Edition 1/2016 | January & February 2016
www.elektormagazine.com

elektor

LEARN DESIGN SHARE

Let There Be LED!
Poised to take over all lighting

Active Crossover
for active loudspeakers

RV-3029-C2
Awfully accurate RTC chip on I²C

2-way radio for data links

eRIC Nitro

Here's your free NFC tag — win one of 500 prizes!

NFC with Elektor and ST

Android I/O Board (3) • ARM Microcontrollers for Beginners • BL600 e-BoB (6) • **Bottlenecks for USB 3.0** • **Circuit Triggered by Sparks** • **EAGLE Tips & Tricks** • eRIC Nitro • **Err-lectronics** • Ethical Robots and Robot Ethics • **Fridge Thermostat Calibration** • Hexadoku • **Let there be LED!** • **Line AC Switch Controlled by Bluetooth Low Energy** • **Magnetic Field Simulation with FEMM** • New Apps for the Red Pitaya • **NFC with Elektor and ST** • Q & A: LEDs • **RF Detector using an Arduino** • Steampunk Electronics • **STK600 Repair** • STMicroelectronics ST25TA Technology • **Tektronix Type 503 Oscilloscope** • **The Electronics Year 2015** • The RV-3029-C2 "Swiss Clock" • **Universal Alarm** • Waveguide • **Windows on the Raspberry Pi** • **Wireless Power Transfer** • **WLAN for Microcontrollers**

PC OSCILLOSCOPES

pico® Technology

PicoScope News

PicoScope 6.11 is now released

UPGRADE NOW

www.picotech.com/downloads

PicoScope Best News

ALL FOR FREE

Users of all PicoScope versions old and new can upgrade to the latest software for free, for the life of their product. You can also download the full software to try it for yourself in demo mode.

PicoScope Good News

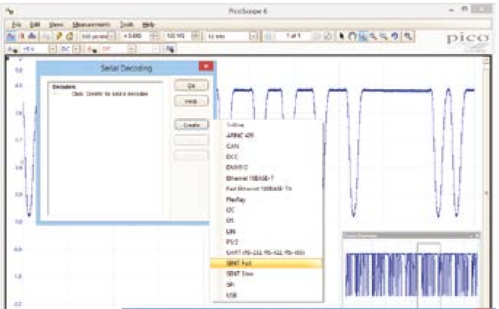
GREAT NEW FEATURES INCLUDING:

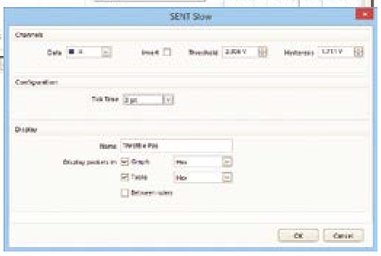
- **16 serial protocols supported as standard:**

Automotive: CAN, FlexRay, LIN, SENT **Avionics:** ARINC 429
Computer: Ethernet 10Base-T, 100Base-TX, PS/2, UART (RS-232, RS-422, RS-485), USB (FS, HS)
Embedded systems: 1-Wire, I2C, I2S, SPI **Lighting:** DMX512 **Hobby:** DCC

- **Touchscreen functions**
- **Mathematical waveform processing tools**
- **Frequency and duty-cycle v time plotting**
- **Advanced waveform mathematics now includes user-configurable filters: High Pass, Low Pass, Band Pass and Band Stop**

SENT focus



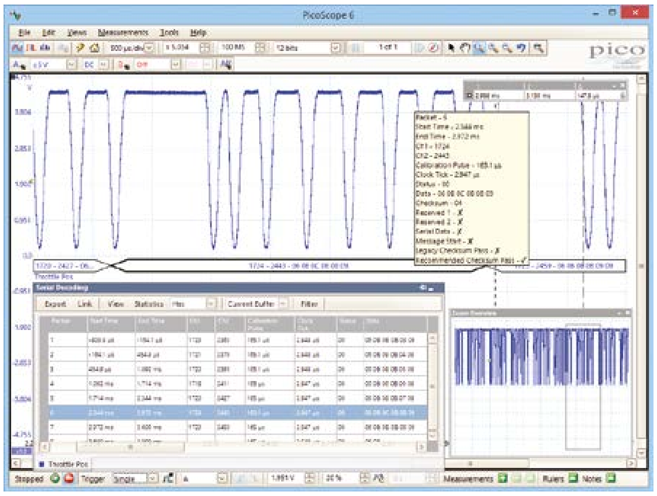


Background

SENT (Single Edge Nibble Transmission) is a serial interface originally designed for automotive applications. Lower cost than other serial protocols, SENT has become popular in high-resolution sensor applications such as throttle position, pressure, mass airflow, and temperature. Data is normally transmitted as two 12-bit data words in a message frame.

The basic unit of time for SENT is a "tick". Each message frame starts with a synchronisation pulse of 56 ticks. Data is transmitted as nibbles - 4 bits of data encoded in the timing of successive falling edges. Nibble time encodes the data in the measured number of tick units. 12 ticks duration = binary 0000 (Hex 0), 13 ticks = binary 0001 (Hex 1), 14 ticks = binary 0010 (Hex 2) and so on up to 27 ticks = 1111 (Hex F).

The message frame ends with a CRC/checksum nibble and optional pause pulse.



SENT decoding with PicoScope

The first step is to acquire the SENT signal of interest using PicoScope. Then select Serial Decoding from the Tools menu.

Click Create and select SENT from the list of available protocols.

In the SENT configuration dialog select the PicoScope Data input channel, Tick Time, Sensor Type, and other parameters as necessary.

Click OK to see the decoded SENT messages in the PicoScope graph display.

If you check the In Table box, PicoScope will display SENT messages in a tabular listing format. Double-click a message in the table to see the same message in the graph display.

Edition 1/2016
Volume 42, No. 469 & 470
January & February 2016

ISSN 1947-3753 (USA / Canada distribution)
ISSN 1757-0875 (UK / ROW distribution)
www.elektor.com
www.elektormagazine.com
www.elektor-labs.com

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

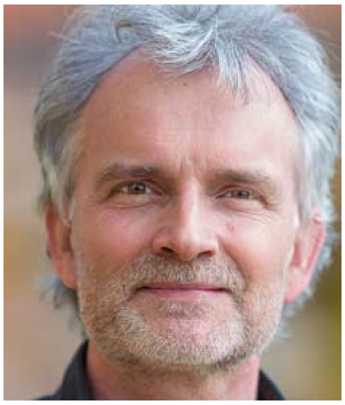
Elektor International Media
78 York Street
London
W1H 1DP
United Kingdom
Phone: (+44) (0)20 7692 8344

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.



Forward-current optimization

At the start of the New Year it's good to look forward with an optimistic attitude in general, still with a realistic eye for anything that occurred in the past year that's "subject to improvement". Along that line, here's a tentative analysis of the forward current I_F.
In 2016 the electromotive force (emf) known as V_{Elektor} and traditionally caused by Elektor Magazine's LEARN – DESIGN – SHARE cycle will rise. We do it by stacking up a few more voltage sources such as the Elektor newsletter, Business TV and Business Magazine, Store, TV, experts, and (breaking news!) our new Aachen, Germany facility. The resulting, higher EMF we hope will increase the peak current $I_{\text{Reader(pk)}}$ through individual loads and even cause the odd spark on a few dusty ones. Back-emf is also appreciated if channeled through our forums and on elektor-labs.com although we have suppression devices in place at critical locations. Admittedly, more output current is only possible if we strive to keep the internal resistance $R_{i\text{(Elektor)}}$ as low as possible. Fortunately Elektor's load $R_{\text{Reader(tot)}}$ is extremely dynamic and even capable of feeding back energy into the circuit — thank you all for that these past 40 years!
As of this edition the chemistry of the rechargeable battery named Elektor Magazine has been enhanced by the addition of a theme article, in this case on LEDs. Our business and media clients have already received the theme allocation for this year as it was printed in Elektor's 2016 Media Kit. In this space I am communicating the allocation for the remainder of 2016 to you, hoping you feel challenged to supply the theme article for an upcoming edition.
March & April: **Microcontroller and Tools**. May & June: **Measurement and Sensors**. July & August: **Power Supplies**. September & October: **IoT and Industry 4.0**. November & December: **Production Technology**.
Don't lag the current $I_{\text{Reader(pk)}}$ by more than a few degrees and contact me as soon as you can.
Enjoy reading this edition and the NFC'ing too!

Jan Buiting, Editor-in-Chief

The Circuit

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

THIS EDITION

Volume 42 – Edition 1/2016

No. 469 & 470 January & February 2016

LEARN DESIGN SHARE

LEARN DESIGN SHARE

eRIC Nitro

powerful low-power radio

ACTIVE CROSSOVER

FOR ACTIVE LOUDSPEAKERS

68

Although active loudspeakers are significantly more complicated than their passive counterparts, they have clear advantages in terms of audio technology and the resulting sound. As shown in this article, there are sound reasons to build your own high-end loudspeakers based on an all-active approach, meaning the cross-over filter has opamps rather than coils and large caps.

116

DESIGN, MAKE, USE

54

eRIC modules are licence-exempt ISM band radios operating in the 433-MHz, 868-MHz or 915-MHz ISM frequency bands. Two versions of the module are available: eRIC4 for 433 MHz, and eRIC9 for 868/915 MHz. They come configured as standard 'serial bridge' modules. Send serial data into the eRIC SDI pin and the module transmits it out on its RF interface. Radio'ing your data — it's a breeze with eRIC Nitro.

LINE AC SWITCH

CONTROLLED BY BLUETOOTH LOW ENERGY

86

Bluetooth Low Energy (BLE) is surprisingly accessible even from a DIY angle. Here we describe the BL600 program and an Android smartphone application with which your AC line switch circuit may be controlled wirelessly. As an unusual detail, the project has a transformerless power supply.

LEARN DESIGN SHARE

NEXT EDITION

Platino Revamped

Platino, Elektor's universal Arduino-compatible AVR-board for 28-and 40-pin ATmega controllers, had a makeover recently and reached version 1.4. In addition to a 3.3-V regulator and better access to the second UART of a 40-pin controller, the component print on the board is also enhanced.

Nixie Clock with Seconds Readout

This remake of the November 2014 Precise Nixie Clock (no. 140013) now also shows seconds on a nixie tube. The addition did require a microcontroller update: the clock is now controlled by a PIC18F4420.

Networked Signal Analyzer

This all-in-one tester comprising a signal generator, oscilloscope functions and a spectrum analyzer exploits the processing power of a PC to process measurement data. The connection and operation with the PC is via Ethernet.

Elektor Magazine edition 2 / 2016 covering March & April
is published on February 23, 2016.
Delivery of printed copies to Gold members 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 PCB Service: Order your own PCBs, both one-offs and larger runs.
www.elektorpcbservice.com

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

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

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

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

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

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

Become a member today!

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

78
Countries

246853
Enthusiastic Members

1031
Experts & Authors

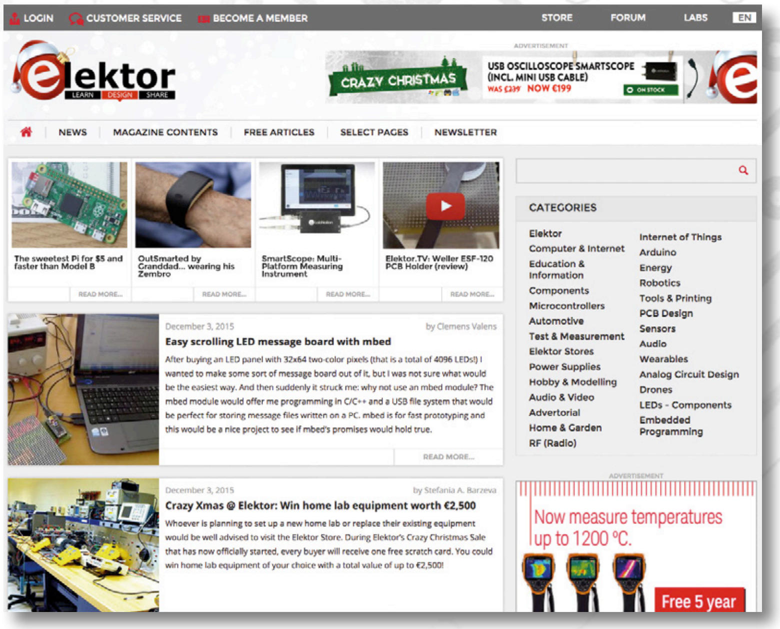
479
Publications

233628
Monthly Visitors

www.elektormagazine.com

A World of Electronics News

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



The sweetest Pi for $5 and faster than Model B

OutSmarted by Granddad... wearing his Zembro

SmartScope: Multi-Platform Measuring Instrument

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

December 3, 2015 by Clemens Valens

Easy scrolling LED message board with mbed

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

December 3, 2015 by Stefania A. Barzeva

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

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

Elektor
Computer & Internet
Education & Information
Components
Microcontrollers
Automotive
Test & Measurement
Elektor Stores
Power Supplies
Hobby & Modelling
Audio & Video
Advertorial
Home & Garden
RF (Radio)

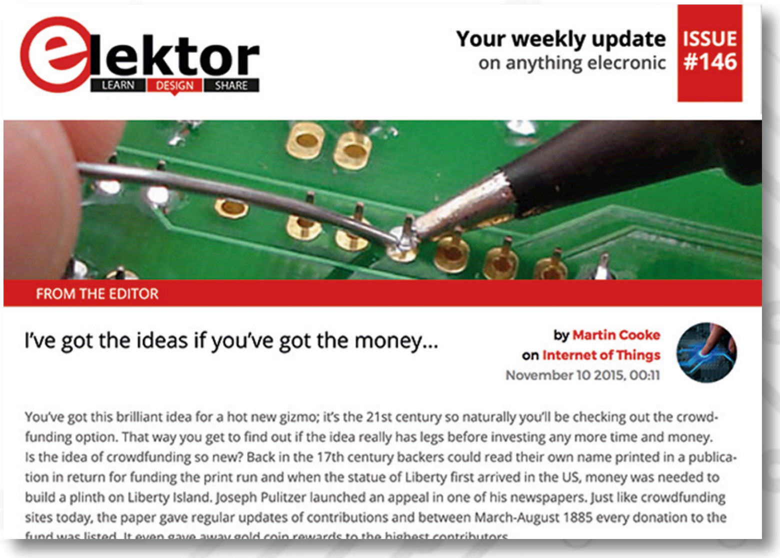
Internet of Things
Arduino
Energy
Robotics
Tools & Printing
PCB Design
Sensors
Audio
Wearables
Analog Circuit Design
Drones
LEDs - Components
Embedded Programming

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



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

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

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

NFC with Elektor and ST

Here's your NFC tag — win one of 500 prizes!

Near Field Communication (NFC) technology for wireless data transfer can be used for contactless payment transactions, identification, or exchanging small amounts of data such as text messages, weblinks and the like.

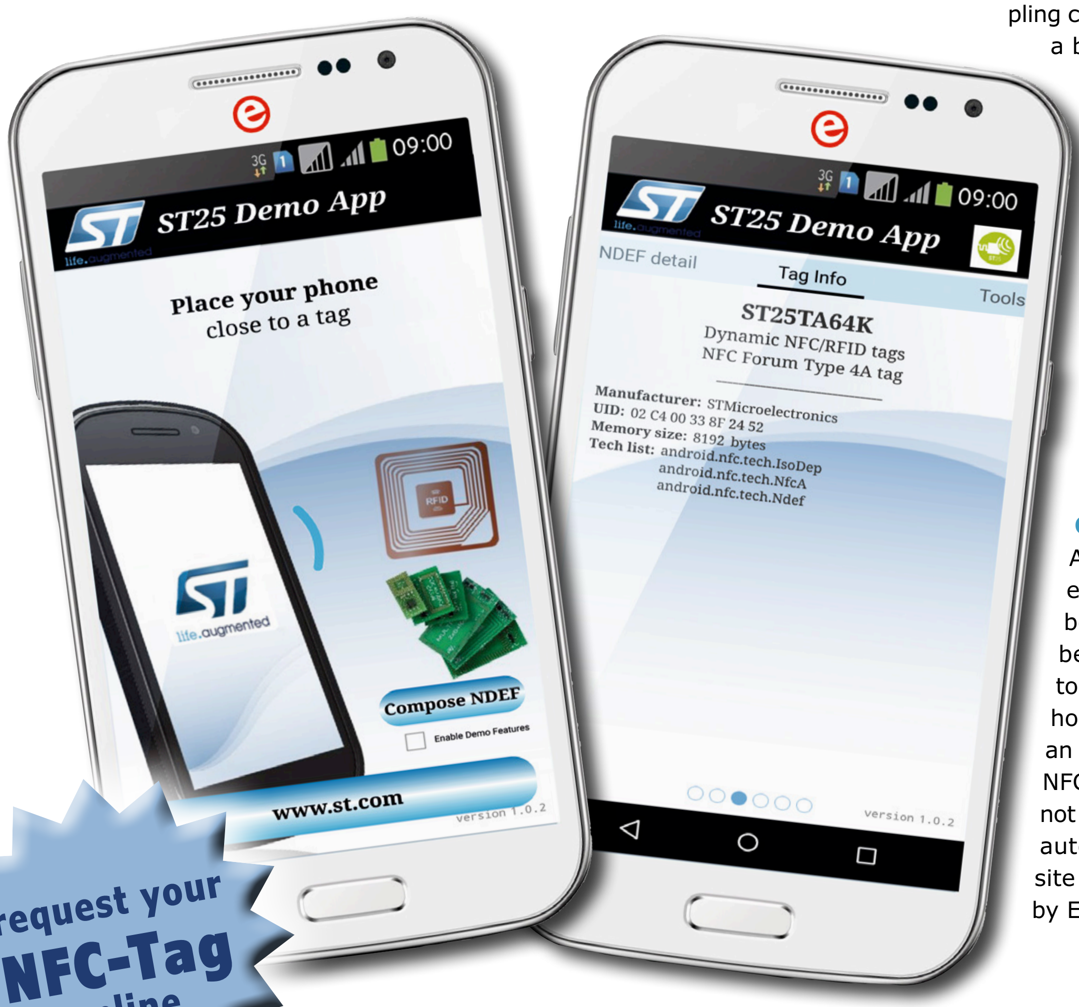
Several well-known semiconductor manufacturers, including STMicroeletronics, have NFC devices in their product lines, including low-power devices that can store data. These can be powered from the electromagnetic field used for wireless communication. That eliminates the need for a battery or other power supply. These ICs can be embedded in compact, sturdy tags in various forms, such as tokens or bank cards.

Free tags for all Gold members

With this edition of Elektor Magazine, our Gold members* receive an NFC tag containing an ST25TA02K chip from STMicroelectronics, who are sponsoring this sampling campaign. With the same format as a bank card, the tag fits perfectly in your wallet and can be used as an electronic business card, among other things. With an NFC-capable smartphone and one of a number of free apps (see the list below), you can personally write your street and e-mail addresses to the tag and read them back. The ST25TA02K has a memory capacity of 256 bytes. To learn more about this IC and NFC technology in general, please see the related article in this edition.

Over 500 prizes to be won

At this point, any practically oriented electronics enthusiast will be itching to play with the tag. But before you do, you should be sure to register for the prize draw. Here's how it works: Hold your tag next to an NFC-capable smartphone with the NFC feature enabled. Even if you do not have an NFC app installed, you will automatically be directed to a website previously set up for this project by Elektor.



Place your phone close to a tag

□ Enable Demo Features

version 1.0.2

ST25TA64K
Dynamic NFC/RFID tags
NFC Forum Type 4A tag

Manufacturer: STMicroelectronics
UID: 02 C4 00 33 8F 24 52
Memory size: 8192 bytes
Tech list: android.nfc.tech.IsoDep
android.nfc.tech.NfcA
android.nfc.tech.Ndef

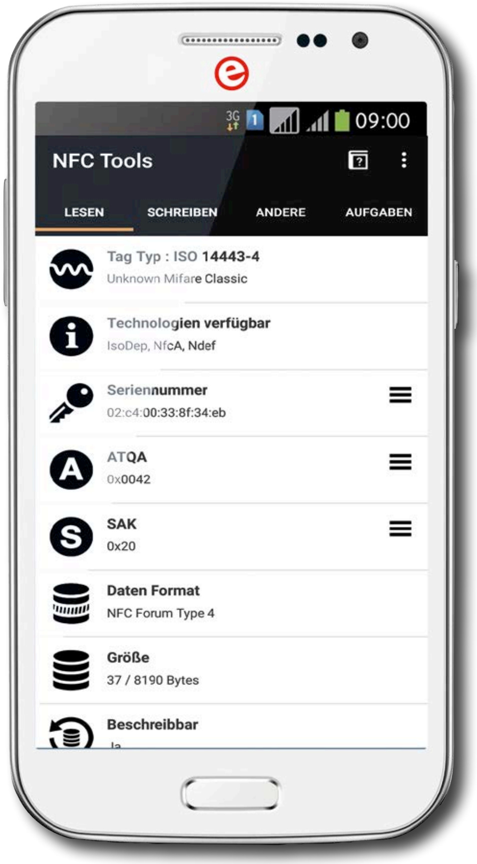
version 1.0.2

request your NFC-Tag online

Elektor Green Members can request a tag online at the following web page: www.elektormagazine.com/stmicro-nfc

This sort of direct jumping to a website is one of the features of NFC, which you can also put to good use for your own personal or PR purposes. A tag (or a bunch of tags) with instant linking to your personal or professional electronics website is bound to be a big hit at the next trade show or similar event, since there aren't many people who have this trick up their sleeve.

Read your number


3G 09:00

NFC Tools

LESEN | SCHREIBEN | ANDERE | AUFGABEN

Tag Typ : ISO 14443-4
Unknown Mifare Classic

Technologien verfügbar
IsoDep, NfcA, Ndef

Seriennummer
02:c4:00:33:8f:34:eb

ATQA
0x0042

SAK
0x20

Daten Format
NFC Forum Type 4

Größe
37 / 8190 Bytes

Beschreibbar
Ja

Now you have a chance of winning one of more than 500 attractive prizes provided by STMicroelectronics:

- **3 Bose Soundlink around-ear wireless headphones**
- **100 STMicroelectronics NFC Discovery Kits** (each with two NFC evaluation boards)
- **400 STMicroelectronics Cloud boards with the ST25TA02K-P chip**

Now it's time to use one of the usual NFC apps to see what other data is lurking on the tag. We tested the card with two popular Android Apps you can get for free from Google Play (see below). Run the app and read the tag by touching the "Read" button (or similar) and then holding the card next to the smartphone again. The example screenshot shows what you see if you use the popular "NFC Tools" app.

What we want is the UID / (card) serial number, which is the unique number written to each tag IC in the factory as an unalterable hex code. That makes it ideal for ID purposes.
Next, navigate to the Elektor NFC website again and register for the prize draw. Along with your street address (important if you are one of the lucky winners), we need the UID of your card. You also have to answer a simple skill-testing question.

Participation is open until **31 March 2016** ultimately. Next, the lucky winners will be drawn and receive notification from Elektor. Of course, we will also keep you informed about the prize draw in our e-newsletter.

We hope you have a lot of fun — and learn a lot — with your NFC tag! ⏮

(150593)

Free Android Apps for UID / card serial number reading:

[1] NFC Tools: https://play.google.com/store/apps/details?id=com.wakdev.wdnfc

[2] ST25 NFC Demo: https://play.google.com/store/apps/details?id=com.st.demo

STMicroelectronics ST25TA NFC Technology

By **Martin Cooke** (UK)

Near Field Communication (NFC) is a standard short-range wireless technology, typically with a range of 10 cm or less. It operates at 13.56 MHz using the ISO/IEC 14443 RF interface at a data rate of 106 Kbit/s. Here we look at an enhanced NFC tag, the ST25TA now being rolled out by STMicroelectronics. Compared to earlier products, the ST25TA has many extra features which should arouse every e-engineer's interest.

