductors available by the pound then from Acme Corp.

Nobody expects Customs & Excise



Back in the 1970s the precious prototypes of Elektor Labs projects had to be transported up and down Europe for showcasing at shows and exhibitions. Borders were very real in those days, and "technology" especially electronics was considered suspicious and expensive by default by anyone paid to guard their national borders. So whenever the MOS Clock 5314, the Crescendo, or the ElektorScope was to embark on a journey to an electronics show in, say, Paris, Elektor lab staff had to fill out 5 forms each with 6 carbon copies for each apparatus to satisfy Customs & Excise in Holland, Belgium, and France, two times each, not forgetting certified translations.

No frolicking with French or German customs. *Pas de papiers*/*Keine Papiere*? (no papers?) ➞ each item valued at NLG 10,000 by default ➞ immediate payment due of NLG/DM 1,000 for taxes ➞ No cash? ➞ Goods confiscated, and one hotel stay for the driver.

Weary of Kafka, red tape and fines when all they wanted was to be on exhibitions, the Elektorians struck an agreement with Dutch Customs for two officers to come inspect "the goods" and do the paperwork "locally" before sealing (?) and allowing the van to be loaded. Not forgetting to accept a cup of coffee and talk shop in the labs. Two officers from Customs Sittard dept. typically arrived late on Friday afternoons and became ardent Elektor fans.

Helped by these gentlemen, a special triple-base-address sticker was designed for securing to each prototype or piece of lab equipment large or small that was to cross borders. It got an impressive stamp from Customs and was further adorned by Elektorians with tech numbers both typed and scrawled. The horrid paperwork got reduced to just a few "carnet" sheets and carbon copies, and the equipment stickers with their hippie founts did the rest. Worry not, it was legal, at least in Holland. For everyone's amusement, here is the sticker I found very secure on the bottom panel of the MOS Clock.



GPS and DCF77 atomic precision around these days but makes for utterly acceptable stability over the long term — say 6 months or longer — and it probably outdid my 1975 seconds+ wristwatch with its wind-up loudly ticking clockwork. Using the grid frequency for reference means there are no adjustments to be made on the MOS Clock — if it seemed off by a second or so, I guess you simply complained to your electricity supplier.

The BCD-less MM5314 is unable to do any serious display driving without resorting to multiplexing and a bunch of driver transistors to do the display segment switching at precisely the right timing. And we need a power supply of course. All was added by Elektor in good Dutch fashion resulting in the schematic in **Figure 2**. And a printed circuit board (PCB), very obviously designed using sticky tape and rub-off Letraset symbols, there being no CAD back in '74.

As is easy to see, this version of the clock simply forgets the time if no power is applied for more than 200-odd milliseconds (thanks, C2) and has to be readjusted whenever power is restored.

The main clock board and the display board were two separate PCBs available from Elektor under numbers 1607A and 1607B respectively, priced at £1.15 and £0.85 again respectively, plus VAT. With confidence, two important add-ons were announced in the final paragraphs of the 1974 article: quartz crystal control and battery backup. Remember, the Real-time Clock (RTC) was still down the rabbit hole back in 1974.

The sequel

Perusing my Elektor 1975 Cumulative Index I easily found the article describing the MOS Clock extensions promised in December 1974. It was: *MOS Clock (2)*, June 1975, pp. 634-636. A further article surfaced, *improved 7-segment for MOS clocks* in the July & August 1975 edition, p. 748.

The quartz crystal timebase reproduced here in **Figure 3** adds one full integrated circuit, the *expensive* Intersil ICM7038A. As far as accuracy is concerned, I doubt it rivals that of the grid frequency in the long term but it's of course necessary if the clock is to continue ticking during

power outages when the emergency supply (**Figure 4**) is supposed to take over. A 9-volt mini power pack (*sic*) gets trickle-charged off the raw DC supply voltage. The article talks about "refresher current" on a "dry battery" which I am unclear about and find highly dubious, possibly dangerous.

Both glorious enhancements to the original MOS Clock 5314 dwell happily on a single PCB no. 1620 priced £0.70 plus VAT. No that's not 2016; none of these boards have lived up to vigorous Dutch broom sweeping in our storage spaces from time to time.

Gosh, those numbers '6' and '9' look sad on a 7-segment display so a TUP, TUN and DUS are thrown in as shown on **Figure 5** to make an additional horizontal segment light. Due to the additional voltage drop it lights a tad less bright than the others.

The attic find

I found the prototype of the MOS Clock 5314 in the section of the attic granted to Retronics at Elektor House, and it appeared to be in good nick. Suspiciously, the power cord was bereft of its plug so I did a quick resistance measurement on the two AC in leads and the case first, then fitted a suitable plug.
The unit powered up perfectly showing a random time, and with no manual in sight was easy to adjust using three pushbuttons on the rear side of the case.

Inside

Not in the least curious I opened the clock's simple aluminum case. Step into it with me (**Figure 6**) and notice that the '6-and-9 aesthetic update' is duly installed. As are a TUP (BC557) and a DUS (1N914). However a number of things are sadly absent:

- mains fuse and switch (all legal at the time);
- battery backup;
- ICM7038A reference timebase;
- a leaky electrolytic (sorry, Badcap Hunters)
- a 110-V primary on the power transformer (sorry, US readers).

Also, erstwhile constructor of the clock, Jan Visser opted to move the three buttons for the clock adjustment from the vertical display board all the way to the back panel (can you think why?). The 50/60 Hz AC line frequency selection was done with a toggle switch also fitted on the rear panel. Switching to 60 Hz at an electronics show in Paris or London without anyone seeing must have been a good trick to extend one's stay at the show by 20%. Or the other way around: a +16.6% time warp in the US of A — where the MOS Clock 5314 never went to despite the awesome customs stickers. But then it wouldn't have powered up over there (can you think why not?). And then I suddenly realized this extremely advanced and accurate clock has no alarm to wake me up in time to do the next *Retronics* installment.

(160102)

ESTD 2004

www.elektor.tv

Retronics is a regular section covering vintage electronics including legendary Elektor designs.
Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

Especially 4U

To celebrate the resurfacing of the MOS Clock 5314, the original article from Elektor December 1974 has been scanned and made downloadable for you all at www.elektormagazine.com/160102.