The latest smartphones and tablets come with NFC features as standard. NFC-capable Android devices can use Google Wallet for tap-and-pay services but they also free-up the device's NFC interface so that it can also be used for reading tags or conducting peer-to-peer exchanges. Apple iPhones have NFC capability but it's used exclusively for the one-touch Apple Pay feature so it's not possible yet to read a conventional NFC tag with one of these phones. Here we will only be looking at NFC messages between active devices and passive tags. During NFC communication the active device or initiator produces an RF field that interrogates, and at the same time, transfers enough energy to power the NFC tag. The tag turns on and sends out information stored in its memory. The tags are small, low cost and require no external power source so they can take on simple form factors such as stickers, key fobs, business cards, etc.

STMicroelectronics' new ST25TA NFC tags

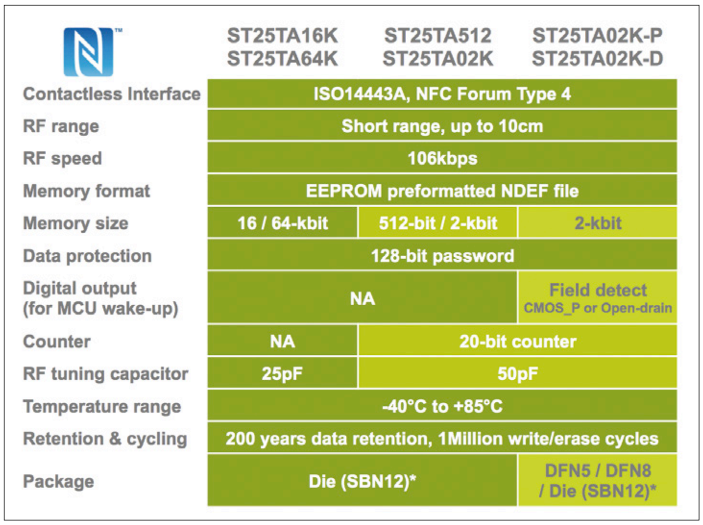
A new family of NFC/RFID tags with ISO/IEC 14443 type A, RF interface supporting NDEF memory and 128-bit password protection has recently been added to STMicroelectronics' existing line of NFC devices. The new ST25TA family of tags pictured in **Figure 1** conforms to the NFC Forum Type-4 Tag RF interface with built-in NDEF (NFC Data Exchange Format) message support. There are currently five different products with embedded EEPROM memory density ranging from 512 bits to 64 Kbits.

Applications

It's becoming increasingly commonplace for smartphone and tablet designs to include NFC capability and there is no doubt that the technology is becoming more widespread. Let's list some areas where we have already seen NFC tags providing cost-effective and reliable solutions.

Wireless pairing

Pairing a device such as a phone or tablet with equipment to allow communication over a Wi-Fi or Bluetooth link can be a fiddly process. An NFC tag is useful here to reduce the necessary hassle to just a simple tap on the tag. To set it up, the equipment's Bluetooth address is first written into the NFC tag. Now when the phone or tablet is tapped on the tag it works as a network credential manager, reading information stored in the tag to set up the network connection automatically. The ST25TA02K-P or ST25TA02K-D NFC tag features an RF signal detector output signal which can be used to wake up a host processor when the tag is activated. The same principle can be used to set up an NFC tag to provide one-tap entry to a secured Wi-Fi network. The very limited range of NFC communications provides some degree of built-in security to the system.



	ST25TA16K ST25TA64K	ST25TA512 ST25TA02K	ST25TA02K-P ST25TA02K-D
Contactless Interface	ISO14443A, NFC Forum Type 4		
RF range	Short range, up to 10cm		
RF speed	106kbps		
Memory format	EEPROM preformatted NDEF file		
Memory size	16 / 64-kbit	512-bit / 2-kbit	2-kbit
Data protection	128-bit password		
Digital output (for MCU wake-up)	NA		Field detect CMOS_P or Open-drain
Counter	NA	20-bit counter	
RF tuning capacitor	25pF	50pF	
Temperature range	-40°C to +85°C		
Retention & cycling	200 years data retention, 1Million write/erase cycles		
Package	Die (SBN12)*		DFN5 / DFN8 / Die (SBN12)*

Figure 1. The ST25TA family at a glance.

Product authentication

High-end goods can often be targeted by counterfeiters. The use of tags, scanned by a customer's NFC-enabled phone can give a guarantee of product authenticity and provides an opportunity for the manufacturer to establish a link with the customer to promote upcoming products or offers.

Business cards

Business cards are available with integrated tags. These allow you to store your business contact information and links so that when the card is tapped against a NFC-capable smartphone it can direct the browser to a web address, online shop, video or downloadable app.

NFC posters

NFC tags embedded in posters work in much the same way that a printed QR code which can be used to direct your browser to a URL. In order to read QR codes you first need to install and launch an app. By contrast an NFC-capable device just needs a tap over the tag symbol on the poster and it will direct the browser to the URL of a video, show a list of nearby outlets appropriate to the poster content, pass on special offers, coupons or information of other promotions.

Industrial identification

Tags don’t need an independent power source and they can be fully encapsulated making them more durable and robust for use in industrial environments. NFC-capable smartphones and tablets with their built-in connectivity options, GPS information and relatively low-cost, are a good choice as an NFC user-interface for a large number of industrial applications. App development is also a relatively mature technology with a wide range of design environments and established expertise in the field. Some industrial applications currently benefitting from the use of NFC tags include maintenance, repair, servicing, component identification, production control, inventory systems, equipment hire management and storage location labelling.

Gaming

One interesting use of an NFC tag is in Google’s low-cost cardboard VR headset where a built-in tag triggers the Android phone fitted to the headset to download or launch the appropriate 3D app. Tags can also be embedded into easily portable figurines associated with video games. You just need to take the figurine with you when you play at a friend’s house. NFC readers in the games console can read the tags to enhance player interaction with the game. Based on data stored in the tag, players can be allowed to take part in the action, unlock special features and also save user data to the tag such as player experience level, points and personal setup preferences.

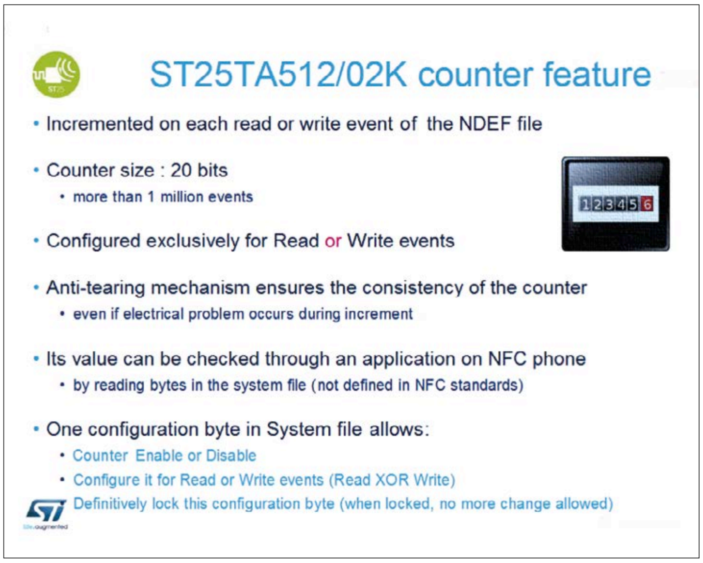
Common features of the ST25TA family

Internal memory of the ST25TA family of NFC tags is organized to provide space for three files. NFC applications use data stored in the NDEF region of the tag’s internal memory. The memory also stores the Capability Container file (CC file) and a STMicroelectronics proprietary System file. The quoted memory size for the tag refers to the space available for the NDEF region. The CC file and the System file are stored elsewhere and can be read and modified using the STMicroelectronics ST25 Demo App for Android. For security purposes NFC communication can only take place with an Android platform when its screen is on and unlocked.

Read and write operations to the NDEF file can be protected by 128-bit passwords. One password can be used to protect against unauthorized read operations and another for write operations. Two bytes in the CC file contain information on the locked or unlocked status. The tag data can be permanently locked so that read or write access is permanently denied.

The Low Density tags

The ST25TA512 and ST25TA02K are both classed as Low Density tags and only differ by the size of their memory, providing 512-bit and 2-Kbit EEPROM capacity respectively. The ST25TA02K-P and -D are part of this Low Density category and feature a digital output pin to wake up an external MCU, BT or Wi-Fi chip. This chip is also the only version available as a 5- or 8-lead UFDFPN package as well as an SBN 12 sawn-and-bumped wafer which all the other chips are supplied in.
These tags also feature an internal 20-bit counter, (**Figure 2**) configurable to keep track of the number of times the NDEF file in the tag has been read or written to (useful to check how often a tag in a public place has been read). The counter value can be accessed via an NFC reader device by reading its value



ST25TA512/02K counter feature

- Incremented on each read or write event of the NDEF file
- Counter size : 20 bits
  - more than 1 million events
- Configured exclusively for Read or Write events
- Anti-tearing mechanism ensures the consistency of the counter
  - even if electrical problem occurs during increment
- Its value can be checked through an application on NFC phone
  - by reading bytes in the system file (not defined in NFC standards)
- One configuration byte in System file allows:
  - Counter Enable or Disable
  - Configure it for Read or Write events (Read XOR Write)
  - Definitively lock this configuration byte (when locked, no more change allowed)

Figure 2. Low Density tags include a counter.

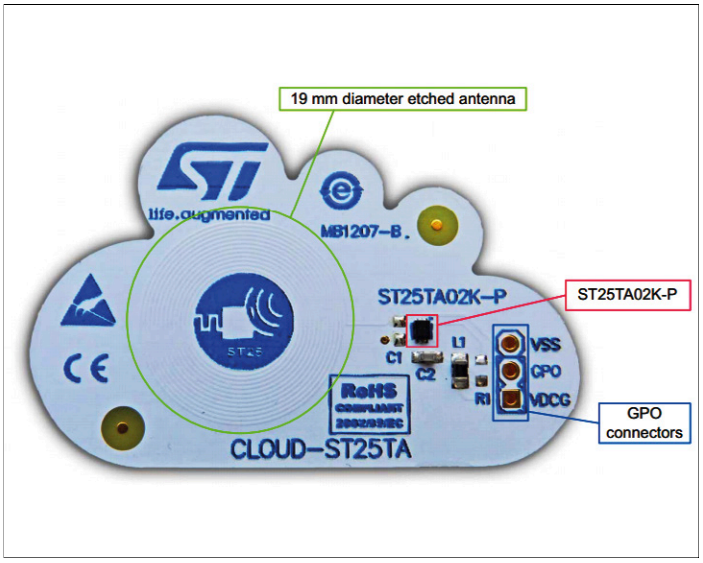


Figure 3. The CLOUD-ST25TA Evaluation board.

The RF signal detector on the ST25TA02K-P NFC tag can wake up a host processor

stored at a location in the system file. Disabling the counter resets the count value to zero. An anti-tearing mechanism ensures that the counter value cannot be corrupted if the RF field is removed during counter update.

The Low Density tags feature an in-built antenna capacitance of 50 pF which allows the use of a smaller antenna coil with a lower inductance. This characteristic ensures tag compatibility with pre-existing 50-pF antenna designs.

To list just a few applications of Low Density tags:
- NFC tag
- NFC token
- NFC key fob
- Smart poster
- Gaming figurines
- NFC business card (name card; vcard)

The High Density tags

The ST25TA16K and ST25TA64K are both High Density tags and offer 16-kbit and 64-kbit EEPROM storage respectively. Their increased capacity makes them suitable for storing larger files such as photo ID applications. These High Density tags do not have an internal counter function implemented to keep track of read/write operations. The internal antenna capacitance is 25 pF, half the value of that in the Low Density tags which means that a matching antenna coil will need to have a higher value of inductance. High density tag applications include:

- Smart posters
- Gaming figurines
- NFC tag, token or key fob
- NFC business card (vcard with ID photo or digital content)

The CLOUD-ST25TA evaluation board

The low cost CLOUD-ST25TA evaluation board shown in **Figure 3** consists of a small cloud-shaped PCB on which is mounted an ST25TA02K-P NFC tag in a UFDFPN5 ECOPACK2 package. It is an example of how the NFC chip can be integrated into an end product such as a smart card or tag. The antenna takes the form of a 12-turn printed coil with a maximum diameter of around 18 mm. This NFC chip used on the evaluation board has a GPO active-high output which is configured by default to generate a rising edge when the RF field produced by a NFC RF host comes within range.

The GPO output can be configured to indicate:

- Session open: An RF session is ongoing;
- MIP (NDEF Message update In Progress): The RF host is writing a non-zero length NDEF message. Indicates that the RF host is changing the NDEF message as defined in the NFC forum;
- WIP (Write In Progress): The ST25TA02K-P is executing a write operation;
- INT (Interrupt): The RF host can force the ST25TA02K-P to pulse the GPO pin;
- State mode: The RF host can control the GPO output state during the session.
- RF busy: An RF host is communicating with the ST25TA02K-P;
- Field Detection: Received RF field strength is sufficient to support RF communication with the ST25TA02K-P.

The VSS connection on the evaluation board connects to ground and VDCG connects to the positive VCC supply of the local circuit which can be in the range of 1.65 V to 5 V. VDCG only sets the GPO signal's High level, reducing power consumption, and the GPO interrupt signal is active High (rising edge) which is uniqiue in the external component market . The chip contains a 2-Kbit EEPROM which supports the standard NFC Data Exchange Format (NDEF) files. The memory is powered by energy from the external RF field and can only be accessed via the RF interface. The ST25TA02K-P is compatible with the NFC Forum Type 4 tag specifications and supports all the corresponding commands.

The ST25 App

STMicroelectronics have developed an Android app called the ST25 demo app [1]. This allows you to read the contents of the tag's files stored in memory (executable STSW-ST25001 and source code STSW-ST25002) and to write an NDEF file to the tag (**Figure 4**) to generate the type of action you wish the tag to perform when it is tapped by an NFC smartphone or tablet. The app allows you to read the files stored in STMicroelectronics tags. **Figure 5** shows the information available on the NDEF file; if you are any good at translating hex to ASCII you can read the short message payload I wrote on my card. Further scrolling gets you to the TAG info showing the tag's 14-character (56-bit) UID and its memory size.
The Tools page allows you to set bits in the tag to lock read or write access. Further options applicable to the ST CLOUD-ST25TA board only, allow you to set the GPO output level, erase data stored in the NDEF, and configure the read/write counter. More scrolling will show the contents of the CC and SYSTEM files. Besides the Android ST25 demo app [1] you can also download its source code, the Gerber files for the CLOUD-ST25TA Evaluation board, and the complete bill of materials.

(150472)

Web Link

[1] www.st.com/st25apk

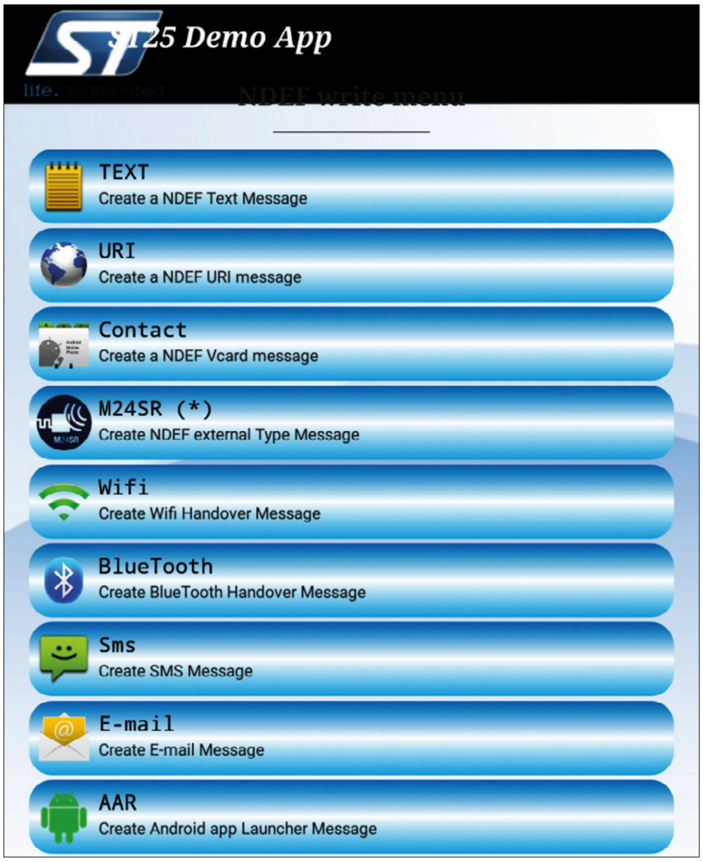
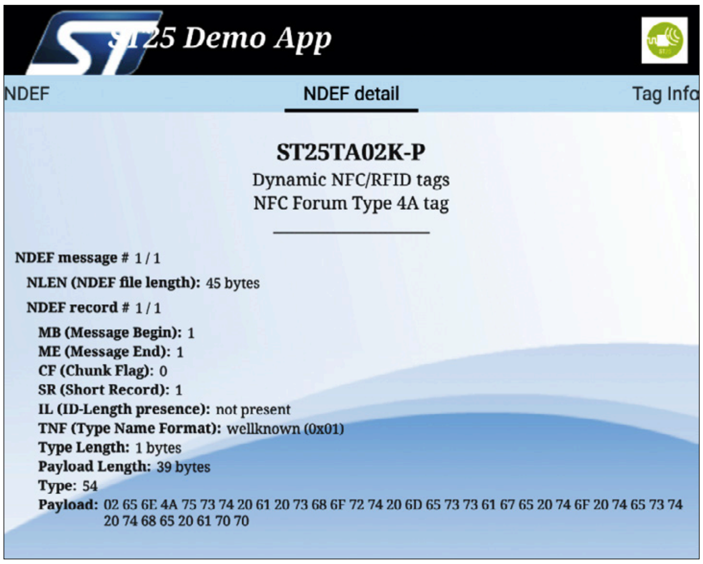

ST25 Demo App
life.augmented
NDEF write menu
TEXT
Create a NDEF Text Message
URI
Create a NDEF URI message
Contact
Create a NDEF Vcard message
M24SR (*)
Create NDEF external Type Message
Wifi
Create Wifi Handover Message
BlueTooth
Create BlueTooth Handover Message
Sms
Create SMS Message
E-mail
Create E-mail Message
AAR
Create Android app Launcher Message

Figure 4. Compose your NDEF message.

Try decoding the ASCII message in Figure 5



ST25TA02K-P
Dynamic NFC/RFID tags
NFC Forum Type 4A tag

NDEF message # 1 / 1
NLEN (NDEF file length): 45 bytes
NDEF record # 1 / 1
MB (Message Begin): 1
ME (Message End): 1
CF (Chunk Flag): 0
SR (Short Record): 1
IL (ID-Length presence): not present
TNF (Type Name Format): wellknown (0x01)
Type Length: 1 bytes
Payload Length: 39 bytes
Type: 54
Payload: 02 65 6E 4A 75 73 74 20 61 20 73 68 6F 72 74 20 6D 65 73 73 61 67 65 20 74 6F 20 74 65 73 74 20 74 68 65 20 61 70 70

Figure 5. Reading the NDEF contents (can you read the message?)

Welcome to the LEARN section

By Jens Nickel

I'm sure you will agree that the subject of electronics is such a wide palette that no one person can hope to stay up to date with all the latest developments in every field.

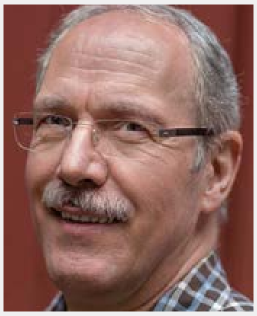
For the last couple of months there have been seven of us here at Elektor, producing and editing content for your enjoyment and enlightenment. Not just in the form of classic printed copy and books but also online including videos. It seemed a natural development that we should separate out, areas of interest between the seven of us. Our aim is to continue the battle, ever onward, searching through the flood of press releases, sifting information, keeping in touch with companies, visiting trade shows, videoing, and chatting with our readers and more.

We have split the whole discipline up into categories and assigned them to the seven of us to coincide with our own intertests and expertise.

So you've built something you're proud of or got a suggestion for an article you would like to see built or a book topic? Maybe you have special knowledge or insight into one of the themes? Don't keep it to yourself; we are always pleased to hear from you. You can reach us through Jan, at email editor@elektor.com.

So this is us and these are our main areas of interest:

Clemens Valens
Microcontrollers & Boards,
Solder Technology & PCBs



Harry Baggen
Test & Measurement,
Simulation & Circuit Design,
Audio

Thijs Beckers
Automotive,
LEDs & Displays,
Computer & Media Players

Jan Buiting
Power Supplies,
Power Electronics,
RF, Retro Electronics

Rolf Hähle
(Home) Automation,
Industrial Electronics

Denis Meyer
Green Electronics, Music,
Hobby & Modelling

Jens Nickel
IoT, Networked Systems,
Software

Basics

From 2016 — so beginning with this edition — we strive for each issue to focus on a particular theme. We all use more lighting this time of year so the winter double issue will be themed on LEDs and Lighting. In this the 'LEARN' section of the magazine you will find the first foundation article in this series. Our freelance contributor Dr. Thomas Scherer with his extensive background knowledge will take you through the development of LEDs and bring you bang up to date with the latest advances in LED technology and OLED lighting. ◄

Waveguide

Peculiar Parts, the series

1

By **Neil Gruending** (Canada)

Coaxial cable is usually the best way to connect radio equipment together but its practical upper limit is about 3 GHz before the cable and connector losses become excessive. There are some specialized cables that are rated for the super high frequency (SHF) band of 3 to 30 GHz but it's much more common to use a length of waveguide instead. Let's take a look at waveguide and see how a hollow metal tube can be a better conductor than cable at those frequencies.

Microwave radio sources were first developed in the 1920's. Once it became apparent that coaxial cable had excessive power losses, researchers started to look for better transmission methods and materials. Round hollow tubes were promising as waveguides but it turned out that it was easier to model the propagation mode and cutoff frequency for rectangular waveguide instead. One of the first real applications for these new waveguides was in World War II radar systems as the "plumbing" between the klystron and the antenna. Waveguide is still available today except that it is usually manufactured as a standardized generic piece with large flanges that are used to connect waveguide pieces together like in the circulator in **Figure 1**.

RF energy propagates through a waveguide using either TE (transverse electric) mode, TM (transverse magnetic) mode or a combination of the two. For these propagation modes the electric and magnetic fields are perpendicular to each other. One field will be transverse to the wave's propagation direction and the other will create a longitudinal loop that propels the RF energy through the waveguide.

The optimum operating point of waveguide is the propagation mode with the lowest cutoff frequency — this is called the dominant mode. If the frequency goes below the cutoff point then it won't propagate and if the frequency is too high then it's possible to have different propagation modes active at the same time which isn't very efficient.

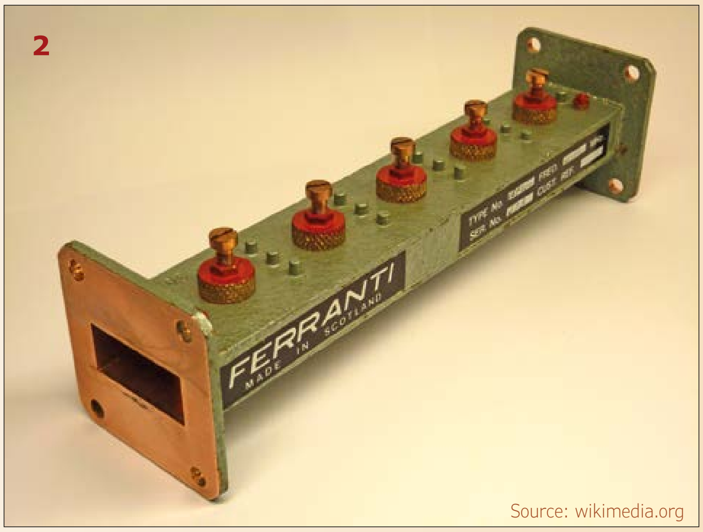
A typical waveguide is made from a rectangular metal tube that has been plated on the inside with copper, silver or gold. The RF signal is mostly propagated in the empty space inside of the tube so the transmissions losses are very low and in practice, limited to any junctions and adapters in the line. But some of the RF waves are reflected off the side and a small electric current will form on the inside edges which can reduce the transmission efficiency. That's why the inside is plated with a high conductivity metal.