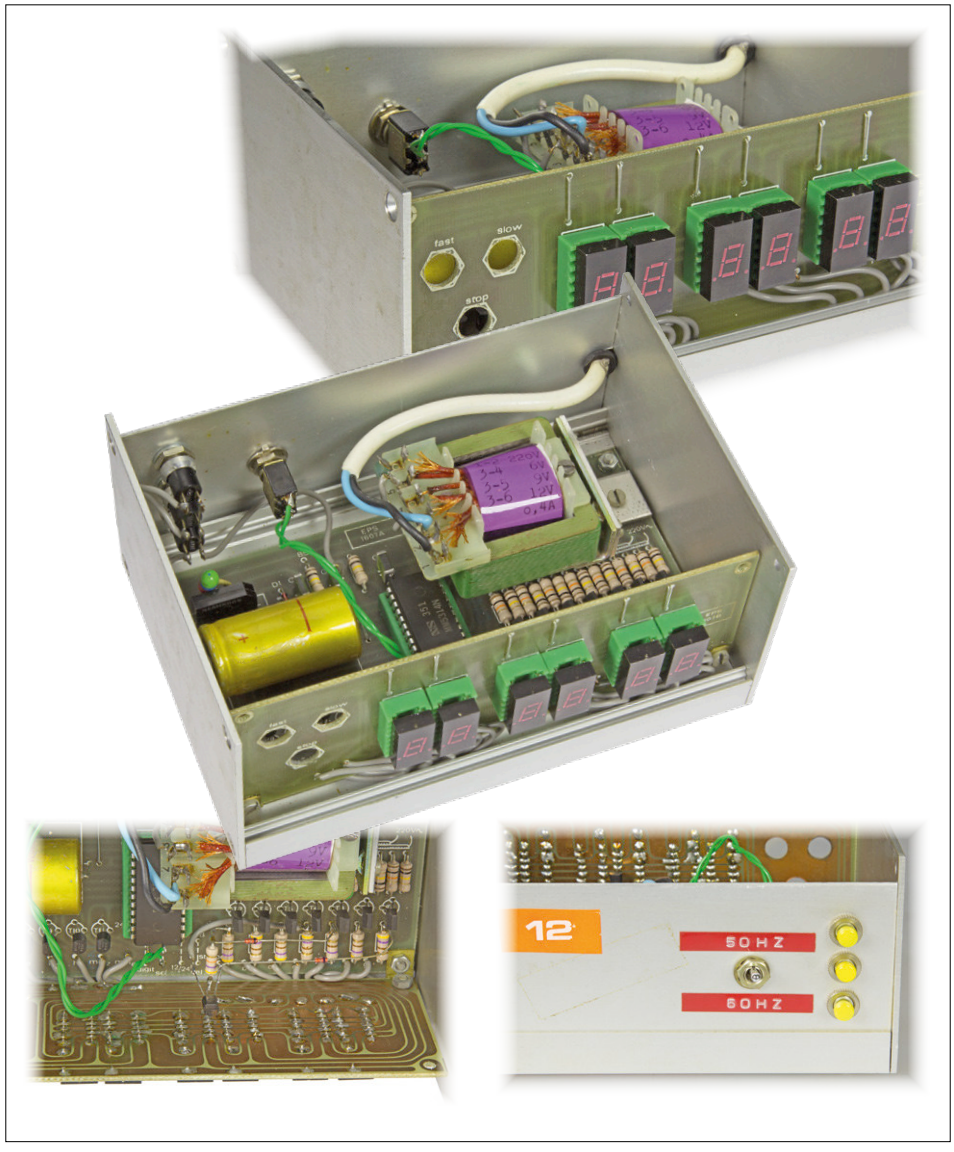


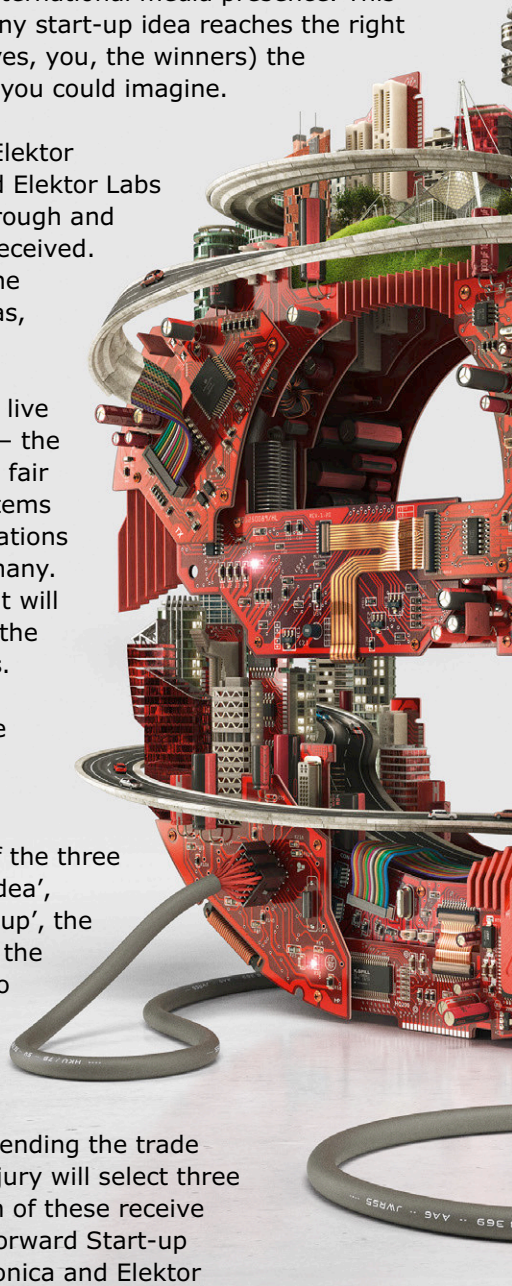
Figure 6. Inside the MOS Clock 5314 case. Its actual dimensions are 150 x 105 x 65 mm. The transparent, red display bezel has an off-white paint layer at the inside. I am uncertain as to how the black lettering was applied to the bezel — it's genius and stood the test of time.

Special Release:

Compiled by **Robert van der Zwan**

The Fast Forward is Worth More

The 'electronica Fast Forward Start-up Award powered by Elektor' initiative brings together, for the first time, the worldwide electronics marketplace, innovative technology and an international media presence. This will guarantee that any start-up idea reaches the right people, giving you (yes, you, the winners) the best possible launch you could imagine.



A jury consisting of Elektor Magazine editors and Elektor Labs engineers will sift through and evaluate all entries received. The best entries in the three categories Ideas, Prototype and Start-ups will be invited to showcase their ideas live at electronica 2016 — the world's leading trade fair for components, systems and electronic applications held in Munich, Germany. Here a final judgment will be made to pinpoint the three overall winners.

Any entry is welcome ultimately October 4, 2016.

As a finalist in one of the three Award categories: 'Idea', 'Prototype' or 'Start-up', the inventor(s) will have the unique opportunity to participate in the electronica Fast Forward Platform.

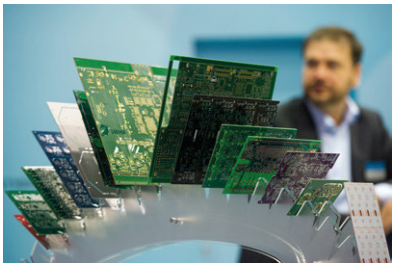
From the finalists attending the trade fair an international jury will select three overall winners. Each of these receive an 'electronica fast forward Start-up Award', which electronica and Elektor

What's the electronica Fast Forward Start-up Award?



The electronica Fast Forward Start-up Award is a partnership between electronica, the world renowned trade show on electronics in Munich, Germany, and Elektor. Far more important is the fact that this prestigious award will give its winner access to the international market of electronics starting November 11, the day the international jury will make public those who win 150,000 euros worth of marketing tools and services. As a matter of course this announcement will take place right at the electronica show. In the words of Campaign Manager René Bohne: "We will bring together everyone: makers, hobbyists, engineers and the business world."

Interesting Submissions? You Bet!



Campaign leader René Bohne, also a member of the jury, knows a thing or two about electronics. He also wants everyone to know this: "Most submissions we received are really interesting and it's great fun to learn more about them. But I also believe that there are many other ideas out there. Bring'm on!"

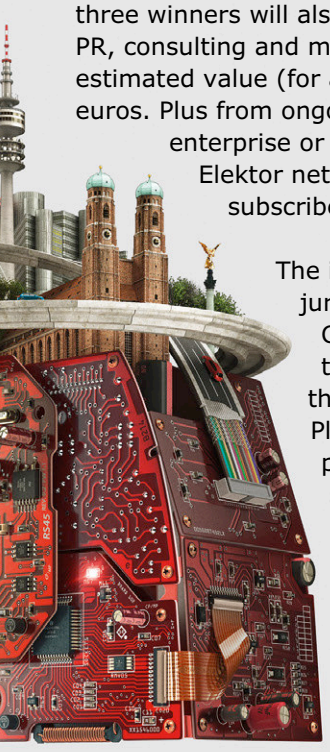
READ ONLY MEMORY

Elektor magazine and its parent publishing company boast a long and rich history. In this space we picture a gem from the past.