With the SHF radio signal contained in the waveguide, the metal tube is perfect for building very low loss pass-band and notch filters "plumber's style" as shown in **Figure 2**. This is a piece of WG16 waveguide covering the famous 10-GHz (3-cm) band.

It's pretty easy to experiment with waveguides at home if you have a frequency generator that can output a high enough frequency — say, 3 GHz or so. All you have to do is couple the signal into a metal tube or can with a coaxial cable and use another cable as a sensor to measure the output power to see how well it's operating. Have fun! ◀

(150540)

2



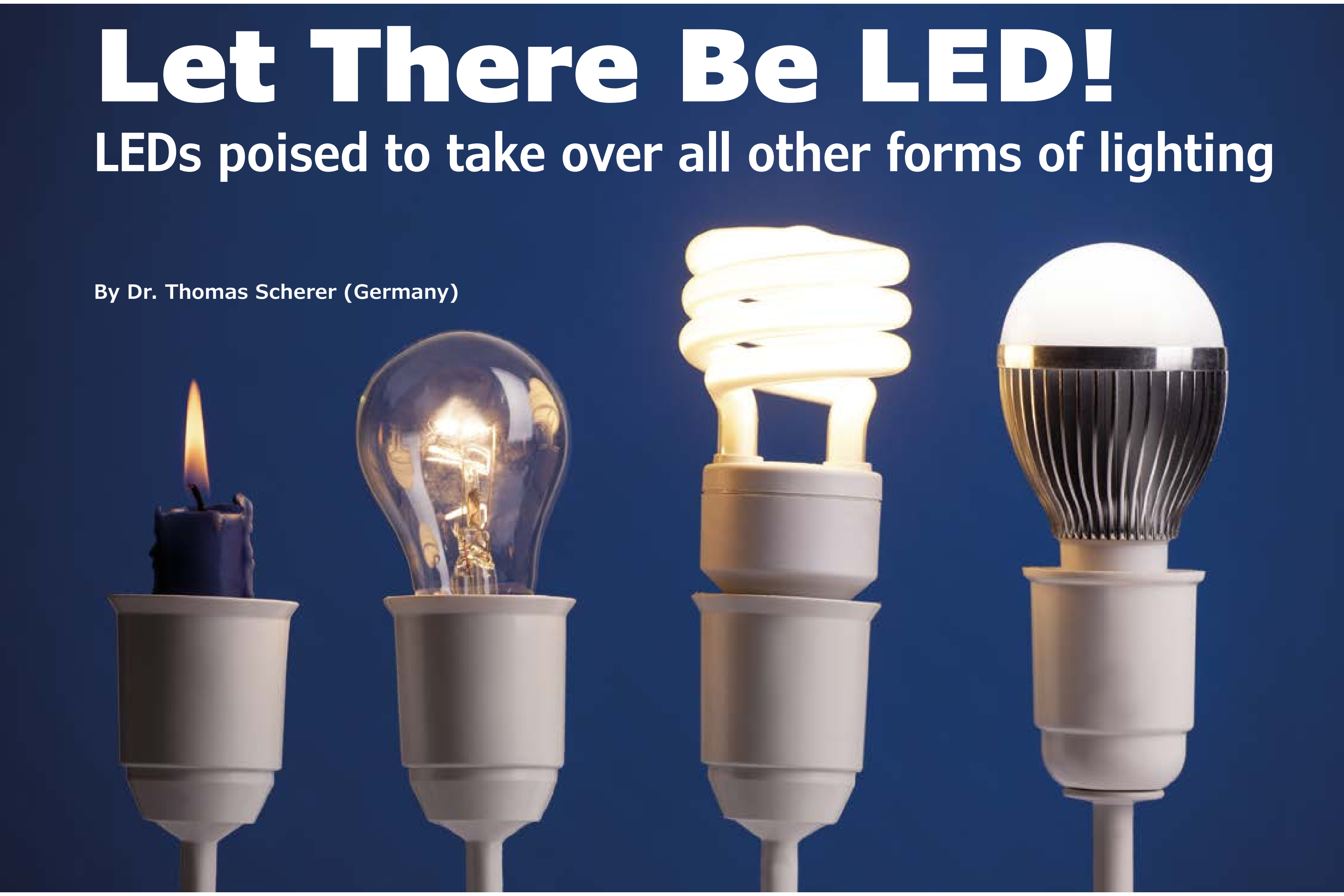
Source: wikimedia.org

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

Let There Be LED!

LEDs poised to take over all other forms of lighting

By Dr. Thomas Scherer (Germany)



It took a while for the change to come, but now the momentum of the change is enormous. LEDs dominate the selection of lighting fixtures in home improvement centers now, and everywhere people know that LED lamps mean lower energy consumption. Even municipal politicians have apparently become enlightened – not only in big cities but also in small towns and villages, streetlights are increasingly being converted to semiconductor lighting. Here we tell you why this is happening.

LEDs actually have an amazingly long history. Of course, discovery of the diode effect in solid-state materials was the first prerequisite. This occurred back in 1874 when the German physicist Ferdinand Braun invented the crystal detector, which was actually a Schottky diode. In 1907 the British radio pioneer Henry Joseph Round discovered that these diode junctions emit light. He even described the occurrence of different colors from different materials in various arrangements at different voltages.

The first LEDs

Things really started moving a number of years later when semiconductors replaced tubes in amplification stages and reliable processes for industrial semiconductor fabrication were developed. R&D efforts were strongly focused on investigating a wide variety of properties of diverse semiconductor materials. The first working LED, which was made from gallium arsenide and emitted infrared (IR) light, was developed in 1955 by Rubin Braunstein at RCA. Progress was rapid in the 1960s: James Baird applied for a patent [1] on the IR LED (see **Figure 1**), and Texas Instruments started series production a short while later in 1962. The world's first commercial LED was housed in a TO18-type metal package with a lens, operated at a current of 100 mA, and emitted 1.5 mW of infrared light at a wavelength of 940 nm. Its efficiency was 1.1%, which was actually not so bad.

The first LED that emitted visible red light was also developed in 1962 by Nick Holonyak at General Electric. Ten years later, his student M. George Craford built the first yellow LED. He also boosted the efficiency of red LEDs by a factor of ten. The prices of red LEDs dropped quickly, and they were initially used as replacements for incandescent and gas discharge indicator lamps. If you can remember the 1970s, you can surely recall the proud faces of the owners of the first digital watches with 7-segment LED displays, which drew so much current that they only lit up briefly when the user pressed a button. LEDs were also indispensable for the first pocket calculators. Incidentally, RCA had also invented blue LEDs by that time. With red, yellow, green and blue LEDs, the full spectrum was covered. LEDs quickly became established as display devices due to their easy drive characteristics, low power consumption and high reliability. However, their efficiency was much too low for lighting applications.

Bright LEDs

It quickly became clear that LED-based lamps would have many advantages compared to conventional incandescent and gas-discharge lamps. This was primarily due to their reliability and long life, which are particularly important for maintenance costs when lamp replacement involves a lot of effort and expense. In comparison to the more efficient fluorescent lamps, LEDs scored better in terms of high switching cycle capacity and fast start-up. However, they needed to produce more light.

The first bright blue LEDs were developed by the American engineer Shuji Nakamura using gallium nitride material, and series production by the Japanese firm Nichia started in 1993. Together with the Japanese scientists Samu Akasaki and Hiroshi Amanom, he received the Nobel Prize for Physics in 2014 in recognition of the growing importance of efficient LED lighting. This is also clearly shown by sales figures. The global volume of the LED market is about USD 16 billion in 2015, with pure lighting technology accounting for roughly half this amount.

In 2006, Nichia achieved an amazing efficiency of 22% for blue LEDs in the lab environment, compared to a meagre efficiency of 3 to 5 percent with incandescent lamps. The corresponding luminous efficacy was 150 lm/W, which is a practically useful level. Since then Cree has already reached the 300 lm/W level. Although they only managed this in the lab, it's still an impressive figure. The physical limit of luminous efficacy for white light with a color temperature of 6,600 K is 350 lm/W.

White LEDs

The next question was how to generate white light with semiconductor devices. As you know, white light is a mixture of various colors that is perceived as white by our eyes. If you were paying attention in physics class, you surely remember the wonderful rainbow spectrum you saw when a beam of sunlight passed through a prism. As thermal radiant emitters, incandescent lamps also generate a similar nice spectrum, but with a yellowish tinge. This is because the blue light at the upper end of the spectrum emitted by the lamp filament at a temperature slightly above 2,000 K is weaker than in daylight. A simple way to make a white LED is to mount a set of red, green and blue LEDs on a substrate in order to produce white light by additive color mixing, just like a television set. These RGB LEDs have the additional advantage that the color of the light can be varied by adjusting the drive levels of the three LED chips. That's exactly what the Ostar device shown in **Figure 2** is designed for. This four-chip device enables precise mood lighting settings, particularly for stage lighting.

However, multi-chip designs of this sort are too complicated and expensive for normal applications. The solution came from a genial invention in 1995 by Jürgen Schneider at the Fraunhofer Institute for Applied Solid-State Physics. Shortly before he retired, he managed to develop a one-chip white LED with a phosphor layer for color conversion. In concrete terms, that means a blue LED coated with a phosphor material. It works on the same principle as a fluorescent lamp: part of the blue light gets converted to longer-wavelength yellow light. This effectively amounts to additive color mixing, as illustrated in **Figure 3**. There you can see the resulting light spectrum at the bottom, with the peak of the original blue light and the relatively broad spectral band of the phosphor. The combination of this broad band (instead of pure yellow light) and suitable spectral mixing yields the desired white light effect with only two colors. **Figure 4** shows a modern high-power LED from Lumileds, with the yellowish phosphor layer completely covering the LED chip. Low-cost 5 mm indicator LEDs selling for just 10 cents each have a similar structure.

Economical non-RGB white LEDs have become the standard. They are used in virtually all LED lamps for household use (**Figure 5**), street lamps (**Figure 6**),

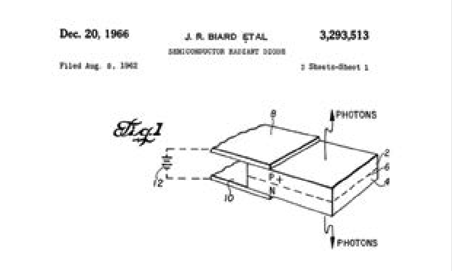


Figure 1. Sketch from the patent for the first LED, described as a "semiconductor radiant diode".

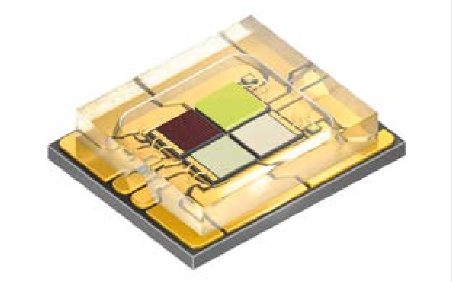


Figure 2. A multi-chip high power LED rated at 30 watts: Ostar Stage from Osram. It has red, green, blue and white LED chips on board and is primarily intended to be used in spotlights for stage lighting.

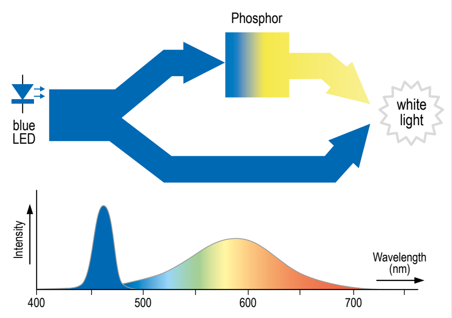


Figure 3. Working principle and spectrum of a one-chip white LED with a phosphor layer for light conversion and mixing.

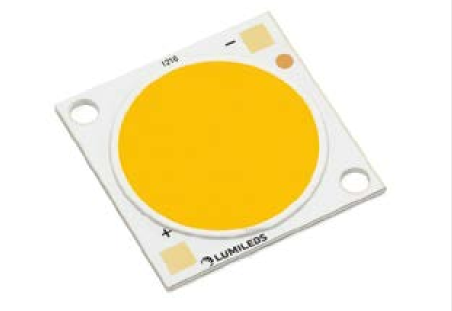


Figure 4. A modern ultra high power LED: the Luxeon CoB 1216. It consists of many blue LEDs mounted on a ceramic substrate and wired in series, all covered by a phosphor layer. This LED produces and astonishing luminous flux of 15,000 lumen.

Efficiency and related matters

Efficiency in terms of lumens per watt (the luminous efficacy) is doubtless the most important consideration for lighting purposes, because it not only affects life cycle costs but also has economic and environmental consequences. Other things that affect the cost are the lifetime and specific factors such as maintenance. For example, energy-efficient lamps are extremely unsuitable for lighting stairwells and the like because they have relatively low switching cycle capacity. LEDs and OLEDs are ideal for this application area. Light quality is also a crucial consideration with regard to intended use. For instance, sodium vapor lamps are very efficient and have been around for a long time, but the very narrow spectrum of the yellow light makes them unsuitable for residential or workplace lighting. Disposal considerations also play a role – for instance, the mercury content of fluorescent lamps means that they must be treated as hazardous waste.

Due to drastically reduced prices, improved efficiency, the prohibition of incandescent lamps (no longer allowed in the EU), and the incredible diversity of types and models, LED lamps already dominate the lighting sector. The market share of LEDs is expected to be as high as 75% by 2020. That's hardly surprising when you consider that an ordinary 10-W LED lamp with a purchase price of about €5 and a lifetime of 30,000 operating hours has a life cycle cost of roughly €80. At present electricity rates, you would pay six times as much for thirty equivalent 60-W incandescent lamps and the power they consume. On top of that come the much higher carbon dioxide emissions.

It looks like other types of lamps are only needed now for special applications. However, even for incandescent lamps there are some exceptions. For example, they are entirely suitable for use in high temperature environments such as ovens. OLEDs certainly have a large potential. With further improvements in efficiency, lifetime and cost, they will be ideal for the residential and workplace sectors where glare-free luminaires with unusual designs are needed.

Table 1

Lamp type	Application	Light quality	Lifetime (kh)	Luminous efficacy (lm/W)	Power (W)
Incandescent lamp	General	Good / yellowish	1	10 to 16	up to 1,000
Halogen lamp	General	Very good	2	12 to 18	up to 2,000
Energy-efficient lamp	Household	Limited / good	5 to 10	50 to 60	3 to 25
Fluorescent lamp	Buildings	Limited / good	5 to 15	60 to 100	10 to 60
LED (in lab)	—	—	up to 100	>300	<1 to 50
LED lamp	General	Good	30	80 to 120	2 to 15
RGB LED	Special	Very good	20 to 50	50 to 150	<1 to 30
OLED	General	Good	>10	40 to 60	<20
Sodium vapor	Street lighting	Yellow / 590 nm	30	150	35 to 1,000
Mercury vapor	Buildings	Bad	>10	40 to 60	50 to 1,000

Light measurement units

The situation with light is just as bad as with magnetism, since there are many units in common use. The most important units are described below.

Luminous flux: Used to compare different forms of lighting. The unit is lumen (lm). The luminous flux indicates the amount of light radiated per unit time. With bright green light (550 nm), a radiated power of 1 W corresponds to 683 lm.

Luminous intensity: Mainly used in connection with small LEDs in the milliwatt range for signaling purposes, etc. The unit is candela (cd). Radiation with a luminous flux of 1 lm per steradian (1 sr is the solid angle of an area of 1 m^2 on the surface of a sphere with a radius of 1 m) corresponds to a luminous intensity of 1 cd.

Illuminance: Generally corresponds to what is regarded as the brightness of an illuminated surface. The unit is lux (lx). When an area of 1 m^2 is lit by a luminous flux of 1 lm, the illuminance is 1 lx.

Luminous efficacy: A measure of the efficiency of the conversion of electrical energy into light, which makes it a good measure for comparison purposes. The unit is lumen per watt (lm/W). The physical limit on the maximum energy efficiency of white LEDs with a not especially uniform spectral distribution is approximately 50%, corresponding to 350 lm/W.

Color rendering index: This unitless quantity (symbol R_a) is the percentage value of correct color rendering (a component of light quality) compared to sunlight, as measured with eight test colors. Sunlight therefore has an R_a value of 100. Only halogen lamps can reach this level. White LEDs achieve values in the range of 75 to 95.

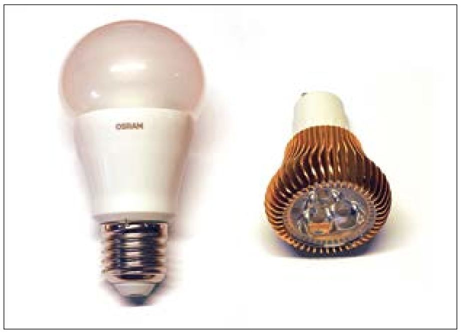


Figure 5. LED lamps are available in every imaginable form with a wide range of color temperatures and versions. Two types are shown here: a 10 W lamp with a conventional form factor and E27 screw base, and a small 4 W spotlight with three LEDs and a GU10 base.

Figure 6. Night view of newly installed LED street lamps in the author's home town.

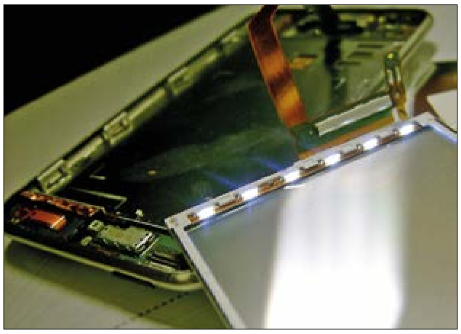


Figure 7. Edge-lit LED backlight from an Apple iPod (photo: ReTheCat, Wikimedia.org)

> The prime advantage of OLEDs is the cheap raw material together with the economical methods of manufacturing.

backlights for computer monitors, mobile devices (**Figure 7**) and television screens, and more recently even in automobile headlamps (**Figure 8**). They are available with power ratings from 50 mW with tiny SMD-0201 packages to several hundred watts with saucer-sized modules. They are the fastest growing segment of the LED market, with enormous growth rates.

OLEDs

Organic LEDs (OLEDs) have a considerably shorter history than conventional inorganic LEDs. In the early 1950s the French physicist André Bernanose observed electroluminescence in organic materials. In 1965 the American firm Multi Dow Chemical patented electroluminescent cells that operated with high AC voltages at around 1 kV and several hundred hertz. In 1975 the British physicist Roger Partridge discovered light-emitting polymers, where were a forerunner of modern OLEDs. The next step was the fabrication of polymer semiconductors, which was achieved by Ching Tang and Steven Van Slyke at Kodak in 1987. However, OLEDs stayed in the research lab for a long time, until devices more suitable for commercial use were developed around 2004. Many current OLED patents are held by Universal Display Corporation. The display sector is dominated by LG, a large Korean manufacturer of television sets and other household appliances.

Modern cells have a two-layer structure as shown in **Figure 9**. First a thin anode layer made from a transparent electrically conductive material (e.g. indium tin oxide, ITO) is deposited on a plastic or glass substrate. A polymer hole transport layer (HTL) is then deposited on the anode layer, followed by the emissive layer (EL) containing the luminescent substances. Finally, a metallic cathode layer is usually applied. When a current flows through the cell, electrons and holes move toward each other and recombine to form excitons. The excitons cause photon emission either directly or indirectly, depending on the selected luminescent substances. High-efficiency OLEDs use metal-organic complexes for this purpose.

The main advantages of OLEDs are the low cost of the materials and economical fabrication, since the layers are not as thin as with inorganic semiconductors and therefore do not require such strict cleanroom conditions. Polymer layers can even be made by printing. Another advantage is the flexibility of the layers. Flexible OLEDs can be produced on flexible substrates, making flexible displays possible. Color OLEDs with very distinct colors, and even white OLEDs, can be made with various luminescent substances. Modern OLEDs have forward voltage of 3 to 6 V, depending on the structure. However, their limited luminous efficacy is still a disadvantage and they cannot compete with inorganic LEDs in this regard. This also applies to the lifetime, which is presently somewhat more than 10,000 hours. However, progress is being made.

Their other properties and easy planar fabrication make OLEDs good candi-

Figure 8. Mercedes E-Class LED headlamp (photo: A. Bailey).

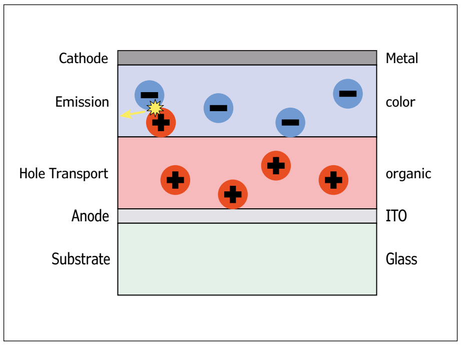


Figure 9. Typical layer structure of modern OLEDs.

Things to watch out for with LED lamps

Thanks to enormous growth in the number and variety of LED lamps, it's hard to tell the sheep from the goats now. However, there are some rules of thumb and criteria that can help you out.

- **Dimmers:** Before you buy a LED lamp, you should consider whether or not you want to use it with a dimmer, because LED lamps that can work with dimmers are a good deal more expensive. If you do want to use a dimmer, it is essential to look for lamps that are explicitly marked on the package as suitable for use with dimmers. Otherwise the lamp will flicker a lot, and in the worst case it will quickly turn into electronic scrap. Although the dimming characteristics of LED lamps have improved a lot in recent years, they are not as good as incandescent lamps. In many cases there is a minimum brightness level at the dim end of the scale, and below that level the lamp simply goes dark. If you insist on having continuously adjustable dimming at low light levels, you have to pay extra for special products with PWM-controlled LED lamps.
- **Light color:** The color temperature of LED lamps is usually specified. It indicates whether the light from the lamp is warm (yellowish) or cold (bluish), and it corresponds to the spectrum radiated by a black body at the specified temperature. A color temperature of 2,700 to 3,300 K yields warm light, which people living at relatively high latitudes usually prefer for lighting in their homes. By contrast, people who live in the Mediterranean region or near the equator are more comfortable with a color temperature above 4,000 K. A cool color temperature is also generally advisable for workplaces. Values in the range of 5,000 to 6,000 correspond to neutral white, also called daylight white. At 8,000 K or so the light is noticeably bluish. Values this high are very uncommon with LED lamps.
- **Light and power:** Although the efficiency of LED lamps is gradually improving, most commercial products only achieve a luminous efficacy of 80 lm/W. This means that you need a 10 W LED lamp to replace a 60 W incandescent lamp, because the latter produces about 800 lm. Typical 3 W LED lamps, which are mainly available with bases smaller than E27, are basically too dim because their light output is only 250 lm, corresponding to a 20 W incandescent lamp. When you buy LED lamps, it is therefore generally advisable to purchase the highest-power type available and avoid saving money in the wrong place.
- **Quality:** The blanket assumption that all LED lamps made in China are of inferior quality is clearly wrong, since you should know that virtually all LED lamps, except a few very expensive types, are made in Asia. The quality of a LED lamp is primarily a matter of good heat dissipation, since that is what determines its lifetime. After all, a 10 W lamp generates 8 to 9 W of waste heat. Visible cooling fins are therefore not a bad idea. Another factor is the complexity of the integrated switch-mode LED driver. However, this is practically impossible to judge from the outside.
- **Black sheep:** Because there is a lot of money to be made in this sector, you sometimes run into some distinctly black sheep. Two years ago the author bought the lamp shown in **Figure 12** as a "bargain priced" extra with another product. Normally it is not so easy to peek at what's inside, or you simply do not have any reason to do so. However, as a result of poor packaging and rough handling during transport, this lamp was already dismantled in the box and revealed what it had inside. Instead of a switch-mode LED driver, there is a suspiciously simple circuit that does not even have an input resistor to limit the inrush current. In addition, the lamp is advertised as a 10-W type but the LED circuit board inside the lamp is marked "7 W", and this was confirmed by a measurement. As so often, it looks like you get what you pay for. Nevertheless, the lamp is still working fine (for now).