The electronica Fast Forward Start-up Award is a pretty prestigious award in the field of electronics — some experts outside of Elektor suggest 'the most prestigious'. And indeed Elektor has much experience to draw from. A blockbuster, the International Circuit Design Competition was launched in the July & August 1995 issue of *Elektor Electronics* magazine. Laurent Lamesch from Luxembourg was awarded First Prize, a THS720 oscilloscope sponsored by Tektronix. Laurent submitted a DIY 50-MHz logic analyzer with an input word width of 16 or 32 bits. The jury was unanimous

PEOPLE NEWS • Angela Marten is not only the Project Manager for the electronica trade show in do with the electronica Fast Forward Start-up Award • Clemens Valens will be the Jury Manager for the Elektor, putting all his efforts in a truly professional judgment by the Elektor team • Before the finalists course), René Bohne will have made sure that everyone on the globe will know what the electronica Fast

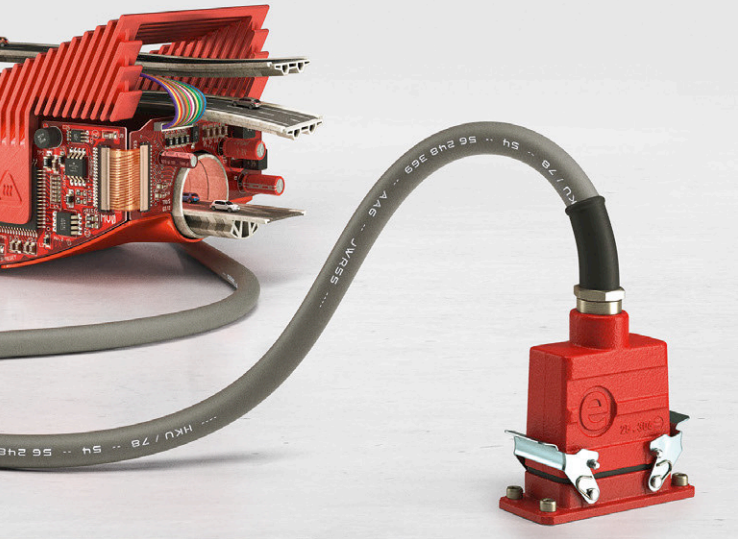
Start-up Award Than €150,000



will be presenting on November 11, 2016. The three winners will also benefit from international PR, consulting and marketing services with an estimated value (for all three) in excess of 150,000 euros. Plus from ongoing endorsement for their enterprise or idea via the international Elektor network of over 250,000 paying subscribers.

The innovators selected by the jury in the category Start-up Companies are expected to be present in person at the electronica Fast Forward Platform along with their product, with the presentation space subject to a fee of €1000 maximum. All selected participants in the category Ideas and Prototypes qualify for complimentary exhibition space, available for free.

If you feel a strong urge to be part of this, go to www.elektor.com/electronica-startup-award



Munich, she is also coordinating everything that has to electronica Fast Forward Start-up Award on behalf of enter the ring in Munich (figuratively speaking of Forward Start-up Award is about. … … …

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.



An interview with **Angela Marten**, Project Manager at Messe München, about the 'Fast Forward' events at electronica 2016

Whence the interest at the Munich Exhibition particularly for start-ups and new business ideas?

electronica is the world's leading trade show and an important platform for the electronics business world. We bring markets together by facilitating the exchange of new ideas and forging important contacts. Start-ups provide a vital stimulus for continued industry growth. Consequently we should focus on them to help establish links with industry and encourage ongoing development.

What are your expectations of the Fast Forward Start Up Platform / the Fast Forward Award powered by Elektor?

The Fast Forward platform at electronica is an exciting new initiative bringing innovative new ideas together with industry's movers and shakers. Start-ups will get the chance to forge international business partnerships and source investment. For our exhibitors it provides a focal point for innovative solutions. Finalists will be showcasing new concepts which for sure will also fascinate our many visitors.

What areas of innovation do you find particularly exciting?

I just find new ideas in electronics fascinating; it doesn't matter if it's something totally innovative and new or whether it's a completely new take on an existing solution which challenges your preconceived ideas.

How do you reach out to anyone with a bright idea or innovative start-up concept?

The electronica trade show as part the Munich Exhibition Center already provides access to a broad network of contacts. Together with Elektor's media exposure within the vibrant and growing Maker scene, start-ups get unparalleled networking opportunities and attention from investors and entrepreneurs via media channels using videos, blogs and newsletters.

Which criteria would you consider the most important in the evaluations?

A panel of judges makes the final decision based on the participants product pitch. They will be looking at the level of technical innovation and its market potential. Design feasibility, estimated manufacturing costs and product sustainability are also important.

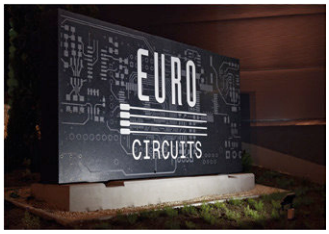
Compiled by **Robert van der Zwan**

The Big Red, What's That?



Elektor's editorial staff are working hard on their newest brainchild. The Big Red is itself a brainchild, but more importantly: this yearbook is also about brainchildren — the intellectual offspring of start-ups, homelabs and prototypers. Anyone who wants to get started in electronics should be in The Big Red. Does The Big Red stand for spirit and sparks? You bet! It's simply the place to be for anyone who wants to put an idea to commercial use. See www.elektormagazine.com for the latest.

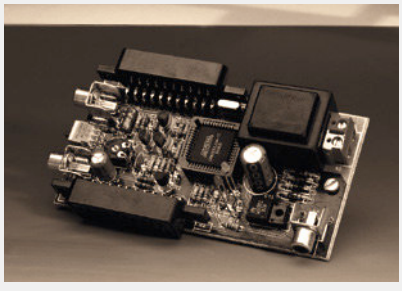
Eurocircuits? 25 Years!



The journey started 25 years ago in a dark pub in the center of Mechelen, Belgium. There, Luc Smets and Dirk Stans shared their dream to start their own company. Today, Eurocircuits generates a yearly turnover of more than €20 million. And Eurocircuits is part of the Elektor family of course, since Eurocircuits and Elektor are partners in the Elektor PCB Service. Congrats Dirk and Luc!

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.



Macrovision was a copy protection technology employed by the rightful owners of Hollywood 'blockbusters' to protect their copyright on VHS/PAL video tapes. In the November 1997 issue of *Elektor Electronics*, a Video Copy Processor is described "which defeats the pulses and other purposely introduced interference that prevents VCR-to-VCR copying of Macrovision-protected VHS/PAL video tapes". Of course, Elektor did not and would not promote illegal copying, so we read "You are entitled to make one copy for backup purposes only." The project was a massive success.

The Emergence of a New Wild West in Cyberspace

elektorethics by **Tessel Renzenbrink**

By 2020 the security of the internet will be so abysmal, that anyone who goes online must assume their systems and data are already compromised. The digital realm will become a new Wild West where anyone who wishes to travel the wires in safety will need to organize it for themselves.

That alarming vision is one of five scenarios describing what the future of the Internet may look like in 2020. The scenarios are developed by the Center for Long-Term Cybersecurity of the University of California Berkeley with the goal of identifying emerging issues that could become a threat to internet security. Created with the input of more than a 100 experts from the private sector, nonprofits, and governments, the scenarios are described in a report published this spring.

Intentional Internet of Things

One of the scenarios considers the consequences of massive adoption of the Internet of Things (IoT). In our present day, there are already so many concerns about the security of the IoT that some have dubbed it the IoV, the Internet of Vulnerabilities. The threats are plentiful: IoT devices often leave the factory without proper security because manufacturers pay too little attention to it, customers don't ask for it and governments don't regulate it. Another vulnerability is the sheer volume: if the IoT takes off, billions of devices will newly come online. All in need of security maintenance such as frequent firmware updates and patches. To organize that at the scale of billions will prove to be a daunting task. Moreover, replacing legacy devices that are no longer supported with software updates, is quite a logistical challenge when they are as ubiquitous as sand on the beach.

The scenario is called Intentional Internet of Things because it assumes governments and companies will roll out the IoT intentionally to address big societal problems like climate change. Such applications can bring benefits, according to the report, but it also raises the stakes for cybersecurity considerably. "Perhaps the greatest risk lies precisely with the greatest benefits: as communities get more networked, they will also grow more vulnerable."