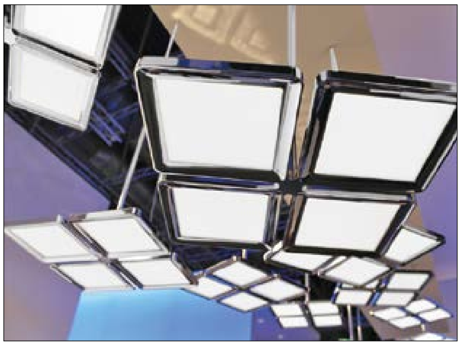


Figure 10. Modern OLED luminaires based on Philips OLED light tiles (photo: Philips) The production facilities for OLEDs have now been transferred to OLEDWorks.

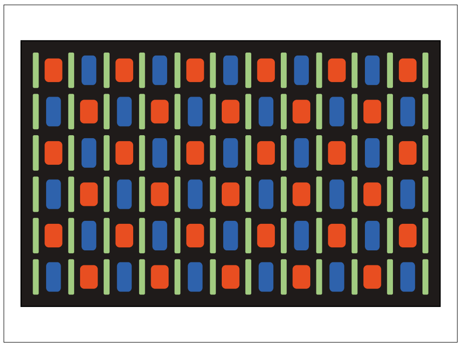


Figure 11. The typical pixel layout of an AMOLED display, in this case for a Samsung Nexus smartphone, is distinctly different from the layout of an RGB LCD.

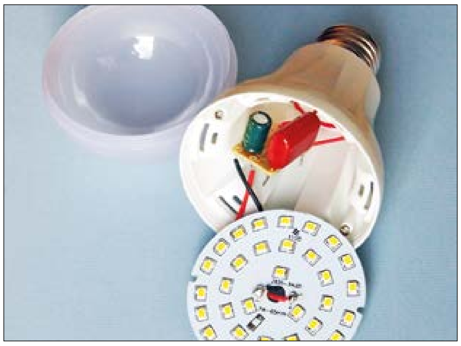


Figure 12. What's inside a cheap LED lamp: a series capacitor, a diode, a filter capacitor and a discharge resistor. The marking on the board says that the rated power is 7 W, instead of the claimed 10 W.

dates for large-area light sources, which are ideal for glare-free diffuse lighting. Presently Osram, Philips and OLEDWorks, among others, are developing OLED light tiles with edge lengths of 5 to 15 cm for lighting applications (**Figure 10**).

2015 is also a breakthrough year for OLEDs. Although there is still a lot of research being conducted in this area and OLED luminaires are presently cutting edge technology with prices to match, the advantage of simpler manufacturing will pay off in future. The breakthrough we are talking about here is in energy efficiency. Previously it was only possible to produce OLEDs with a luminous efficacy of 15 to 20 lm/W, basically the same as incandescent lamps, but now modules with 50 to 60 lm/W are feasible, putting them in the same ballpark as energy-efficient fluorescent lamps. With the largest dimensions currently possible, they can be used directly as backlights for LCDs because they do not present any problems with non-uniform light distribution. But why should you use an LCD when you can generate light directly with LEDs? The first commercial breakthrough for displays is already a fact. Large-screen OLED monitors are very expensive and only recently dropped below the €2K threshold for television sets in 55-inch format, but active matrix OLED (AMOLED) displays (see **Figure 11**) are now used in hundreds of smartphone and tablet computer models. Even cameras are being equipped with OLED displays. In small-format displays, the brilliant, high-contrast image with highly saturated colors and no viewing angle dependency outweighs the other drawbacks.

Outlook

In light of rapid technological progress towards very high efficiency and low cost, it can be assumed that non-LED lamps will very soon be limited to niche markets. Particularly with fluorescent lamps, this is good news due to disposal issues. However, what is often overlooked is that LED lamps also have an environmental downside. This does not relate so much to the LEDs themselves, but instead mainly to the necessary ballasts in the form of integrated switch-mode LED drivers, which end up on the electronics scrap heap at the end of their service life.

In vehicle headlamps, LEDs not only provide higher brightness and longer life than the vehicle itself, but also enable smart lighting that specifically avoids oncoming traffic to prevent blinding glare at night. If OLEDs continue to improve, they could open the door to glare-free lighting designs as well as innovative or transparent luminaires. This would eliminate the need for fluorescent or LED backlights for monitors, since OLED screens do not need backlighting. ⏮

(150183)

Web Link

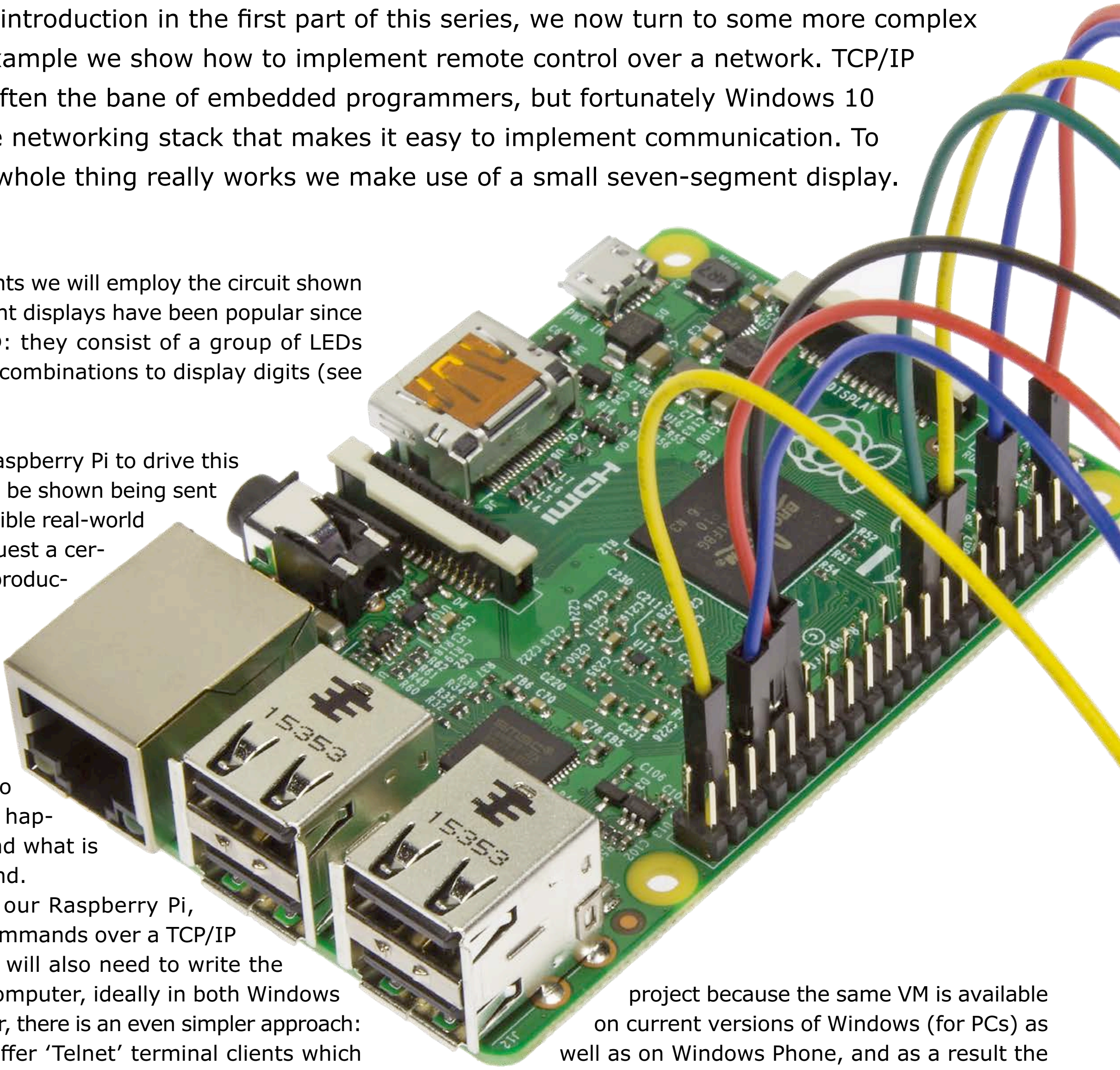
[1] First US LED patent (PDF): www.freepatentsonline.com/3293513.pdf

Windows on the Raspberry Pi (2)

Remote control of a seven-segment display

By **Tam Hanna** (Germany)

Following on from our introduction in the first part of this series, we now turn to some more complex applications. In this example we show how to implement remote control over a network. TCP/IP implementations are often the bane of embedded programmers, but fortunately Windows 10 includes a ready-made networking stack that makes it easy to implement communication. To demonstrate that the whole thing really works we make use of a small seven-segment display.



For the following experiments we will employ the circuit shown in **Figure 1**. Seven-segment displays have been popular since before the age of the LCD: they consist of a group of LEDs which can be lit in various combinations to display digits (see **Figure 2**).

Our application uses the Raspberry Pi to drive this display, with the number to be shown being sent over the network. One possible real-world use of this would be to request a certain number of items on a production line.

Telnet client

What makes implementing networked applications difficult is that when designing a protocol it is necessary to keep in mind both what is happening at the client end and what is happening at the server end.
In this case the server is our Raspberry Pi, which listens for control commands over a TCP/IP connection. Of course, we will also need to write the code to run on the client computer, ideally in both Windows and Linux versions. However, there is an even simpler approach: both Windows and Linux offer 'Telnet' terminal clients which implement a very simple protocol designed for exchanging textual information between machines.

Under the Microsoft operating system it is usually necessary to enable the Telnet client before it can be used: see the FAQ at [1] for more information.

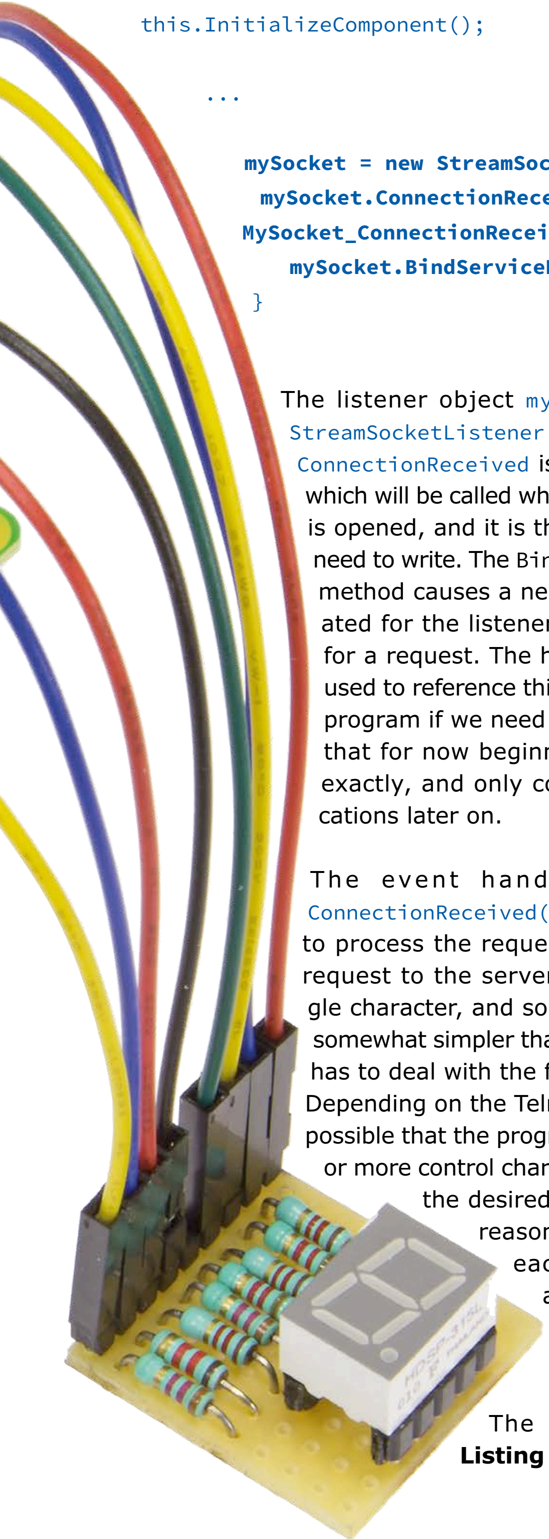
It's all asynchronous!

C# is a programming language whose runtime environment is implemented as a virtual machine (VM). This is relevant to our project because the same VM is available on current versions of Windows (for PCs) as well as on Windows Phone, and as a result the compiled server code for the Raspberry Pi can also be used on other platforms.
In order to prevent the use of polling loops or other potentially blocking constructs, Microsoft forces developers to use 'asynchronous programming'. For example, if a web server wants to receive characters over a TCP connection it must create a 'listener' object for incoming requests at the beginning of the program. After the listener object has been initialized it is passed a function which is to be called whenever a client opens a TCP connection and sends a request to the server. This event

handler function (see also the first part of this series [2]) contains the code to deal with the request.

The listener runs in its own thread, as if it were a separate program running in parallel with the main code. The consequence is that there is no point in the main program where it has to spin its wheels waiting for a request to arrive before proceeding: such ‘synchronous programming’ can lead to the whole application grinding to a halt.
For convenience we create the listener directly in the main page code (again, see the first part of this series [2]):

```
public MainPage()
{
     this.InitializeComponent();

          ...

            mySocket = new StreamSocketListener();
             mySocket.ConnectionReceived +=
            MySocket_ConnectionReceived;
                mySocket.BindServiceNameAsync(“234”);
 }
```



The listener object `mySocket` is of type `StreamSocketListener`.`MySocket_ConnectionReceived` is the event handler which will be called when a TCP connection is opened, and it is this function that we need to write. The `BindServiceNameAsync` method causes a new thread to be created for the listener, where it will wait for a request. The handle ‘234’ can be used to reference this thread later in the program if we need to. We recommend that for now beginners copy our code exactly, and only contemplate modifications later on.

The event handler `MySocket_ConnectionReceived()` contains the code to process the request. In our case the request to the server consists of a single character, and so the handler will be somewhat simpler than a web server that has to deal with the full HTTP protocol.
Depending on the Telnet client used, it is possible that the program will receive one or more control characters in addition to the desired character. For this reason we have to check each incoming character one by one to determine whether it is ‘normal’ or not.

The code is shown in **Listing 1**. First we create a

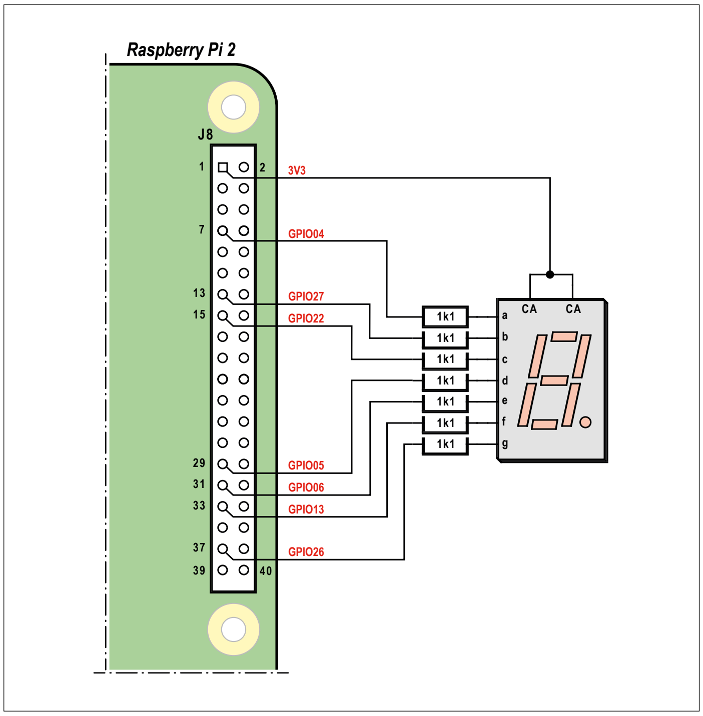


Figure 1. Our simple circuit displays digits sent over a network connection.

`DataReader` object, which will accept the characters that arrive over the TCP connection.
The line

```
uint sizeFieldCount = await reader.
LoadAsync(sizeof(byte));
```

causes the code to wait until a character (a character has a size of one byte) is received. The variable `sizeFieldCount` will be 1 if a valid character (as opposed to a control character) has been received.

The test can be found at the point marked ‘(1)’ in the listing: if a control character is seen, the `continue` statement causes the loop to start again from the beginning. Alternatively, if the character is valid its ASCII code is placed in the byte variable `what`, so that it can be processed by subsequent commands.

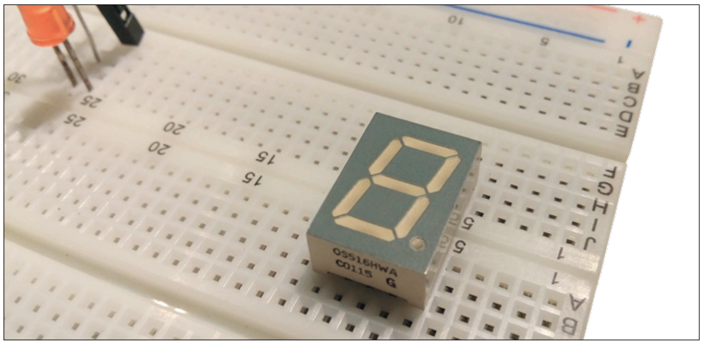


Figure 2. Most enthusiasts should be able to find a seven-segment display like this one in the junk box.

Listing 1. Receiving and checking characters.

```
private async void MySocket_
ConnectionReceived(StreamSocketListener sender,
StreamSocketListenerConnectionReceivedEventArgs
args)
{
    DataReader reader = new DataReader(args.
       Socket.InputStream);
    while (true)
    {
        reader.InputStreamOptions =
           InputStreamOptions.Partial;
        byte what;
        uint sizeFieldCount = await reader.
           LoadAsync(sizeof(byte));
        if (sizeFieldCount != sizeof(byte)) // (1)
        {
            continue;
        }

        what = reader.ReadByte();

        … // Code for processing
          // of character received
        // ASCII-Code is found in what
    }
}
```

LED control

To drive our seven-segment display we shall use a little trick from the world of systems programming. A key-value (KV) store is a kind of array whose elements can be of any desired type. Indeed, we can even write objects to the store. Elements are accessed using a key, which again can be of any desired type. The `configPin` method populates a data structure of this kind with the required GPIO pin objects (see [2]) to let us access the pins on the Raspberry Pi.

The method is called for each pin that is connected to the seven-segment display, in each case passing the pin number (using the conventional Raspberry Pi numbering system) and a character from ‘a’ to ‘g’ specifying the corresponding segment of the display. Here is the code for this function, which writes a GPIO object to the KV store. The function is also responsible for configuring the pin as an output.

```
private void configPin(int rpiPinNumber, char
ledSegmentChar)
{
    GpioPin workPin;
    workPin = myGPIO.OpenPin(rpiPinNumber);
    workPin.SetDriveMode(GpioPinDriveMode.Output);
    myKVStore.Add(ledSegmentChar,workPin);
}
```

Once the KV store is set up it is easy to access GPIO pin for a given LED segment via its name, from ‘a’ to ‘g’.

Now we need a routine that, given a digit received over the network, will tell us which LED segments should be lit. Again a little trick: we will represent the set of segments to be lit as a sequence of characters (from ‘a’ to ‘g’) stored as a string. The following function lights the required LEDs when passed a string (for example “abde”) specifying the desired pattern.

```
void setLeds(String activeSegmentsString)
{
    killLeds();

    for (int i = 0; i < activeSegmentsString.Length;
      i++)
    {
        myKVStore[activeSegmentsString[i]].
             Write(GpioPinValue.Low);
    }
}
```

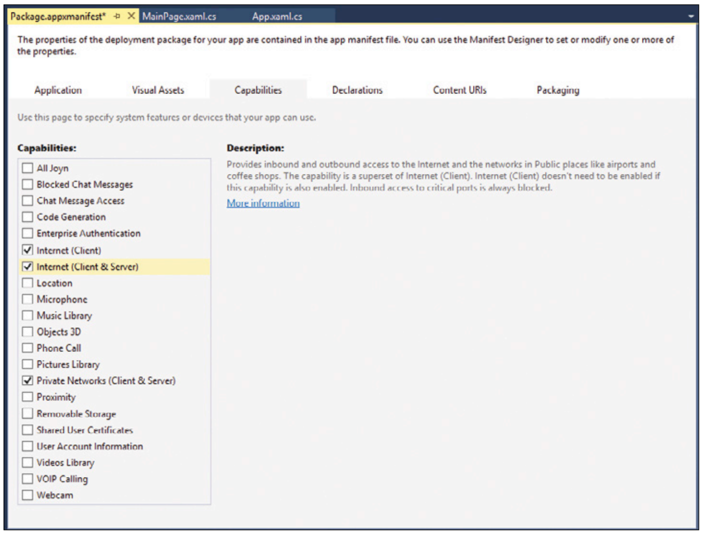
The expression `activeSegmentsString[i]` references the character occurring in position i in the string, which will be a segment letter from ‘a’ to ‘g’. Then `myKVStore[activeSegmentsString[i]]` gives us a GPIO pin object corresponding to that letter. We take the pin low and the LED segment will light. The function `killLEDs()` at the beginning turns off all the segments, and works in a similar way:

```
void killLEDs()
{
    String darkSegmentsString = “abcdefg”;
    for (int i = 0; i < darkSegmentsString.Length;
i++)
    {
        myKVStore[darkSegmentsString[i]].
Write(GpioPinValue.High);
    }
}
```

All we need to do now, when we receive a digit from 0 to 9, is to call `setLEDs()` with the right string to light the segments we want passed as its parameter. This can be done efficiently using a switch-case construct (see **Listing 2**). The code should be clear enough: note that we start with a check on the ASCII code for the character to verify that it lies in the range of digits from 0 to 9.

Incidentally, we do not run into any problems with the speed of this code when turning segments on and off using string parameters. The processing speed of the Raspberry Pi 2 is much higher than conventional microcontrollers, and as we saw in the first part of this series, the input/output driver is, shall we say, fast enough.

The whole program can be downloaded from the Elektor website as a Visual Studio project [3].



The properties of the deployment package for your app are contained in the app manifest file. You can use the Manifest Designer to set or modify one or more of the properties.

Use this page to specify system features or devices that your app can use.

Capabilities:

- □ All Joyn
- □ Blocked Chat Messages
- □ Chat Message Access
- □ Code Generation
- □ Enterprise Authentication
- ☑ Internet (Client)
- ☑ Internet (Client & Server)
- □ Location
- □ Microphone
- □ Music Library
- □ Objects 3D
- □ Phone Call
- □ Pictures Library
- ☑ Private Networks (Client & Server)
- □ Proximity
- □ Removable Storage
- □ Shared User Certificates
- □ User Account Information
- □ Videos Library
- □ VOIP Calling
- □ Webcam

Description:

Provides inbound and outbound access to the Internet and the networks in Public places like airports and coffee shops. The capability is a superset of Internet (Client). Internet (Client) doesn't need to be enabled if this capability is also enabled. Inbound access to critical ports is always blocked.

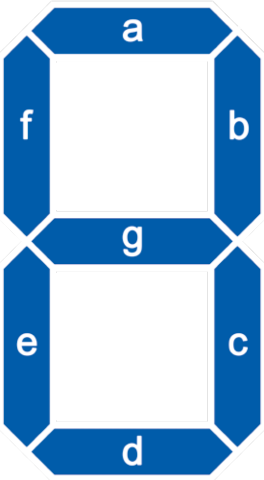
More information

Figure 3. The correct permissions need to be given to the app for it to function correctly.

Permissions

In order to ensure that the app works properly we need to set up the necessary permissions for it in Visual Studio. Microsoft secures apps using a system whereby developers must state in advance which system functions are going to be used by the program. Click on the file Package.appxmanifest and in the 'Capabilities' column tick the three attributes shown in **Figure 3**.

That is the final step (for now!): launch the program on the Raspberry Pi and connect to the machine over Telnet. Send a single digit and you should see it appear on the seven-segment display.



Conclusion

Single-board computers like the Arduino are ideal control and monitoring devices: they are optimized for the high-speed monitoring of processes. Ease of communicating with the outside world is a lower priority, as anyone who has had to deal with an Ethernet shield or who has struggled with the dreaded Arduino Yún bridge library will attest.