To read the entire article go to:
https://www.elektormagazine.com/articles/the-emergence-of-a-new-wild-west-in-cyberspace/10111

PEOPLE NEWS • Tanja Pohlen is the Project Manager for The Big Red, the new annual publication from to the fore every year • A new editorial staff led by Tessel Renzenbrink (known for her contributions space and credit to any electronics engineer who wants to combine technology with business • Patrick promote The Big Red on Elektor TV ... ...

New Wild West



Elektor bringing start-ups, homelabs and prototypers to Elektor Ethics) will compile the Big Red, giving Wielders and Jan Buiting used a €2000 banknote to

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: **Bill Morgan**

Age: **64**

Education: **Electronic and Electrical Engineering BSc from Surrey University; MBA from Strathclyde University; Diploma in Business Administration from Singapore Institute of Management; Chartered Engineer in UK**

Currently working on: **I am putting together an offering for teaching kids coding in three different age groups**

Who is Bill Morgan?

A person who is interested in all forms of technology but especially electronics, and who wants to pass on his experiences to future generations.

What will be the most key electronics development?

Artificial Intelligence will be the most interesting and simultaneously the most frightening field of endeavor. If we get it right it will liberate us. If we get it wrong it will imprison us or worse.

What makes Australia different from the US in terms of innovation in electronics?

Australia does not have Silicon Valley or anything even close, so we do not have an electronics or technology culture.

Which project are you most proud of and why?

I was one of many engineers working on the production line of the EMI scanner, now known as the CT Scanner. The scanner was invented by EMI, headed by Sir Godfrey Hounsfield, who in 1979 received the Nobel Prize for his work on the scanner.

Who are your (scientific) idols?

Professor Eric Laithwaite was the inventor of the linear motor. His invention is known better for the use as the propulsion for hover trains. Nicola Tesla is most famous for developing alternating current, but his work also led to advances in wireless communications, lasers, x-rays, radar, lighting, robotics, and much more. Then there is Sir Frank Whittle. He was a British pilot, aviation engineer and inventor who is best known for the invention of the jet engine. Last but not least: Sir Godfrey Hounsfield, an English electrical engineer who developed the diagnostic technique of X-ray computed tomography, the CT scanner.

What do you hope to accomplish within the next five years?

A nationwide program to teach kids coding and move the focus of Australia from the mining and agricultural to technology and innovation. ⏮

(160086)

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 October 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 2/2016 (July & Ausgust) is: **79ADF**.

The €50 / £40 / $70 book vouchers have been awarded to: Luc Vandormael (Belgium), Vinnie Pierdominici (USA) and Julian Muscat (Malta)..

Congratulations everyone!

			7		C	D						5	F		6
	9		A			2		F	3	6		8			E
			F				9	B	0	2	7			4	A
	4	E		A				C		5	8	2			7
			0		8	1			C	9	B			7	
				6		3		5			A		C		0
	A		C	7				2		3		9		1	
7				4	A					0	1		2		
		3		8	2					A	C				D
	D		E		F		6				4	C		A	
9		C		B			A		5		6				
	6			D	1	5			B	F		4			
8			5	F	4		2				9		B	0	
6	7			C	D	9	3	8				A			
A			9		E	6	B		7			F		D	
F		B	3						4	D		1			

1	5	B	C	2	E	6	0	D	3	9	4	A	F	7	8
8	6	3	E	1	A	9	7	2	B	5	F	0	C	D	4
2	D	F	4	B	3	C	8	E	0	7	A	1	5	9	6
0	7	9	A	D	F	4	5	8	1	C	6	B	E	2	3
A	E	0	B	3	D	5	4	F	7	6	2	C	8	1	9
C	F	8	3	6	1	A	9	0	4	B	D	2	7	E	5
9	4	1	6	7	B	8	2	A	C	E	5	3	D	F	0
5	2	7	D	C	0	E	F	9	8	1	3	4	6	A	B
7	0	6	5	4	2	B	D	1	E	A	9	8	3	C	F
3	8	2	1	0	5	7	6	4	F	D	C	E	9	B	A
4	A	D	9	E	8	F	C	3	6	2	B	5	1	0	7
B	C	E	F	A	9	1	3	7	5	0	8	6	2	4	D
D	B	C	2	8	4	3	E	5	A	F	7	9	0	6	1
6	1	5	0	9	7	2	A	B	D	3	E	F	4	8	C
E	3	A	8	F	6	D	1	C	9	4	0	7	B	5	2
F	9	4	7	5	C	0	B	6	2	8	1	D	A	3	E

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.

reichelt elektronik
Technology connects.