Windows 10 has the opposite emphasis: the system is optimized for efficient communication with the user and with other computers. The socket interface code we have seen here will run equally well on a normal workstation or on a Windows Phone; the disadvantage in comparison to Linux systems is the relatively poor real-time performance.

This disadvantage can be mitigated by using dedicated sensors or a second processor for process control, and in the next issue we will look at the programming interfaces that make this possible. Until next time! ⏮

(150519)

Web Links

[1] http://windows.microsoft.com/en-us/windows/telnet-faq

[2] www.elektormagazine.com/150465

[3] www.elektormagazine.com/150519

Listing 2.
Lighting the correct segments of the display.

```
what = reader.ReadByte();
if (what >= 48 && what <= 57)
{
    switch (what-48)
    {
        case 0:
            setLeds(“abcdef”);
            break;
        case 1:
            setLeds(“bc”);
            break;
        case 2:
            setLeds(“abged”);
            break;
        case 3:
            setLeds(“abgcd”);
            break;
        case 4:
            setLeds(“fgcb”);
            break;
        case 5:
            setLeds(“afgcd”);
            break;
        case 6:
            setLeds(“afgedc”);
            break;
        case 7:
            setLeds(“abc”);
            break;
        case 8:
            setLeds(“abcdefg”);
            break;
        case 9:
            setLeds(“abcdfg”);
            break;
    }
}
```

Router headaches

The Raspberry Pi 2 running under Windows 10 can sometimes cause trouble for routers: the author's Asus RT-N12E was giving ping latencies of around ten seconds. The problem can be worked around by adding a second router behind the first to provide the connection between the workstation and the single-board computer.

Advanced Functions

From 8 to 32 bits: Microcontrollers for Beginners (7)

Enabling the Brown-Out Detector

Just as in the world of eight-bit microcontrollers, ARM microcontrollers feature a ‘brown-out detector’ (or ‘BOD’ for short). This can trigger an interrupt to the CPU, or reset it, when the power supply voltage falls below a threshold value that you can configure. This threshold voltage is normally set below the recommended operating voltage of the device, and so the BOD provides a protection function to halt the operation of the microcontroller in the event of power loss before it can cause data corruption or other undesirable effects.

The SAM D20 is typical of ARM Cortex-M0+ microcontrollers in this respect: it has two BODs which can trip at different supply voltage levels. The first, BOD12, is factory-configured. It is always active when the CPU is active and monitors its 1.2 V core supply, which is derived using an internal voltage regulator. More interesting for us is the second BOD, BOD33, which monitors the 3.3 V power supply at VDDANA. This BOD can monitor the voltage either in ‘continuous mode’ or in ‘sample mode’. In sample mode the BOD measures the supply voltage on a given clock edge and then goes to sleep until the next clock edge: the clock is obtained from the internal 1 kHz source via a prescaler which can be configured to divide by up to 65536. A hysteresis, generally between 35 mV and 170 mV, can also be enabled.

Now we can look at the corresponding program, ‘Test of the BOD-module’, which initializes and configures the BOD33 module. **Listing 1** follows the usual pattern, where we first create a configuration structure (here `config_bod`) and then populate it with default values. We change values in the structure from their defaults as required, and then pass the settings to the BOD33 module. Finally we enable the module. In this example we configure the brown-out detector in continuous mode and to reset the processor when the voltage falls below the threshold voltage. The threshold voltage setting is 0x27 and hysteresis with thresholds V_{BOD+} and V_{BOD-} is enabled: see **Table 1**. If hysteresis is not enabled, the threshold voltage will be as indicated under V_{BOD} in the second part of the table; see also **Figure 1**.

The available voltage setting configuration values are given in the column labeled ‘BOD33 level’. In our example, where

Listing 1. The configuration function for the BOD33 module is straightforward.

```
void configure_bod(void)
{
  struct bod_config config_bod;
  bod_get_config_defaults(&config_bod);

  config_bod.mode          = BOD_MODE_CONTINUOUS;
  //config_bod.mode        = BOD_ACTION_NONE;
  config_bod.action        = BOD_ACTION_RESET;
  config_bod.level         = 0x27;
  config_bod.hysteresis    = true;

  bod_set_config(BOD_BOD33, &config_bod);
  bod_enable(BOD_BOD33);
}
```

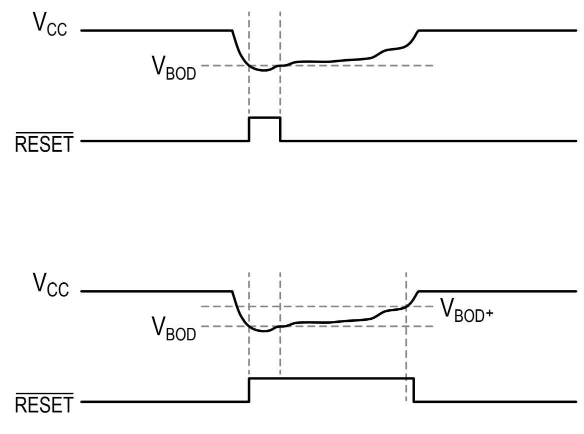


Figure 1. BOD operation, with hysteresis enabled (above) and with hysteresis disabled (below). (Diagrams and screenshots: Atmel).

By **Viacheslav Gromov** (Germany)

In the previous installments of this series we have covered all the important peripheral units and the ASF libraries that work with them. Here, in this penultimate installment, we look at some more advanced functions. The projects we describe are more or less independent of one another, and so you are free to pick and choose the ones that interest you and try them out. We will also give a few tips and tricks. A bonus project using the event system can be found at www.elektormagazine.com/articles.

we use 0x27 (39 decimal), this gives a typical V_{BOD+} value of 2.84 V and a typical V_{BOD-} value of 2.77 V. These threshold voltages can be adjusted by changing the configuration value in the program.

The main function calls the `system_init()` function in addition to the BOD configuration function mentioned above. The main loop is empty, and you can add your own code here. The BOD33 is active and will cause a processor reset if the supply voltage falls below the set threshold.

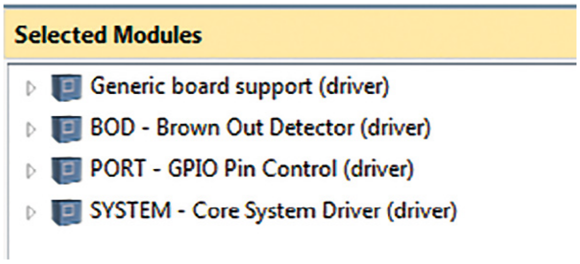
The BOD configuration can be changed to suit your own project. For example, if instead of

```
config_bod.action          = BOD_ACTION_RESET;
```

you use the command (commented out in the configuration function code)

```
config_bod.action          = BOD_ACTION_NONE;
```

you can then use the function `bod_is_detected(BOD_BOD33)` to 'manually' poll the BOD at any point. The function returns `true` in the case that the BOD33 module has seen that the supply voltage has fallen below the threshold, and you can then take whatever action is necessary. The detection flag should be cleared before polling it again, using the command `bod_clear_detected(BOD_BOD33)`.



Selected Modules

- Generic board support (driver)
- BOD - Brown Out Detector (driver)
- PORT - GPIO Pin Control (driver)
- SYSTEM - Core System Driver (driver)

Figure 2. Only the BOD library needs to be included using the ASF wizard.

We are now in a position to transfer the program to the board: note that for this program only the BOD library needs to be included using the ASF wizard (see Figure 2). Normally there will be nothing to see when the program is run: only when you have added some code in the main loop and then deliberately reduce the supply voltage below the threshold value will you be able to see the effect of the BOD in action.

Further information on the BOD33 module can be found in the SAM D20 datasheet from page 143 onwards, and at http://goo.gl/yVtRvK.

Table 1. BOD33 threshold voltages with and without hysteresis (source: Atmel).

Symbol	BOD33 level	Condition	Minimum	Typical	maximum
V_{BOD+}	6	Hysteresis ON	--	1.715 V	1.745 V
	7		--	1.75 V	1.779 V
	39		--	2.84 V	2.92 V
	48		--	3.2 V	3.3 V
V_{BOD-} or V_{BOD}	6	Hysteresis ON or Hysteresis OFF	1.62 V	1.64 V	1.67 V
	7		1.64 V	1.675 V	1.71 V
	39		2.72 V	2.77 V	2.81 V
	48		3.00 V	3.07 V	3.2 V

Sleep and reset functions of the SAM D20

The system driver library, which is present in all new projects, includes commands to access the sleep and reset functions of the SAM D20. You will already be familiar with the function `system_init()` from this library, used to initialize everything in the microcontroller. The most important commands relating to the sleep and reset functions are as follows.

- The function `system_reset()` resets everything in the microcontroller when it is called. Only peripherals which cannot (or should not) be reset, such as the real-time clock, are unaffected.
- The function `system_get_reset_cause()` returns a value which can be compared against various symbolic constants to determine the reason for the most recent reset. For example, this could be a software reset or a BOD reset.
- The function `system_set_sleepmode()` determines the behavior of the next call to the function `system_sleep()`. The degree of reduction in current consumption depends on the mode selected. A range of symbolic constants are available to specify the different modes, such as `SYSTEM_SLEEPMODE_IDLE0`: this is passed as the only parameter to the function. **Table 1** shows which peripherals are enabled and which disabled in the various sleep modes.
- The function `system_voltage_reference_enable()` requires a single parameter, which can be a symbolic constant such as `SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE`, to specify which voltage reference source is to be enabled. To disable a voltage reference use `system_voltage_reference_disable()` with the same parameter.
- The function `system_get_device_id()` returns a 32 bit integer. The value is a identification number for the device, unique among devices with the same part number. Most microcontrollers in the ARM Cortex-M0+ family offer a device ID function.

Further information can be found at www.atmel.com/Images/Atmel-42120-SAM-D20-System-Driver-SYSTEM_Application-Note_AT03260.pdf.

Table 1. Overview of system and peripheral operation in various sleep modes (source: Atmel).

Sleep mode	CPU clock	AHB clock	APB clocks	Clock sources	System clock	32 kHz	Regulator mode	RAM mode
IDLE 0	Stop	Run	Run	Run	Run	Run	Normal	Normal
IDLE 1	Stop	Stop	Run	Run	Run	Run	Normal	Normal
IDLE 2	Stop	Stop	Stop	Run	Run	Run	Normal	Normal
STANDBY	Stop	Stop	Stop	Stop	Stop	Stop	Low Power	Source / Drain biasing

SAM D20 tips and tricks

Peripheral power consumption

If you have not looked at the SAM D20 datasheet recently, now would be a good time to do so. In January 2015 the datasheet was updated to improve clarity and to add new information, often in the form of tables. Inevitably this means that the page numbering has changed.

In any case it is a good idea, now that we have looked at all the peripheral modules, to take a quick look at chapter 32 on page 571 (previously page 570) which describes the electrical characteristics of the microcontroller and peripherals, including current draws and permissible supply voltages. Table 32-9, which gives the current consumption of the microcontroller in various sleep modes, and Table 32-10, which gives the consumption of the peripheral modules, are particularly worth a look. The information in these tables is especially useful when working on low-power projects, helping to decide which peripheral modules are really needed and when the microcontroller can be put into sleep mode.

FreeRTOS

Even on the relatively small SAM D20 ARM Cortex-M0+ microcontroller it is possible to run a real-time operating system ('RTOS' for short), albeit without any form of graphical user interface. A popular choice for the 'M' family is FreeRTOS, which is also included in the ASF. It is interesting to take a look at the four example programs provided. FreeRTOS is an independently-developed system and works across a range of platforms: each manufacturer simply has to modify it to suit their particular microcontrollers and peripherals. The main advantage of this operating system is the high level of abstraction it offers. Once the initial barrier of understanding the task structure has been overcome, FreeRTOS makes it quick and easy to build large projects without worrying about which bits and bytes need to be set in which registers. More information on FreeRTOS, including a quick start guide, can be found at http://www.freertos.org.

Other interesting example programs

There are several libraries that we have not examined in this course because they require the use of expansion boards. They are nevertheless interesting to look at for the basic principles of the libraries involved, even if you do not go into them in great detail. Examples are the demonstration programs for interfacing to an SD card using various file systems, for interfacing to Atmel flash memory devices, and for calibrating the 8 MHz clock source. The corresponding API documentation is also useful.

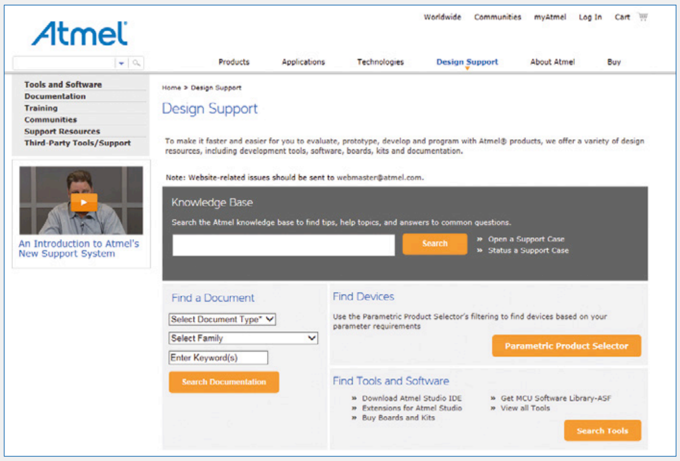
Programming tips

In our forums several people have reported difficulty in establishing communication between the SAM D20 board and Atmel Studio. Atmel's support site (**Figure 1**) at http://www.atmel.com/design-support/ has useful information regarding problems with recognizing the board or with updating the firmware: you can search for your particular problem and if you cannot find a solution, you can seek advice from Atmel experts.

When working within Atmel Studio 6 you will find that the integrated compiler is rather helpful and intelligent. For example, if you read in an application manual that the parameter for a certain command must be a pointer to a structure and you enter the structure itself as the parameter, Atmel Studio will accept the code and give a warning on compilation. For this reason you will sometimes see in the example programs that a structure is used when the pointer is strictly necessary. A similar remark applies to function prototypes, which in general are not mandatory. This user-friendliness makes life a bit easier for the programmer.

One final tip:

occasionally when compiling you may see an unexpected make file error. Recompiling the program from scratch usually cures this.



To make it faster and easier for you to evaluate, prototype, develop and program with Atmel® products, we offer a variety of design resources, including development tools, software, boards, kits and documentation.

Note: Website-related issues should be sent to webmaster@atmel.com.

Search the Atmel knowledge base to find tips, help topics, and answers to common questions.

Use the Parametric Product Selector's filtering to find devices based on your parameter requirements

Figure 1. Design support on the Atmel website.

The Peripheral Access Controller

As with many Cortex-M microcontrollers, it is possible to block the CPU in the SAM D20 from having write access to certain peripheral registers. The details vary from peripheral module to peripheral module, but it is usually the control register that is affected. If this write protection is enabled using the Peripheral Access Controller (PAC) the CPU is no longer able to change values in the register, thus preventing accidental access and hence potential errors. The corresponding ASF library is very simple: the project 'PAC Test1' demonstrates how the write protection can be enabled and disabled. As **Listing 1** shows, the program consists only of a main function, in which, after the initialization code, the port register is write protected using the command

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT),
   ~SYSTEM_PERIPHERAL_ID(PORT));
```

The parameters are the ID for the peripheral module in question and its complement (using the '~' operator). Having to provide the complement of the ID provides some protection against accidentally activating the write protect function.

The main loop causes LED0 on the board to flash. Before changing the bit in the port register (using the toggle command that we have seen previously) the write protection must be disabled; it can be re-enabled afterwards. To prevent an interrupt (which might need to access the port) from occurring while this code is being executed, we precede it with

```
system_interrupt_enter_critical_section();
```

to disable all interrupts, and follow it with

```
system_interrupt_leave_critical_section();
```

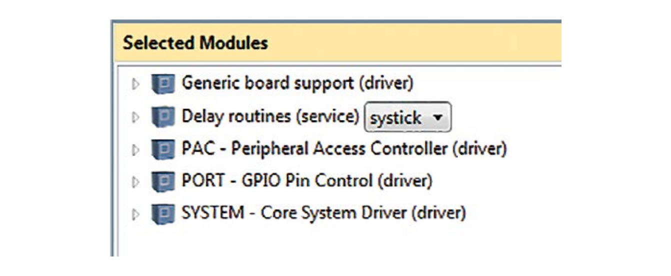
to re-enable them.

Listing 1. An excerpt from the main program showing the code to toggle the LED.

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
while(true){
   system_interrupt_enter_critical_section();
   system_peripheral_unlock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
   port_pin_toggle_output_level(LED_0_PIN);
   system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
   system_interrupt_leave_critical_section();

  delay_ms(500);
}
```

The command to disable write protection is `system_peripheral_unlock()`, taking exactly the same parameters as the enable function.
The command `delay_ms(500)` makes the period of the LED flashes equal to 1 s. Try out the program by downloading it to the board: the required libraries in the ASF wizard are shown in **Figure 1**.

Further information on the PAC can be found in the datasheet from page 34 onwards, and at http://goo.gl/kyuzWc.



Selected Modules

- Generic board support (driver)
- Delay routines (service) systick
- PAC - Peripheral Access Controller (driver)
- PORT - GPIO Pin Control (driver)
- SYSTEM - Core System Driver (driver)

Figure 1. The libraries that must be included in the project. Alongside the standard libraries we also need the 'Delay' and 'PAC' libraries.

Non-volatile memory (NVM)

The non-volatile memory area, in which the processor can store data that will be preserved even when power is lost, is structured like an EEPROM. As **Figure 1** shows, a part of the NVM is reserved for the bootloader and for internal settings, such as calibration values for the oscillators.
The middle area is available for user programs. More details on the structure of the NVM and how to calculate addresses can be found at http://www.atmel.com/Images/Atmel-42114-SAM-D20-Non-Volatile-Memory-Driver-NVM_Application-Note_AT03247.pdf.
We can now take a brief look at the project 'First test with NVM', which illustrates the three most important operations: erasing, writing and reading. After erasing the NVM a string is stored in it; then, whenever SW0 is pressed, the string is output over the UART interface.

The code includes the file 'asf.h' and then proceeds to function prototypes and the declaration of arrays. Next comes the configuration function for the U(S)ART and the short configuration function for the NVM. Here the configuration structure is first populated with its default values, and then we change the member `manual_page_write` to 'false', which means that the memory operates in automatic write mode. The settings are then transferred to the NVM module: see **Listing 1**. **Listing 2** shows the heart of the program, where the array `page_buffer` is filled with 64 values (the size of a page) from 0 to 63 using a 'for' loop. To be on the safe side we erase the page with the command

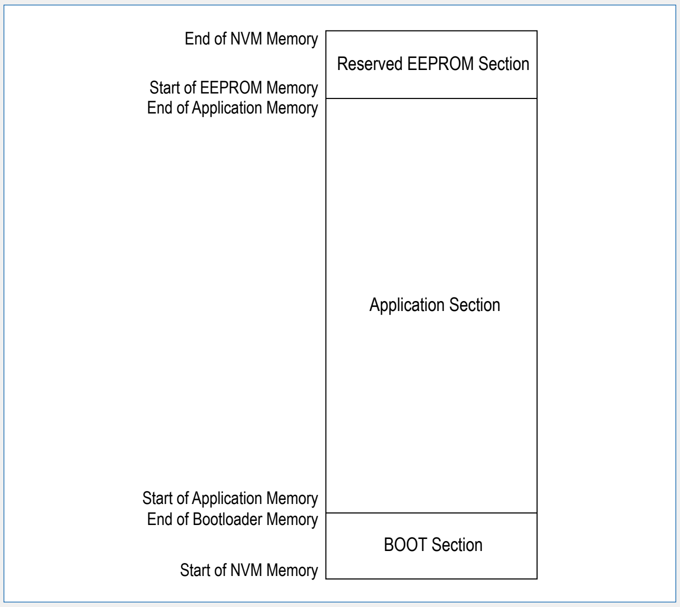


Figure 1. Memory areas in the NVM. Only the middle area is of interest to us.

Listing 1.
The compact NVM configuration function.

```
void configure_nvm(void)
{
  struct nvm_config config_nvm;
  nvm_get_config_defaults(&config_nvm);
  config_nvm.manual_page_write = false;
  nvm_set_config(&config_nvm);
}
```

```
while(nvm_erase_row(100 * NVMCTRL_ROW_PAGES *
   NVMCTRL_PAGE_SIZE) != STATUS_OK){}
```

The 'while' loop waits until the erase operation is complete. The parameter is the calculated page address (see the application note). Now we can write the contents of the array `page_buffer` to the NVM using the function

```
while(nvm_write_buffer(100 * NVMCTRL_ROW_PAGES *
   NVMCTRL_PAGE_SIZE, page_buffer, NVMCTRL_PAGE_
   SIZE) != STATUS_OK){}
```

(again, inside a 'while' loop) where the parameters are the calculated page address, a pointer to the array, and the page size.

Inside the main loop we use a read command to copy the contents of the NVM into the array `page_buffer_receive`. The command uses the same parameters as the write command and again is placed inside a 'while' loop. The command following is used to output the array over the U(S)ART interface.

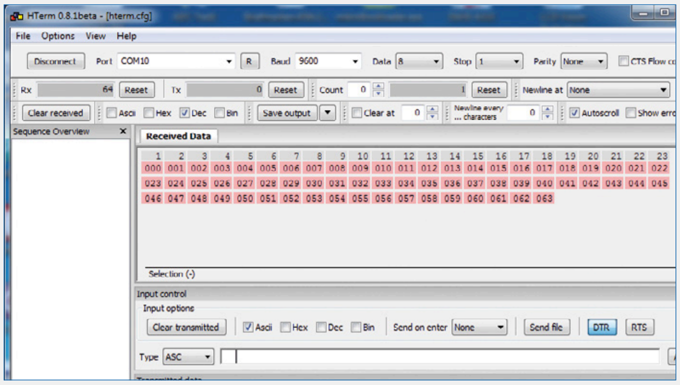
Listing 2. The most important part of the main function.

```
for (uint32_t i = 0; i < NVMCTRL_PAGE_SIZE; i++) {
  page_buffer[i] = i;
}
while(nvm_erase_row(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE) != STATUS_OK){}
while(nvm_write_buffer(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE, page_buffer, NVMCTRL_PAGE_SIZE) !=
    STATUS_OK){}
while (true) {
  while(port_pin_get_input_level(BUTTON_0_PIN)){}
  while(nvm_read_buffer(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE, page_buffer_receive, NVMCTRL_PAGE_
    SIZE) != STATUS_OK){}
  while(usart_write_buffer_wait(&usart_instance, &page_buffer_receive, 64) != STATUS_OK){}
  while(!port_pin_get_input_level(BUTTON_0_PIN)){}
}
```

```
while(nvm_read_buffer(100 * NVMCTRL_ROW_PAGES *
    NVMCTRL_PAGE_SIZE, page_buffer_receive, NVMCTRL_
    PAGE_SIZE) != STATUS_OK){}
while(usart_write_buffer_wait(&usart_instance,
    &page_buffer_receive, 64) != STATUS_OK){}
```

The two commands are enclosed inside a further 'while' loop that monitors the state of SW0, and so both are executed when SW0 is pressed. Transfer the program to the board and open a terminal program with settings as shown in **Figure 2**. Press SW0 and you should see the values stored in the NVM. This project uses the NVM library and the polled version of the U(S)ART library.

Further information can be found in the datasheet from page 25 onwards, and at http://goo.gl/afjKJj.



Port COM10 Baud 9600 Data 8 Stop 1 Parity None □ CTS Flow co

Rx 64 Tx 0 Count 0 1 Newline at None

□ Ascii □ Hex ☑ Dec □ Bin □ Clear at 0 Newline every ... characters 0 ☑ Autoscroll □ Show erro

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23
000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022
023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045
046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063
```

☑ Ascii □ Hex □ Dec □ Bin Send on enter None

Type ASC

Figure 2. Terminal settings and output when SW0 is pressed.

Interrupt and clock commands

In the application notes at http://www.atmel.com/Images/Atmel-42122-SAM-D20-System-Interrupt-Driver_Application-Note_AT03261.pdf and http://www.atmel.com/Images/Atmel-42119-sam-D20-D21-Clock-Management-Driver-CLOCK_Application-Note_AT03259.pdf there are numerous commands for controlling interrupts, clocks and other system-level features, which are definitely worth a look. Here are the highlights:

- `system_interrupt_set_priority()` requires as its parameters the vector for an interrupt (which can be a symbolic constant) and a priority level from 0 to 3, again possibly as a symbolic constant. The command sets or changes the priority of the specified interrupt. `system_interrupt_get_priority()` requires just the interrupt vector and returns the priority level of the corresponding interrupt.
- `system_interrupt_get_active()` takes no parameters and returns the interrupt vector for the currently active interrupt: the command can for example be used in an interrupt service routine.
- Commands such as `system_clock_source_osc32k_set_config()`, which take a configuration structure containing settings as a parameter in the usual ASF fashion, can be used to enable, disable and configure clock sources during program execution.
- The function `system_clock_source_get_hz()` allows you to determine the frequencies generated by the various clock sources. The parameter represents the clock source, and again symbolic constants are available. The function `system_apb_clock_source_get_hz()` is similar, returning the current frequency of the peripheral bus (APB) clock.
- Functions such as `system_cpu_clock_set_divider()` allow configuration of which divider is used for various purposes. The parameter can be a symbolic constant representing the divider to be used. ◀

(150324)

EAGLE Tips & Tricks (1)

Meet ULP and program EAGLE the way you want it

Neil Gruending (Canada)

Rather than tutoring on schematics and PCB tracks, let's take a look at a less known but exciting feature of Cadsoft's EAGLE: writing user language programs (ULPs). Personally I find that it's easier to learn a new programming language by reviewing an example to learn how it works. For this installment we will walk through the scripting environment in EAGLE and then we will look at a short example program.

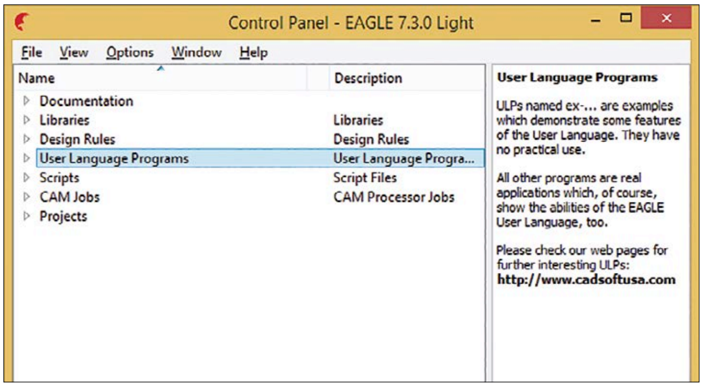
Programming in EAGLE

One of the unique features of EAGLE is that you can write your own programs that can interact with schematics, boards and even EAGLE itself. This is a lot more powerful than a many other PCB software packages that usually only provide some sort of rule mechanism to control how the various tools are applied. The Control Panel window shown in **Figure 1** is where you can find and run the user programs that EAGLE knows about. By default, EAGLE will look in C:\EAGLE-7.3.0\ulp for any files with a .ulp extension and then list them for you so that you can run them on a schematic, board or library by right clicking on the program name.

EAGLE also has scripts which are quite a bit different than user programs. Scripts are more like a macro language for doing things like importing files or configuring EAGLE whereas a user program is an actual program that can have a lot more functionality. For more information about the scripting language, please refer to the EAGLE manual.

Introduction to user programs

EAGLE user programs are text files with a .ulp extension that are written in user language program (ULP) syntax. The EAGLE PDF manual doesn't include the ULP documentation but it is available from the Cadsoft website [1] and the online help. If you've ever programmed in C before then you will be right a home with the ULP syntax because they are very similar. You can open and edit a ULP program by right clicking on it the EAGLE Control Panel and choosing Open. This will open the EAGLE Text Editor like in **Figure 2**. It's a good basic editor but I prefer work with color syntax highlighting and fortunately there's a syntax file available [2] for Notepad++[3]. You add the ULP syntax file to Notepad++ by opening the Language menu and choosing 'Define your language' which opens the User Defined Language window. Now click on the Import button and select the ULP syntax XML file that you downloaded earlier. Once you restart Notepad++ you will see an 'EAGLE ULP' entry in the Language menu. Now when you open a ULP you will get syntax highlighting like in **Figure 3** after adjusting the font settings and disabling the automatic spell checking.



Name	Description
Documentation	
Libraries	Libraries
Design Rules	Design Rules
User Language Programs	User Language Progra...
Scripts	Script Files
CAM Jobs	CAM Processor Jobs
Projects	

User Language Programs

ULPs named ex-... are examples which demonstrate some features of the User Language. They have no practical use.

All other programs are real applications which, of course, show the abilities of the EAGLE User Language, too.

Please check our web pages for further interesting ULPs: **http://www.cadsoftusa.com**

Figure 1. Control Panel window.

ULP file format

Now that we have set up our text editor let's take a look at the example ULP program in **Figure 4**. If you run it in a schematic or board the program will create a file called example1.txt in the EAGLE installation directory (C:\EAGLE-7.3.0 for me) with the text '1+2=3' in it.

The first line in the example starts with #require. This tells EAGLE that this program requires EAGLE version 7.0 or greater to run. The #require isn't required, but I usually stick it in just in case I end up using functionality that's only available in specific EAGLE versions.

The next line is the #usage declaration. This tells EAGLE what to display in the control panel when you click on the program like in **Figure 5**. Each line of text that following the usage line is surrounded by quotation marks to make them install a string for the usage directive. The '\n' at the of the first line is a new line character that tells EAGLE to start a new line of text. You can have 1 line for the usage or many, it's up to you. Both #require and #usage are example of directives, or commands to control how EAGLE interprets the rest of the ULP program. Directives are easy to spot in ULP because they always start with a # character, just like the C preprocessor.

The next few lines are comments or text that is ignored by EAGLE when executing the ULP but are still useful to the programmer. Single line comments always start to the right of two slash characters (//). Multiple line comments are always between '/*" and '*/' characters. The '/*' character flags the start of the comment and the '*/', the end. It's also important to remember that you can't nest multiple line comments, the first '*/' character after the comment has started will close the whole comment.

The first part of a ULP file is what EAGLE calls the declarations. This is where you can define any variables or functions you need. Our example starts by declaring two string variables named one and two. A variable declaration always begins with one of the following variable types:

- **int**: an integer number in the range of −2147483648 to 2147483648
- **char**: a single ASCII character or an integer in the range of 0 to 255
- **real**: a floating point number
- **string**: a sequence of characters that's usually used to store textual information

After the variable type are the variable names. Like most programming languages, EAGLE requires that variable names start with a letter or an underscore. Numeric digits are allowed after the first character like 'variable1'. It's also important to note that ULP variables are case sensitive which means that 'a' and 'A' would refer to two different variables. Our example defines 1 variable per line but you can also define more than 1 variable at a time by using commas between the names like this: 'int a, b, c;'. The example also shows how to initialize a variable with an 'equals' sign (=).

The example also shows how to define a simple function called 'add'. It takes two integer parameters together and then returns their sum as an integer.

The last part of a ULP file is the statement section which is where the program action instructions start. Our example uses the add function to add 1 + 2 and then stores them in a new variable named 'c'. After that comes a compound statement using EAGLE's built in output function. Compound statements always have opening and closing braces which controls the scope of the statements inside of them. In this case we're using the output function to tell the printf function to write to a file called example1.txt.

Conclusion

So far we've looked at how to run ULP programs and their basic structure. Next time we will start looking at a more complicated example that will read data from a design.

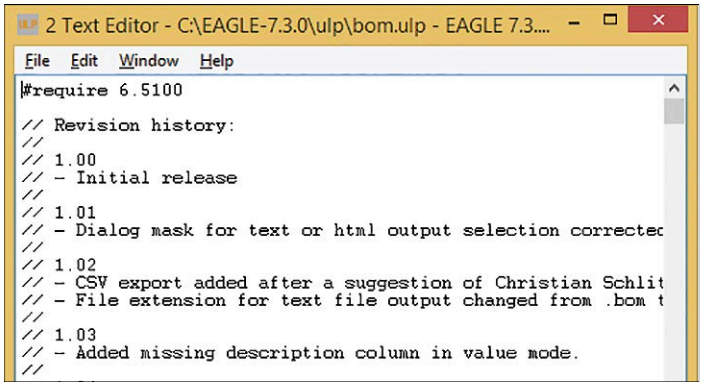
(150331)

Web Links

[1] www.cadsoftusa.com/downloads/documentation

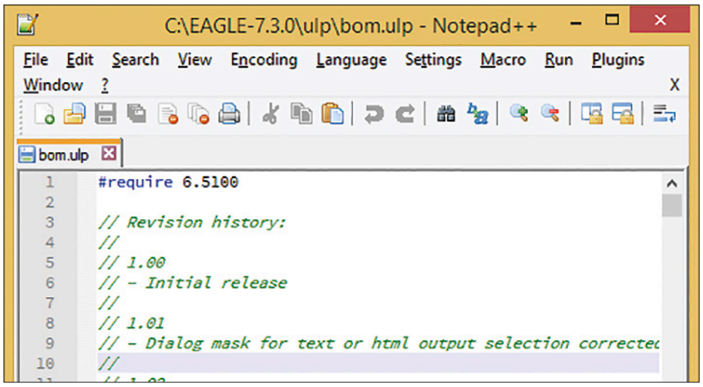
[2] www.cadsoftusa.com/downloads/file/notepad_userdefined-lang.zip

[3] https://notepad-plus-plus.org/



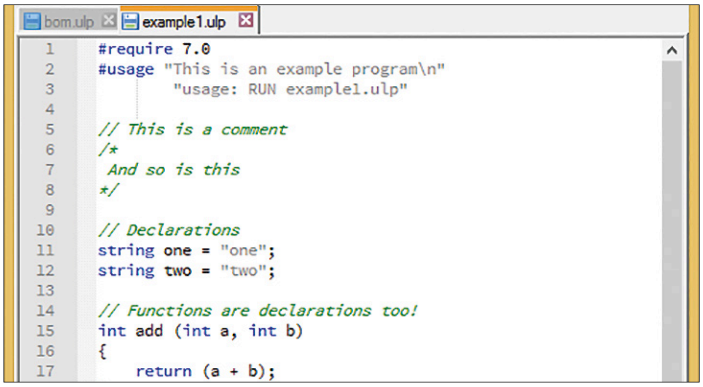
```
#require 6.5100

// Revision history:
//
// 1.00
// - Initial release
//
// 1.01
// - Dialog mask for text or html output selection correctec
//
// 1.02
// - CSV export added after a suggestion of Christian Schlit
// - File extension for text file output changed from .bom t
//
// 1.03
// - Added missing description column in value mode.
//
```

Figure 2. EAGLE Text Editor.


C:\EAGLE-7.3.0\ulp\bom.ulp - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins
Window ?
bom.ulp

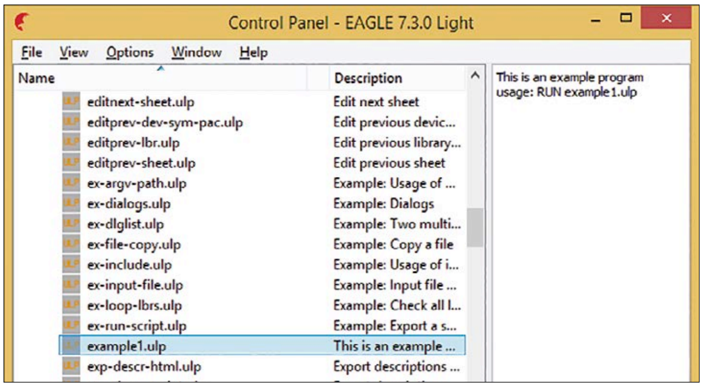
```
#require 6.5100

// Revision history:
//
// 1.00
// - Initial release
//
// 1.01
// - Dialog mask for text or html output selection correctec
//
```

Figure 3. Notepad++ Text Editor.



```
#require 7.0
#usage "This is an example program\n"
        "usage: RUN example1.ulp"

// This is a comment
/*
 And so is this
*/

// Declarations
string one = "one";
string two = "two";

// Functions are declarations too!
int add (int a, int b)
{
    return (a + b);
```

Figure 4. Example ULP program.

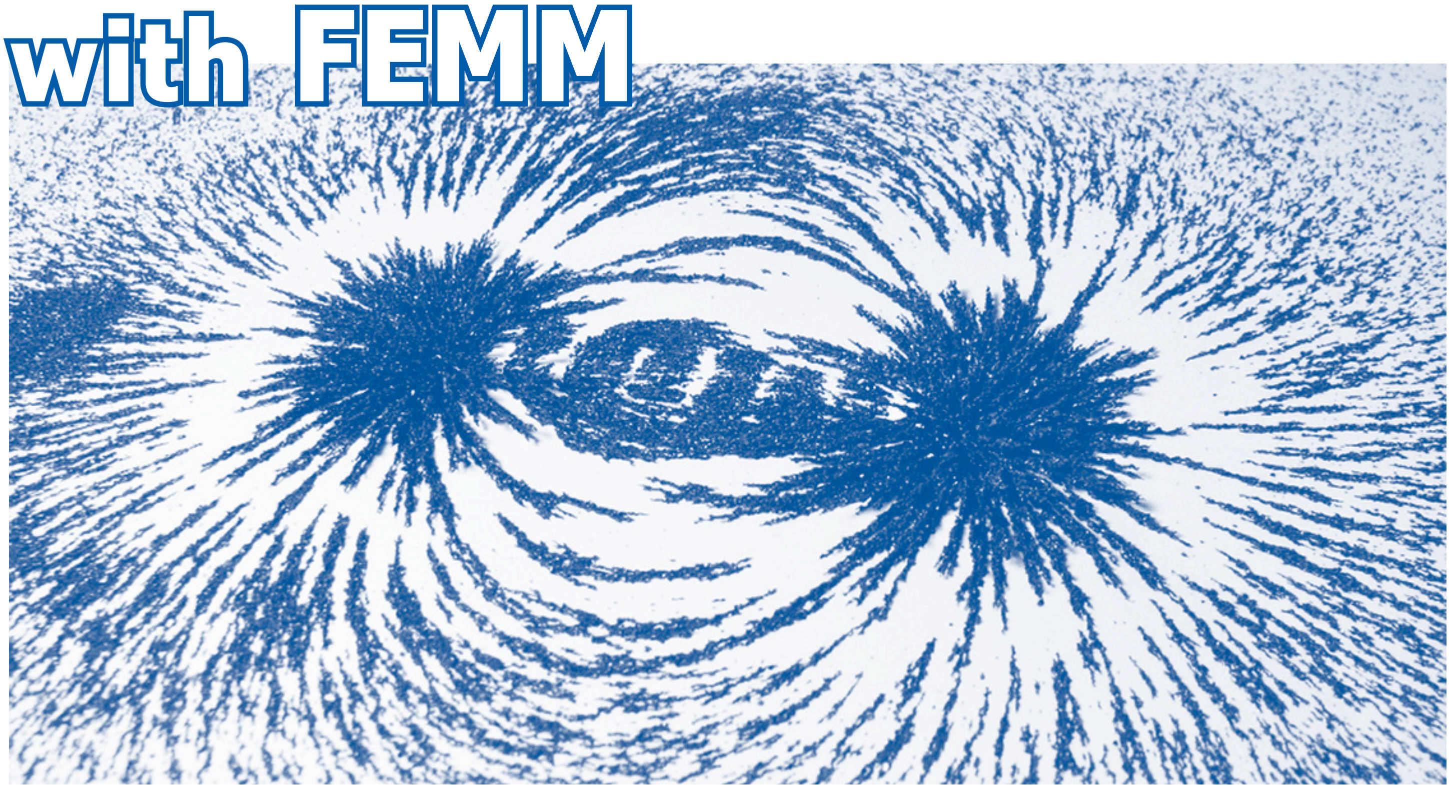

Control Panel - EAGLE 7.3.0 Light
File View Options Window Help

Name	Description
editnext-sheet.ulp	Edit next sheet
editprev-dev-sym-pac.ulp	Edit previous devic...
editprev-lbr.ulp	Edit previous library...
editprev-sheet.ulp	Edit previous sheet
ex-argv-path.ulp	Example: Usage of ...
ex-dialogs.ulp	Example: Dialogs
ex-dlglist.ulp	Example: Two multi...
ex-file-copy.ulp	Example: Copy a file
ex-include.ulp	Example: Usage of i...
ex-input-file.ulp	Example: Input file ...
ex-loop-lbrs.ulp	Example: Check all l...
ex-run-script.ulp	Example: Export a s...
example1.ulp	This is an example ...
exp-descr-html.ulp	Export descriptions ...

This is an example program
usage: RUN example1.ulp

Figure 5. Example program in the Control Panel.

Magnetic Field Simulation with FEMM



Spot-on dimensions of coils and transformers

Inductive parts including coils and transformers are components that have always had their vagaries. Frequently they are customized for specific applications, sometimes without proper understanding of their actual operation. By simulating these objects first, you can understand them better.

By **Martin Ossmann** (Germany)

Many solutions exist for calculating and simulating magnetic fields — and for representing them visually. An excellent Public Domain program for handling this is FEMM by David Meeker [1], which uses the Finite Elements Method (FEM).

A precondition in FEMM is that the design is rotationally symmetrical, but this is normally the case in practice. This article provides real-world examples to get you started. The simulation files are ready for downloading at [2], enabling anyone to reproduce and understand the results. Even better, you can easily modify the files for solving other problems.

FEM basics

At this stage we need mention only the key aspects that are important for our simulations. To master the basics of FEM itself, please check out the vast body of literature and the large number of tutorials on the web. With FEM simulations you first need to specify the simulation domain. Generally you select a sphere large enough to accommodate the entire magnetic field of interest. If you are unsure whether the domain you selected was large enough, simply double the radius of the sphere and check whether the results alter significantly. On the other hand it makes sense to keep the domain relatively small, as otherwise it can take ages to calculate the results. In FEM we disassemble or 'decompose' the domain into small elements. FEMM uses triangular elements for this purpose and we need to determine how fine the decomposition needs to be. In the vicinity of small, critical details of geometry (e.g. air cores and thin slices) we need several triangular elements at least to display the smallest size of detail. You can make the decomposition very much coarser in peripheral or fringe areas.

Differing levels of fine precision are specified in our examples. If you are uncertain whether the decomposition is sufficiently detailed, you can reduce the degree of precision (by halving the parameter *MeshFactor*). If the results do not alter greatly, you can carry on, assured that everything is set up correctly. You should avoid excessive decomposition though, as this again would lead to very lengthy computation times.

Long thin air coils

All of this becomes tangible in our first example, in which we're dealing with the magnetic field of a long and thin cylindrical coil. **Figure 1** illustrates the relevant set-up. The left-hand boundary is the axis of rotation. Rotating the semicircle generates the sphere that forms the simulation domain. The long, narrow rectangle is the opening for displaying the winding of the coil. The act of rotating this produces the slender cylindrical coil.

Figure 2 displays the decomposition into triangular matrices on the top of the cylindrical coil. You can see how, in the field of the coil, the decomposition is finely detailed in order to simulate this thin layer well. Towards the outer surface the decomposition becomes coarser.

In FEMM we can also display the simulation results graphically, e.g. the magnetic field strength as 'density plot' or by using vectors. **Figure 3** shows the result for the upper end of the cylindrical coil. Inside the coil we find, as expected, a concentrated and uniform (homogeneous) field. Outside this area the field is meager. Using FEMM you can also evaluate the field strength spatially at various points.

Results for 'Long Thin Coil':

```
L (by formula) = 19.90 µH
L (FEMM)    = 17.23 µH
H_Inner (by formula) = 1666.6 A/m
H_Inner (FEMM)    = 1636.8 A/m
```

The results calculated according to the formula and by FEMM simulation turn out to be in close agreement.
In order to visualize the coil arrangements more clearly we can export the data in our examples using a Lua script (see **inset**) into POV-RAY software [3], enabling us to render scenes in 3-D. **Figure 4** shows the 3-D representation of the simulation calculations for the long, thin cylindrical coil.

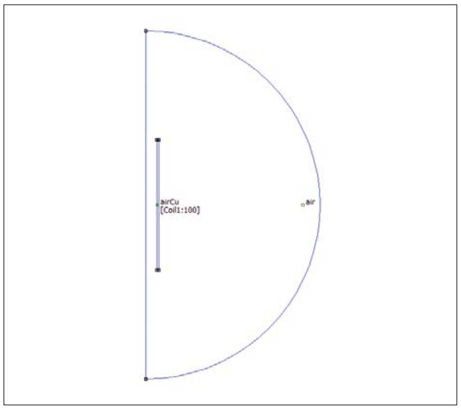


Figure 1. Simulation domain.

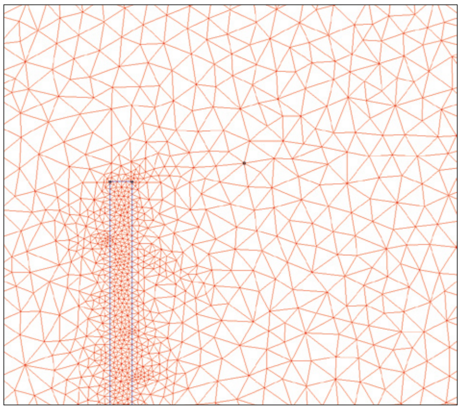


Figure 2. Triangular partitioning.

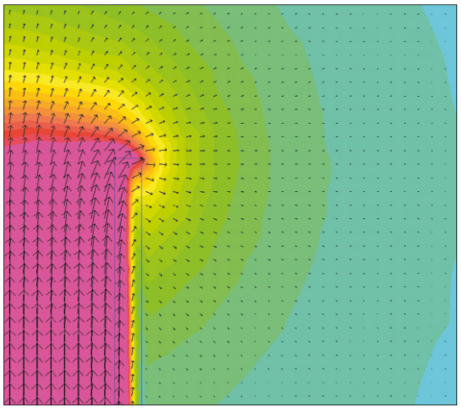


Figure 3. Density/vector plot.

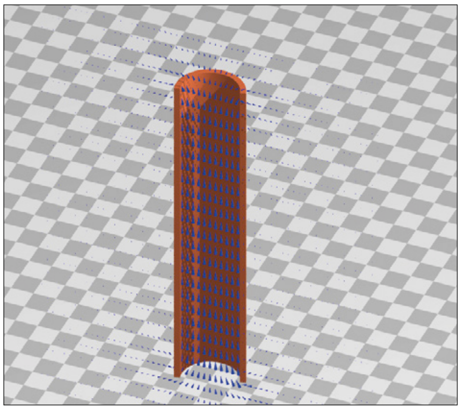


Figure 4. 3-D image of the long thin coil.

The copper-colored tubular section represents the coil. The blue arrows indicate the strength and direction of the magnetic field.

Ferrite rods

Now we're going to calculate a cylindrical coil wound on a rod of ferrite material, the actual object being shown in **Figure 5**. The coil has 60 turns, whilst the ferrite core is 10 mm in diameter and 40 mm long. The measured inductance for this arrangement is L = 216 µH.
In the simulation we retrofit the air coil used previously with a core made of material having a relative permeability of 2200. **Figure 6** shows the result in the form of a 3-D field. The simulation delivers an inductance of L_{FEM} = 215 µH, which matches the measured value closely.

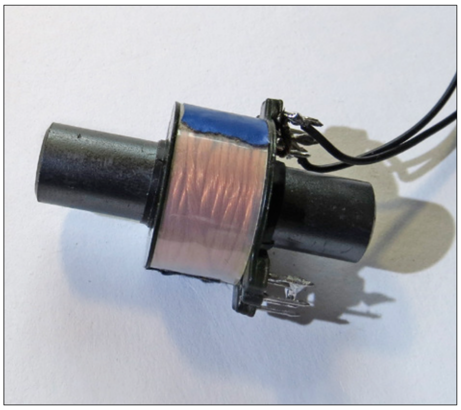


Figure 5. Coil with ferrite rod core.

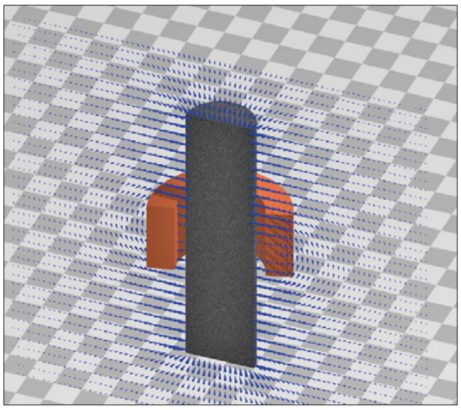


Figure 6. Simulation of a coil with ferrite rod core.

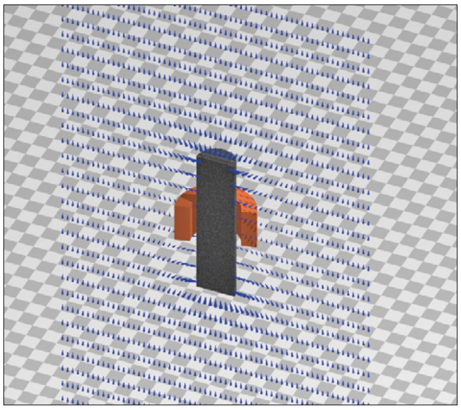


Figure 7. Ferrite antenna in a uniform field.

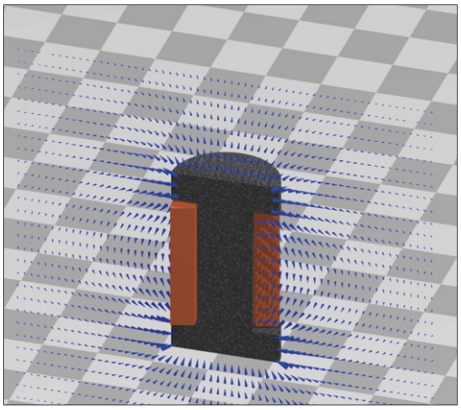


Figure 9. Coil wound on drum core.

Results for 'Coil with ferrite rod core':