- More than 45 years of experience
- 24-hour shipping
- More than 50,000 products

RASPBERRY PIs

... from reichelt.co.uk, with low prices and fast delivery!



THE REICHELT ADVERT

rch.lt/vuk

FIND OUT MORE ►



HAS NEVER BEEN EASIER TO OPERATE!

MicroSD WITH MAX2PLAY SOFTWARE for your Raspberry Pi

With Max2Play, a browser-based administration interface for Raspberry Pi, you can install and administrate applications (Max2Play plug-ins) on a mini computer **with no knowledge of Linux.**

Scope of delivery:

- 16 GB micro SD card including Max2Play software
- 1-year licence code (expires automatically)



RASP M2P 16GB 1Y

£17.16

ALL THE CAPABILITIES OF MAX2PLAY!

http://rch.lt/m2p

DISCOVER NOW!

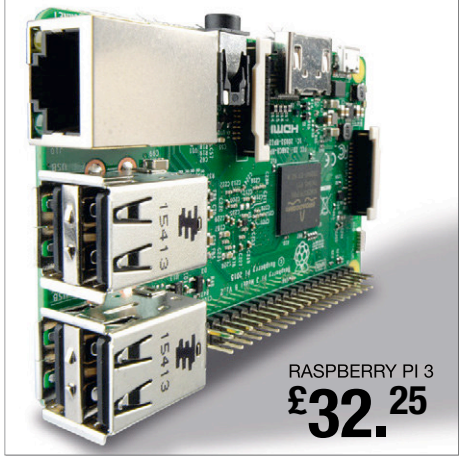
THE LATEST GENERATION!

RASPBERRY PI 3

with 4x 1.2 GHz & on-board WLAN

10x faster than the first generation!

- 1200 MHz ARM Cortex-A53 64-bit quad core processor
- 1024 MB RAM LPDDR2
- VideoCore IV dual core GPU
- WLAN, BT, LAN, USB, HDMI, 40 I/O, CSI, DSI, Audio, ...
- Supports Windows 10 IoT, Raspbian, RaspBMC, Arch Linux, ...



RASPBERRY PI 3

£32.25

NEW! KEEP EVERYTHING IN VIEW AT NIGHT TOO!

RASPBERRY PI NIGHT VISION CAMERA

with 8 Megapixel resolution

- Photo: 3280 x 2464 pixels
- Video: 1080p @ 30 fps 720p @ 60 fps
- Image sensor: 1/4"



RASP CAN 2

£25.71

JETZT ABONNIEREN & GEWINNEN!

Wir verlosen unter allen Newsletter-Neuanmeldungen eines Monats einen tollen Gewinn für Technikfans!

http://rch.lt/NEWSLETTER

INNOVATIVE PRODUKTE
ATTRAKTIVE PREISE
AKTIONEN & VORTEILE

JUST A FEW CLICKS TO YOUR AUDIO NETWORK PLAYER:

RASPBERRY / MAX2PLAY BUNDLE

including HiFiBerry DAC+ Light Shield

- Raspberry Pi 3 Model B
- Micro USB adapter (5.1 V / 2.5 A)
- Raspberry Pi & HiFiBerry housing
- 16 GB micro SD card including Max2Play software
- 1-year licence code (expires automatically)



RASP M2P BDL 1

£94.51

Daily prices! Price as of: 25.07.2016

Prices in £ plus statutory VAT, plus shipping costs · reichelt elektronik, Elektronikring 1, 26452 Sande (Germany)

Onlineshop languages:

PAYMENT METHODS:

SOFORT BANKING | VISA | amazon payments | PayPal

SHOP CONVENIENTLY ONLINE!

www.reichelt.co.uk

ORDER HOTLINE: +49 (0)4422 955-360

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