```
Coil using rod core, 60 turns
  L (measured) = 215.7 µH
  L (FEMM)     = 214.9 µH

Coil without core
  L (measured) = 42.7 µH
  L (FEMM)     = 41.3 µH
```

Figure 11. Radiating coil for inductive charging.

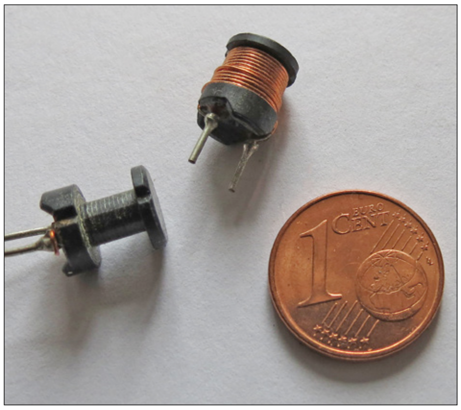


Figure 8. Drum core.

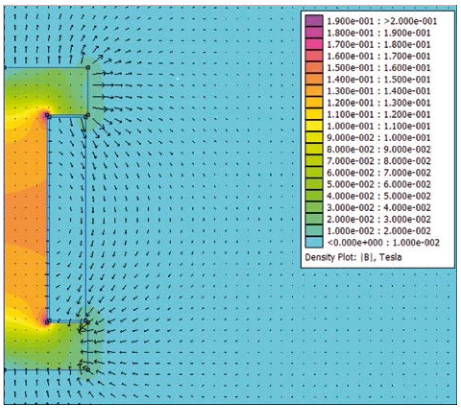


Figure 10. Field strength for coil wound on drum core.

The measured values agree closely with the values calculated using FEMM. Adjusting this geometry enables us to calculate numerous variations on the ferrite core theme. It's equally easy to predict the inductance of ferrite antennas. You can also predict the extent to which the inductance changes when you displace the winding along the ferrite rod.

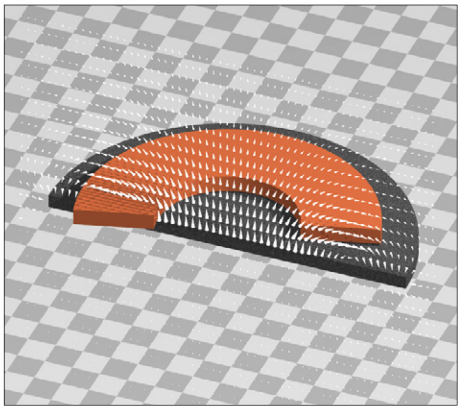


Figure 12. Circular model of the radiating coil.

Ferrite antennas

As well as the inductance of a ferrite rod antenna, it is of course interesting to know the voltage that will be induced in it when placed in a magnetic field. FEMM can handle this too. Up to now we have always performed field calculations without any external magnetic field. Now we introduce, by means of a suitable choice of boundary conditions, a magnetic field of *H* = 1 A/m with a frequency *f* = 200 kHz (Long Wave). In this situation we employ a cylindrical shaped simulation domain, which simplifies the formulation of the boundary conditions. The induced voltage is around 130 mV. In the air coil (without any core) a voltage of about 20 mV would be induced. This is easy to work out, simply by assigning to the core a relative permeability of 1 (air core). The modestly-dimensioned 40 mm long ferrite rod now multiplies the induced voltage six-fold.

In **Figure 7** we have another 3-D image. At the lower end of the core we can see the magnetic field entering it and then seeping out again at the top. In this way we can use FEMM calculations to good effect for visualizing and understanding magnetic fields.

Drum cores

Now we want to investigate how much the core material of a drum (or 'cotton reel') coil is affected (influenced) when we pass a specified current through the coil. Let's take a coil having L = 100 µH (such as Reichelt Electronics type L-007HCP, **Figure 8**), which is specified to have a current rating of 900 mA. Dismantling the coil, we discover it has 56 turns. We shall now size up the core geometry and transfer this information into the FEMM simulation.

In **Figure 9** shows the field pattern at a current of *I* = 1 A, just slightly above the maximum current specified. You can see that a strong field is created around the coil opening. In this kind of coil it leads readily to elevated eddy current losses if the wire used is too thick.

Result for 'Drum core':

```
  L (measured) = 99.8 µH
  L (FEMM)     = 94.5 µH
```

The measured and FEMM results are once again in agreement.

What interests us now is the magnetic flux density, B, in the core. We can display this in FEMM as a false-color image, with the result shown in **Figure 10**. The greatest flux density, of around 200 mT, is achieved at the inner vertices. The planar areas of the core are modulated to a visibly weaker extent. It is evident that the core material is in a state well below saturation; a figure around 300 mT is normally used in power ferrites. As you can see, you can use FEMM calculations for dimensioning power-rated magnetic components as well.

Radiating coil for inductive charging

In this next example we investigate the radiating coil used for charging smartphones in the QUATpower wire-free charger module WCTX-12USB (Pollin Electronic; eBay item 141448022615). In this case the coil is circular and the ferrite plate rectangular (**Figure 11**).

In our simulation we have substituted a circular disk for the square plate, retaining similar dimensions. In the results (**Figure 12**) you will see that it is still possible to determine the inductance pretty accurately. This is because the field strength in the outer regions of the coil, beneath the plate, falls off relatively quickly. So the precise shape of the plate doesn't matter too much.

Results 'Radiating coil 20 turns on plate':

```
  L (measured) = 23.0 µH
  L (FEMM)     = 22.5 µH
```

Actual result and FEMM calculation in close agreement again!

Pot cores for wire-free energy transfer

Next we turn to a design used for inductive charging. Two PS47 pot cores each have 20 turns of wire and are positioned some 8 mm apart (**Figure 13**). Like this they form a (poorly coupled) transformer. This time the calculation is a bit more complicated, because we need to determine several parameters of a transformer in order to define its magnetic behavior correctly. **Figure 14** illustrates two possible equivalent schematics along with their parameters. The upper one employs the customary division into main inductance (L_m) and stray inductance (L_s) together with an ideal transformer with a transfer relationship of 1:*M*. The lower right-hand equivalent schematic is used, for example, in SPICE. This employs the separate inductances L11 and L22 of the two windings together with the coupling factor *k*.

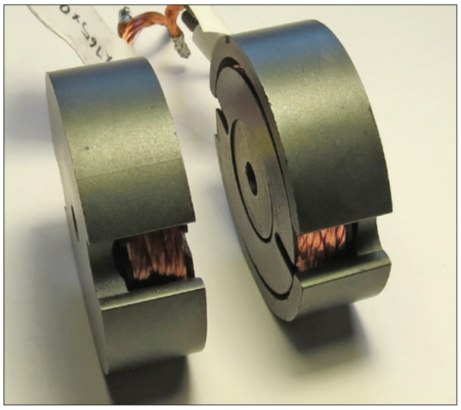


Figure 13. Pot core for inductive energy transfer.

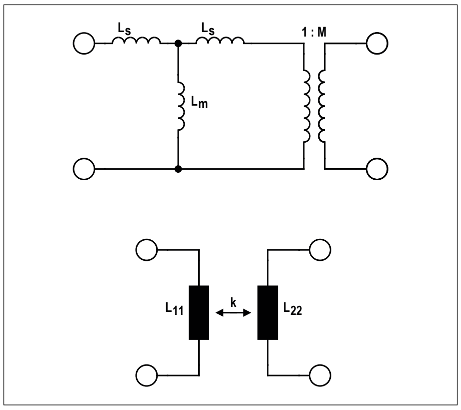


Figure 14. Transformer equivalent diagrams; SPICE is below.

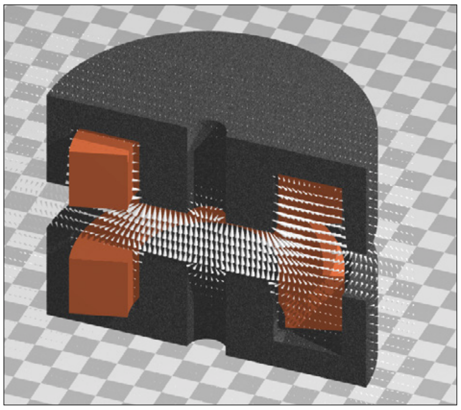


Figure 15. Pot cores (sometimes called cup cores).

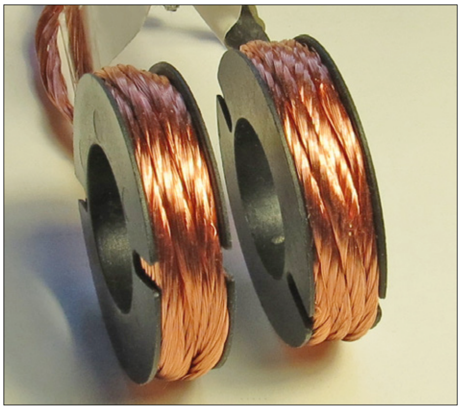


Figure 16. Air-cored transformer.

As our arrangement is symmetrical (same construction in the primary and secondary windings) L11 = L22 is valid. In this case *k* is simple to define metrologically; the problematic part is the no-activity (quiescent) transfer ratio (if *k* is close to 1, as in a good transformer). You are then better off measuring the no-activity short circuit inductances.

Results 'Pot core PS47':

```
Nprim=Nsec=20 i.e. M=1

Measured:
  L11=L22  = 48.0 µH
  k        = 0.35

FEMM:
  L11=L22     = 46.7 µH
  L21=L12=Lm  = 17.4 µH
  Ls          = 29.3 µH
  k           = 0.37
```

Good correlation between measured values and FEMM calculations once again. The simulation is shown in **Figure 15**. To determine transformer parameters, we calculate the two field situations. First let's implant some primary current and determine the flows in the two windings, assuming no activity on the secondary side. Then we repeat everything with current in the secondary winding and nothing happening on the primary side. From the current flow data you can then determine all the surrogate parameters of the transformer. In our case the degree of coupling (*k* = 0.37) does not get our pulse racing but that's the price you pay for contactless energy transfer through air gaps. Now you can also explore how much influence a different layout of the windings or alternative core geometries would have.

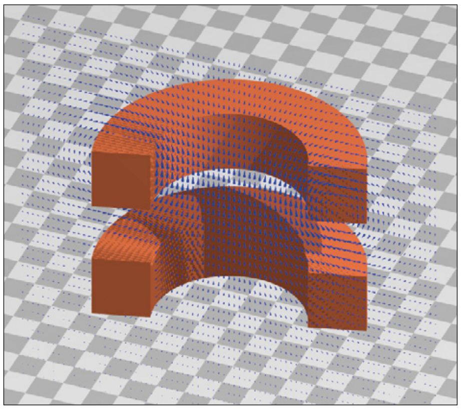


Figure 17. Simulation of an air-cored transformer.

Theoretically of course you also have the opportunity to simulate transformers with multiple windings. On the other hand, the calculations and equivalent diagrams become substantially more complex.

Air coil transformer

Once you have completed the arrangement using pot cores, you can omit the cores and rapidly calculate a set-up using air coils (**Figure 16**).

Figure 17 shows the field pattern. At k = 0.2 the coupling is clearly worse than before. The inductances are L11 = L22 = 11 µH, in both cases again with twenty turns.

Results 'Air coils':

```
Measured:
  L11=L22 = 11.7 µH
  k       = 0.2

FEMM:
  L11=L22     = 10.6 µH
  L21=L12=Lm = 2.0 µH
```

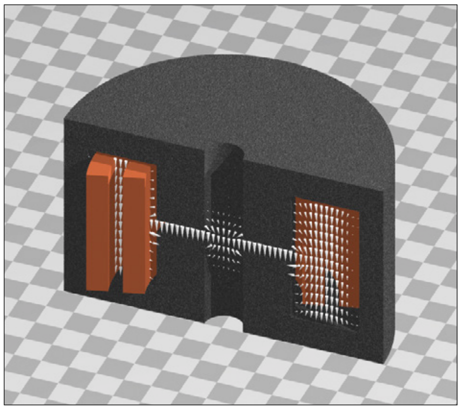


Figure 18. Transformer with air gap.

```
  Ls          = 8.6 µH
  k           = 0.19
```

Pot core transformer with air gap

In our final experiment we'll calculate a transformer with an air gap and good coupling, as used for example in flyback converters. We'll re-use our PS47 cores, this time directly above one another, with an air gap in the middle. The windings are now phased radially. Although both windings have 20 turns, the current set-up is no longer symmetrical. It's now evident in the equivalent schematic that the transfer ratio M of the ideal transformer is no longer M = 1 and the two inductances L11 and L22 are not equal any more.

Results 'Transformer with air gap':

```
Nprim=Nsec =20

FEMM:
  L11 =      94.60 µH
  L12=L21=   95.62 µH
  L22 =     105.79 µH
  Lm   =     90.42 µH
  Ls   =      4.18 µH
  k    =      0.96
  M    =      1.06
```

The coupling factor (k = 0.96) is now relatively high. The distribution of the field depends of course on the currents in the windings. In our example I_1 = 1 A and I_2 = −1.2 A. Since the currents are more or less equally opposed (i.e. energy flows through the transformer from the primary to the secondary winding), the field between the two windings is particularly strong. However, since the currents do not cancel out exactly, there is an additional field in the hollow core (**Figure 18**).

Your turn now

Files for all of the set-ups in this article are ready to download at [2]. For each example there's a dedicated .LUA file, which produces all the other files when processed in FEMM. This way all the information is contained in one single file and can be adapted easily by modification to your own needs. ⏮

(130565)

Web Links

[1] www.femm.info/wiki/HomePage

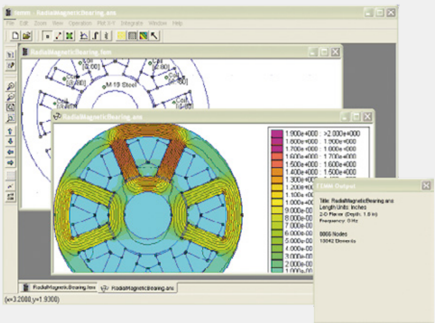
[2] www.elektormagazine.com/130565

[3] www.povray.org

Control scripts using Lua

FEMM includes a graphical user interface (GUI). However, its author recommends using the built-in script language Lua for fully automatic control. This lets you configure geometries and all other parameters precisely and reproducibly. All the examples in this article are available as Lua files, so you can execute them without further ado. The advantage of Lua is its expandability, meaning that you can write additional functions yourself that you find handy. The author uses a whole raft of functions like this to simplifying the geometry entry process.

Finite Element Method Magnetics



Tips and Tricks

From and for readers

Here's another clever way of making electronickery less frustrating.

Repairing a faulty STK600

By Wolfram Pioch

The STK600 from Atmel is a powerful and flexible development board for AVR controllers, although at around $300, it's not exactly cheap. So after accidentally mis-wiring a STK600 to the LED header connectors of two STK500 boards, I was not particularly pleased to discover that the STK600 had stopped responding to commands. Worse, its current draw had risen to several hundred milliamps. It appeared that current was being turned straight into heat.

*Using my finger as temperature sensor, I withdrew it very rapidly when I found the offending component, located close to the power connector (**Figure 1**, circled in red).*

A search online indicated I was not the only one to suffer this fault. At [1] is a video that shows how to replace a defective TPS62040 DC-DC converter, which comes in an MSOP Powerpad package, with a thermal pad on the underside [2]. That didn't improve my mood, as apart from getting hold of one of these chips, even installing it would be fraught with some risk.

But there's a simpler and cheaper method. The voltage regulator on the STK600 must produce 5 V with 6.1 V on its input. This task can also be undertaken by a low-dropout regulator like the LM2940 (5 V), which also offers a greater input voltage range [3]. In contrast to the switching regulator, it can also cope with a full 1 A.

Taking out the defective IC is relatively unproblematic using a hot air blower (once removed, the board's current consumption should revert to a few mA once more).

*Fortunately the capacitors on the input and output of the switching regulator make substituting the linear regulator entirely feasible. In preparation for this you need to slightly shorten and bend out the pins of the LM2940, as shown in **Figure 2**. For safety reasons we need to glue an insulating underlay (PC board material without any copper cladding) below it.*

*In **Figure 3** you can see how the regulator is soldered to the capacitors beside where the switching regulator was removed.*

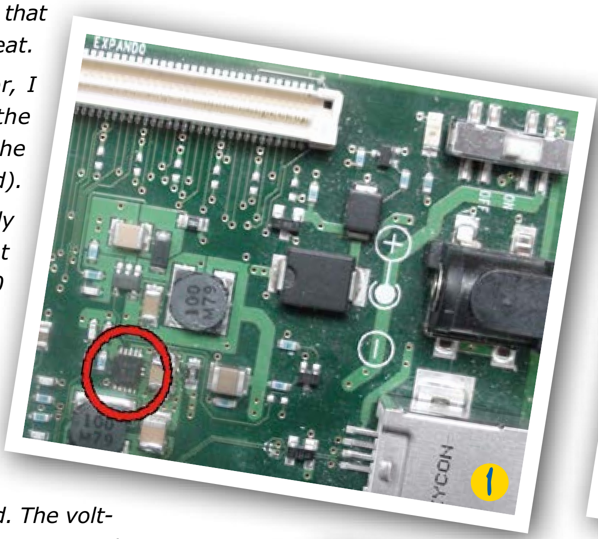
The result: the STK600 is back in service again.

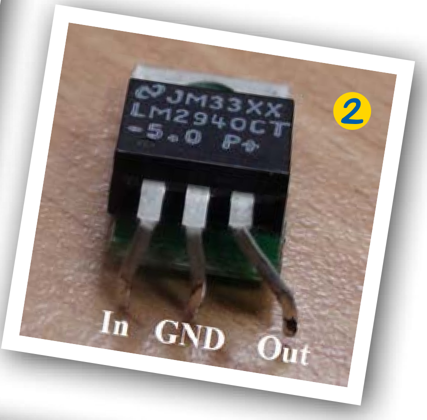
(150562)

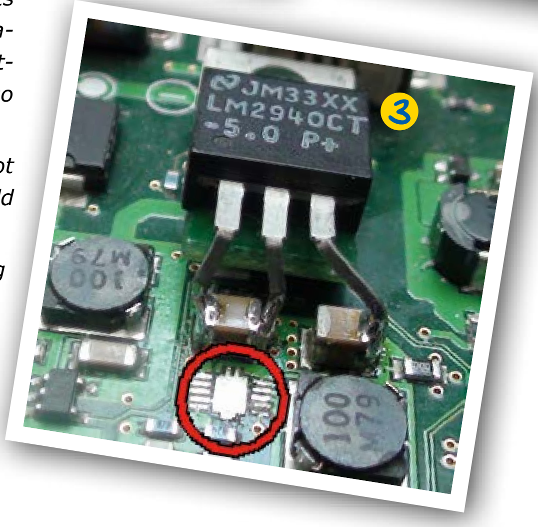
[1] www.youtube.com/watch?v=Xr4maDazcWY

[2] www.ti.com/lit/ds/symlink/tps62040.pdf

[3] www.ti.com/lit/ds/symlink/lm2940-n.pdf







Have you come up with an inspired way of solving a really challenging problem? Or found an ingenious but 'alternative' way of using some component or tool? Maybe you've invented a better or simpler way of tackling a task? Do write in – for every tip that we publish, we'll reward you with $40 (or local equivalent)!

Q&A LEDs

(Vitually) Everything You Ever Wanted to Know About ...

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

Some things taken for granted by the electronics pros are not entirely clear to enthusiasts. With LEDs, some facts need to be known if you want to have them last a long time and/or be as bright as possible.

Q *Why should you always use series resistors with LEDs?*

A Series resistors are actually not essential for LEDs; what matters is ensuring a constant current through the LEDs. The series resistors commonly used with small LEDs convert a constant supply voltage into a constant LED current. This is necessary because LEDs are basically diodes, as the name says. Consequently their current versus voltage characteristics are non-linear (**Figure 1**) and they have rather low dynamic resistance (**Figure 2**). These characteristics can also vary considerably between different types and different manufacturers. For this reason, LEDs cannot be operated directly from a constant voltage because small voltage variations would cause large current variations. Even if the voltage is set very precisely, thermal drift would lead to destruction of the LED because the forward voltage of LEDs drops with increasing temperature.

With a series resistor, the current through the LED is relatively constant. The greater the difference between the supply voltage and the forward voltage, the more constant the current. The forward voltage depends on the color of the LED and is approximately 1.7 V for red, 2 V for yellow and 3.5 V for blue or white. You should use the following formula to choose the value of the series resistor for small LEDs:

$R = (V_S - V_{LED}) / I_{LED}$

With a green LED and a supply voltage

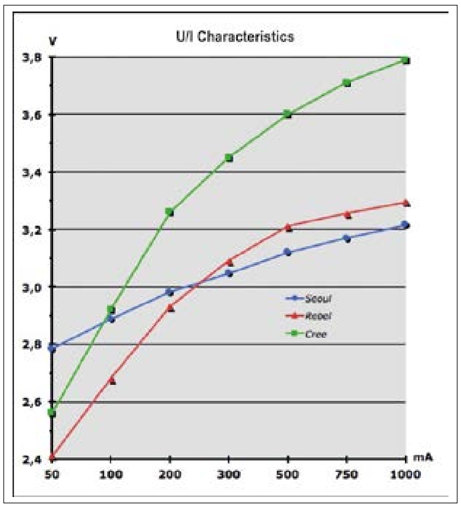


Figure 1. Current versus voltage characteristics of three different 3-watt power LEDs.

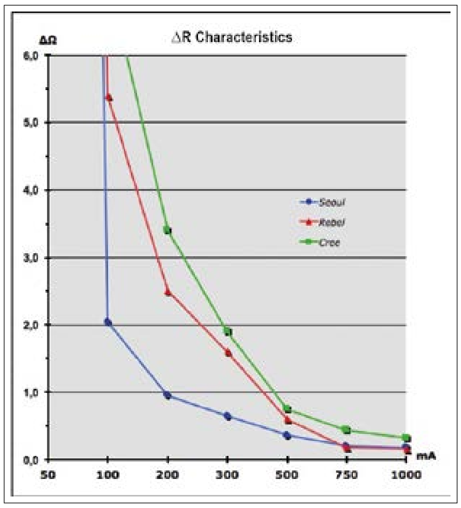


Figure 2. Dynamic resistance versus current characteristics of the same three power LEDs.

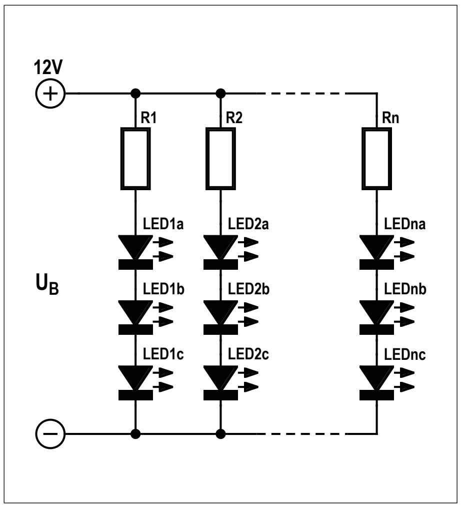


Figure 3. Basic circuit of a light chain with LEDs.

V_S of 5 V, along with a typical LED current of 20 mA, this gives roughly 150 Ω. As you can see, in this case 60% of the power is dissipated in the series resistor, which is why switch-mode constant current sources are used instead of series resistors at relatively high current levels. By the way, it is also a good idea to use series resistors on the outputs of microcontrollers to limit the current to 20 mA, as otherwise the power dissipation of the IC could be relatively high.

Q *Should LEDs be connected in parallel or in series? How many LEDs can be connected together? What do I need to pay attention to?*

A You should **never** connect LEDs in parallel, due to device-to-device differences in the forward voltage (even with devices from the same production batch). If you did so, the voltage on all the LEDs wired in parallel would be the same but the LED with the lowest forward voltage would conduct the most current. This LED would not only be brighter than the others but also hotter due to the higher current, which would ultimately cause it to fail from the overload. The situation only gets worse when you connect even more LEDs in parallel. The only feasible option is series connection. In theory you can connect as many LEDs (of the same type) in series as you want. You just need to have enough voltage left over for the series resistor that stabilizes the current. With a constant current source, the limit is set by the maximum output voltage of the current source. A drawback of a series circuit is that if a LED fails open, all of the LEDs in the series circuit go dark.

To alleviate this problem, ready-made LED chains (for example, for Christmas lights) usually consist of several series LED strings connected in parallel, each with its own series resistor. LED chains with a 12 V power supply usually have many series strings of three LEDs with their own resistors, all connected in parallel (see **Figure 3**). Although this results in higher power dissipation, it is tolerable considering the relatively low power level.

Q *What do I have to pay attention to with power LEDs?*

A To start with, everything that applies to small LEDs. In addition, series resistors are not economical with power LEDs, so constant current sources are used instead. For energy efficiency reasons, they are designed as switching regulators. A lot of specialized ICs are now available for this purpose. Power supplies with constant-current output are also readily available. Two aspects are important in this regard: the current from the current source must match the power LEDs, and the maximum output voltage must be somewhat higher than the total forward voltage of the LEDs connected in series. With ten white LEDs in series, this can easily amount to 36 V or more. By the way, you should **never** connect several series LED circuits in parallel to a high-power current source. Each series circuit requires its own current source.

Power dissipation is also an important issue: the LEDs must be properly cooled. Even power LEDs rated at 3 W like shown in **Figure 4**, are often bonded to a circuit board with an aluminum core and a heat sink is attached to the other side. It's hard to exaggerate the importance of this aspect, since the lifetime and light yield drop with increasing temperature.

Q *Are there also LEDs that can be connected directly to the AC line voltage?*

A The answer here comes from Radio Eriwan: yes, in a manner of speaking. However, these are not simple LEDs but instead modules containing the ballast electronics in addition to the actual LEDs, in order to supply a suitable constant current to the LEDs (**Figure 5**). You could buy one of these modules if you do not want to pay for an extra power supply or you are cramped for space. However, it's not a particularly low-cost solution.

You can use the circuit shown in **Figure 6** to connect small LEDs with a rated current of 20 mA directly to the AC line voltage. **However, due to the relatively high voltage you must be very careful when building this circuit and connecting it to the AC line. It is essential to observe the applicable safety precautions. This circuit is not suitable for beginners!**

(150489-I)

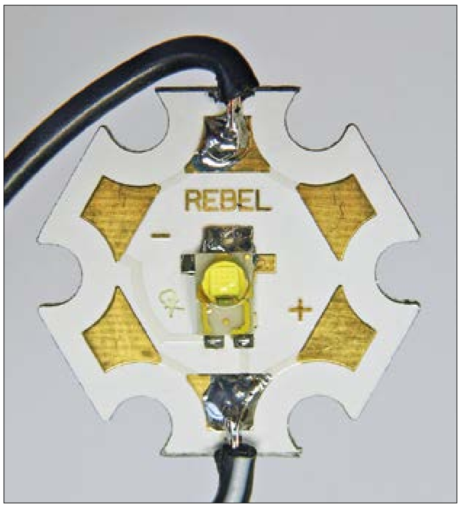


Figure 4. A 3-watt power LED on a circuit board with an aluminum core for heat dissipation.

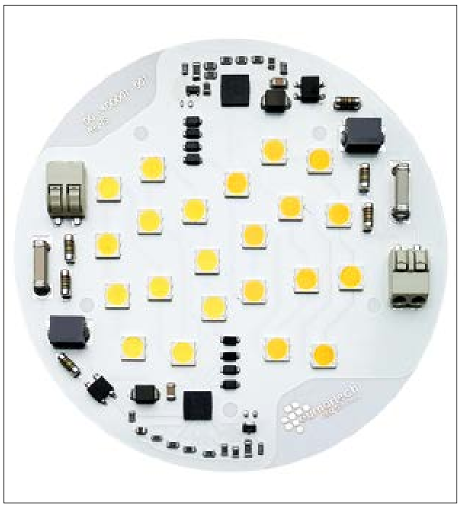


Figure 5. A Eurnortech LED module for direct connection to the AC power line.

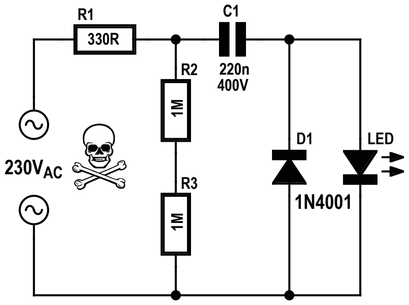


Figure 6. One way to connect a small 5-mm or 3-mm LED directly to the 230-V AC grid. Observe the applicable safety precautions!

Visit Elektor at Embedded World 2016

Elektor International Media will be an exhibitor at the Embedded World show in Nürnberg, Germany, on February 23–25, 2016. The Embedded World trade fair is all about security for electronic systems, distributed intelligence, the Internet of Things, e-mobility and energy efficiency, and of course the booming market called embedded systems.

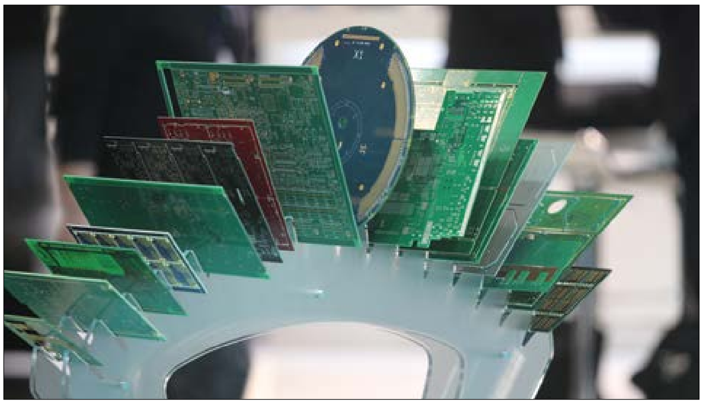


Photo: NürnbergMesse / Frank Boxler (2015)

embedded world
Exhibition&Conference
... it's a smarter world

On Elektor's 35 m^2 floor space you have a unique opportunity to meet & greet editors, lab engineers and publishers, and see live demos of upcoming and published projects like **Lumina**, **Wireless Quiz Buttons RGB Style**, **6-Digit Nixie Clock**, **DDS Function Generator**, and **eRIC Nitro**. Elektor's presence is not limited to **stand 4A-518** though as our **Elektor TV** and **Elektor Business** crews are out and about on the show grounds to spot and cover hot topics.

Breaking news on special offers, activities, and demos planned for the show will be posted in the Elektor Newsletter sent to Members on Fridays, so stay tuned. We kick off here with free access to the show. CU!

Mark the Date:	**February 23–25, 2016.**
Mark the Place:	**Exhibition Centre Nürnberg, Hall 4A, Booth 518.**
Mark the url:	www.embeddedworld.de/voucher, code: B319128

EAGLE Version 7.5

CadSoft announce that EAGLE Version 7.5 has just been released with new features and version 7 upgrade savings. EAGLE V7.5 includes Fusion 360, an integrated ECAD-to-MCAD workflow in partnership with AutoDesk, and PCBSim, an integrated simulation and signal integrity tool powered by Felicitas.

Version 7.5 is said to have improved compatibility through Gerber Import; Import & Export of DXF files; and incorporation of a number of other popular import and export ULP extensions. It also features usability enhancements like scalable icons to support retina displays, file locking, and various small improvements and fixes.

www.cadsoftusa.com (150568-3)

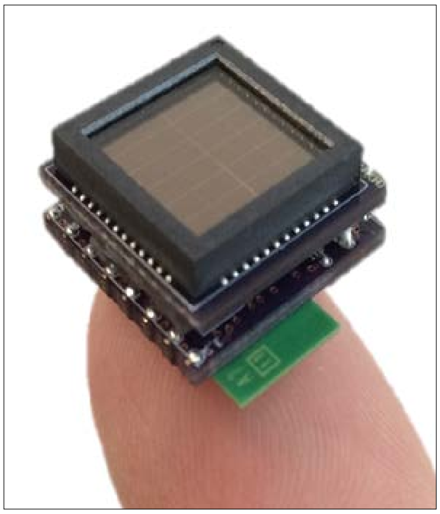
World's Lowest-Power, Battery-Free Energy Harvesting PMICs

Cypress Semiconductor's new family of Energy Harvesting Power Management Integrated Circuits (PMICs) enables tiny, solar-powered wireless sensors to be made for Internet of Things (IoT) applications. Cypress claim their devices to be the world's lowest-power, single-chip Energy Harvesting PMICs, for use with solar cells as small as 1 cm^2.

The new PMIC devices are fully integrated, making them ideal for battery-less Wireless Sensor Nodes (WSNs) that monitor physical and environmental conditions for smart homes, commercial buildings, factories, infrastructure and agriculture. A complete, battery-free Energy Harvesting solution is available that pairs the S6AE101A PMIC, the first device in the new family, with the EZ-BLE™ PRoC™ module for Bluetooth® Low Energy connectivity, along with supporting software, in a $49 kit.

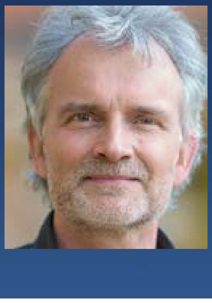
The new PMICs feature startup power of 1.2 µW — 4x lower than the nearest competitor — and consumption current as low as 250 nA, maximizing the power available for the sensing, processing and communications functions of a target application. The fully-certified, small-form-factor EZ-BLE PRoC module works with the PMIC devices to contribute to the low power and ease-of-use of an energy harvesting system solution.

www.cypress.com (150568-1)



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

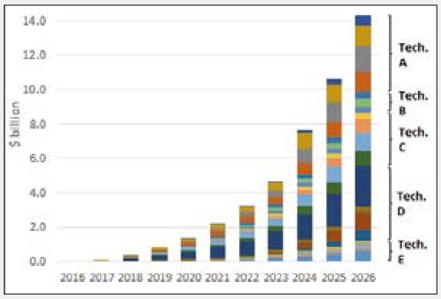


$14bn Market for Advanced and Post Li-Ion Batteries in 2026

Whilst existing battery technologies such as lead acid and lithium-ion batteries will dominate the battery market in the following 10 years, advanced and post lithium-ion batteries given the right conditions can potentially take a considerable dent of about 10% of the whole battery market by 2026. This is mostly because some of them are already in use in niche segments that will present high growth in the following 10 years. Indeed advanced and post lithium-ion battery technologies can reach a market value of $14 billion in 2026. These forecasts and detailed analysis is found in an IDTechEx Research report "Advanced and Post Lithium-ion Batteries 2016-2026: Technologies, Markets, Forecasts".

One interesting point in the development of these technologies is that they are not targeting the established end-use markets as entry point, such as automotive, but emerging niche markets. Indeed whilst automotive remains one of the main markets for battery technologies, 2015 is seeing the emergence and consolidation of alternative niche markets in which new battery technologies can find spaces for development without being exposed to fierce competition from incumbent companies and their technologies.

These niche segments are allowing these emerging technologies to mature and the incumbent companies to reach the scale and the security of supply that the automotive industry and consumer electronics require.

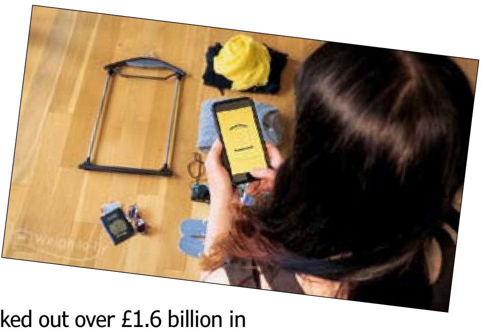


www.IDTechEx.com (150541-1)

Kickstarter: Revolutionary Smart Luggage Scale

Weigh-to-Fly is an extendable luggage scale which sends real-time updates to your phone as you pack your case. It is the ideal travel companion to help you avoid excess baggage charges when you fly.

The device has an adjustable, lightweight, telescopic frame that slides and locks to fit whichever size case you want to take. Once the case is placed on top, the sensors in each of the corners send updates via Bluetooth to the traveler's smartphone, allowing them to see instantly how close they are to their weight limit as they pack. The smartphone app contains details of every major airline's weight restrictions so the user simply types in which airline they are flying with. It even takes account of the weight of the device (450 g) and the scale's unique "coat hanger" design means it's also useful when you get to your destination. Also incorporated is a proximity sensor, which sends a notification to your phone if you leave your case behind or move too far away from your luggage.

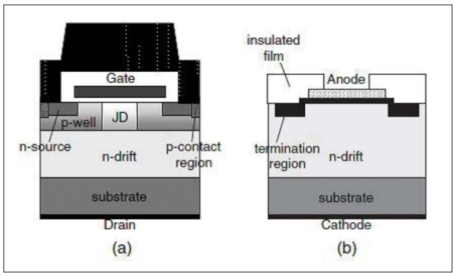


Weigh-to-Fly is the brainchild of Software Electronics Engineer Jamie Cruickshank: "One in five passengers are charged an average of £25 for excess baggage at the airport. In the UK alone, passengers forked out over £1.6 billion in charges in a single year. Multiply this globally and it is a huge problem. Not only is it costing passengers dearly but it causes stress, hassle and inconvenience at the airport." Some of the rewards included in the £50,000 funding / £69.99 rrp Kickstarter crowdfunding campaign are early bird deals and a chance to visit the design team to collect your product in person.

(150541-2) www.weigh-to-f.ly/kickstarter

Power Electronics: Silicon Carbide Gains Traction

Research reported in the Japanese Journal of Applied Physics (JSAP) by researchers at Mitsubishi Electric Corporation describes the development of a new power module made from a SiC metal-oxide-semiconductor field-effect transistor and a SiC Schottky barrier diode. The team successfully trialed the module in a train traction inverter — a device used to convert the direct current from the power source to three-phase alternating current suitable for driving the propulsion motors — with promising results. For a power module in a traction inverter, low power loss, miniaturization, high voltage rating, and high temperature environmental resistance are required. The team prepared the SiC MOSFET for the power module by n-type doping the junction FET region: this reduced on-resistance of the device at high temperatures. By combining the SiC MOSFET with a SiC SBD — a diode which allows for fast and efficient switching — a power module was created for a traction inverter rated at 3.3 kV / 1500 A.



A new traction inverter system equipped with their power module is stable, highly efficient and reduces switching losses by 55% compared with conventional silicon-based inverters.

(150541-3) http://jsap-bulletin.jsap.or.jp/en/

Addressing Embedded Platform Performance Bottlenecks for USB 3.0

By **Abhishek Gupta** (Cypress Semiconductor)

High-definition (HD) and ultra high-definition (UHD) video camera designs can illustrate the many benefits of using a standard serial bus such as USB 3.0. Today's HD/UHD cameras generate real-time data that must be captured, processed locally, then quickly transferred to remote devices such as security monitors and pooled storage services. Even a 5-Megapixel camera at 24 frames per second (fps) can generate a 2.4-Gbps data flow. Designs relying on older serial interfaces such as USB 2.0 and Wi-Fi 802.11n simply cannot keep up with camera data, let alone handle real-time transfers to other devices within the user network. Lack of high-speed data paths can lead to higher embedded design complexity and cost, such as upsizing frame buffers and requiring larger local storage elements. These design elements can increase overall cost and development times.

USB 3.0 provides an extremely effective choice for today's high-performance embedded systems developers. Its dual-bus architecture allows for communication with legacy devices while SuperSpeed USB can achieve device transfer rates of 5 Gbps. In addition, USB 3.0 has 3X more power efficiency than USB 2.0. Due to its broad industry popularity, SuperSpeed USB is a natural choice for both internal and external data path management with the best combination of high performance, reduced power consumption, and cost structure.

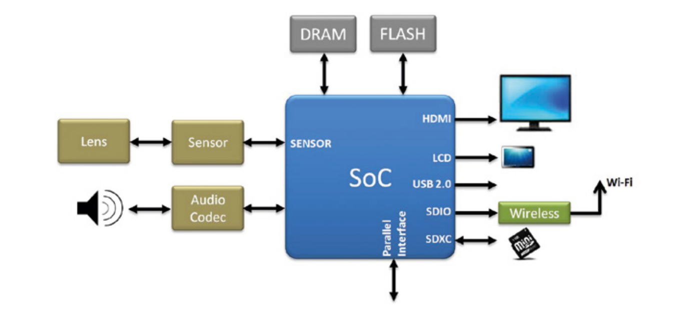


Figure 1. Video camera architectural block diagram.

Table 1. Data transfer times for a 30-minute 1080p HD (H.264) video		
Interface	**Transfer Rate**	**Transfer Time**
Wi-Fi 802.11n Optimized	300 Mbps	8 minutes
Wi-Fi 802.11n Typical	80 Mbps	30 minutes
Wi-Fi 802.11g	54 Mbps	45 minutes
Wi-Fi 802.11b	11 Mbps	3 hours 39 minutes
USB 2.0 Optimized	320 Mbps	7 minutes 30 seconds
USB 2.0 Typical	200 Mbps	12 minutes

Camera system-on-chip (SoC) architectures are highly tuned for gathering and processing data from high-resolution image sensors using specialized on-chip video and image processing DSPs. Design implementations such as those shown in **Figure 1** use an SDXC interface for local video/image storage, typically in the form of mini/micro SD Cards. HDMI ports provide external video transfer to external monitors. The SoC must also manage the on-board LCD screen. USB 2.0 and Wi-Fi ports can provide for external data sharing. Unfortunately, these SoC interfaces cannot keep up with the data rates generated by the image sensors. Custom SoC solutions can lead to long development times due to their design complexity.

Wanted: bandwidth

Certain product categories such as commercial security cameras provide limited on-board storage. They instead rely on fast external interfaces such as wired or wireless LAN (Wi-Fi) to transfer their video streams onto a local server. Some consumer cameras provide removable storage card (mini/micro SD). However, they tend to not be equipped with a local LCD screen for local content viewing. Thus they depend on the recorded content to be transferred to an external device for storage and viewing. These devices often utilize a variety of popular interfaces such as Wi-Fi, USB 2.0, or Bluetooth for data transfer.

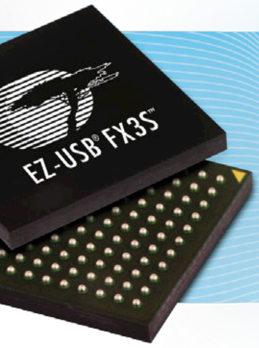
Wi-Fi is highly dependent on the availability of a fast reliable network infrastructure. Even an optimized 802.11n network may only be capable of supporting a 300-Mbps data rate.

USB 2.0 has an effective bandwidth of only 280-320 Mbps (35-40 MB/s). Even so, most devices can only deliver 200-240 Mbps. Sustained USB 2.0 transfer rates depend heavily on software driver and platform optimizations.

Table 1 illustrates the expected data transfer time for a 30-minute 1080p HD (H.264) video of ~18 GB using USB 2.0 and various Wi-Fi alternatives. Notice that some cases take longer to transfer the video stream than viewing the video stream itself requires!

Camera OEMs choosing to address this performance bottleneck are left with two choices: (1) Develop their own next-generation ASIC or (2) Wait for next-generation SoCs.

Either choice can easily result in an 18- to 24-month product introduction delay. Alternately, designers can upgrade their existing platforms using off-the-shelf USB SuperSpeed components. SuperSpeed USB can operate 10 times faster data than USB 2.0, making it an ideal serial interconnect to address the video transfer bottleneck. Off-the-shelf controllers such as the Cypress EZ-USB® FX3STM SuperSpeed USB Controller [1] provide on-board ARM9 core, USB 3.0 functionality, plus two storage ports (configured either as SDIO 3.0 or eMMC 4.41) as shown in **Figure 2**.



As a high-speed serial interconnect, USB 3.0 (SuperSpeed USB) has found widespread usage among personal computers plus peripheral devices such as video monitors and storage peripherals. Embedded designers are finding that highly integrated USB 3.0 devices can address their high-speed internal data transfers. USB 3.0 can also provide a low-cost interconnect to other devices, along with faster data throughput, lower power, smaller design footprint, reduced BOM cost, and shorter development schedules.

FX3STM deployment

The FX3S can easily be integrated into a video camera platform by connecting the FX3S to the camera SoC via its programmable GPIF interface (**Figure 3**). The Cypress FX3S GPIF II interface is a fully configurable parallel interface that connects to an external ASICs/SoCs or FPGAs. In this configuration, the FX3S and its associated SD Cards merely appears as a USB storage device to the SoC. The compressed video data can be stored in the same manner as the current camera implementations without any impact on the software stack.

This implementation produces several architectural advantages. First, it provides a low-risk, economical means of adding a SuperSpeed USB port. Second, it moves SD Card management away from the SoC, releasing its CPU bandwidth for other critical tasks. The FX3S GPIF II, configured as a 16-bit multiplexed address/data bus running at 100MHz, provides an ~1.8× data WRITE (Figure 3, arrow 1) performance improvement over current implementations.

The dual-SD interface on the FX3S can be configured in a RAID 0 configuration using a second mini/micro SD Card, improving performance while providing end users with increased storage. The FX3S can deliver a video READ throughput (Figure 2, arrow 2) of ~720 Mbps for a dual SD card configuration. This reduces the transfer rate for the same 18-GB video to less than 4 minutes, delivering a significantly higher performance over existing architectures. With the increased size of storage in cameras, longer recording times and transition to higher-resolution video qualities such as 4 K, this performance gap will continue to widen significantly. The addition of this particular controller does not significantly impact battery life. The FX3S is expected to add ~97 mW of active power to the total camera platform power consumption. Assuming a 1160-mAH, 3.8-V camera battery; the FX3S would consume an additional 13.43 mAH (assumes a 95% conversion efficiency) of energy for the same 30 minute video or ~1.2% impact on the battery life. Adding a second SD Card (~270 mW) for higher capacity and better performance will impact total battery life by ~4.3%.

Conclusion

SuperSpeed USB is a natural choice for embedded applications that require high-speed device data transfer connections. Developers are left with the choice of waiting for the next generation of SoCs and take the associated schedule and opportunity cost risk or evolve their platforms around commercially available solution to meet the address this platform bottleneck. Powerful devices such as the Cypress FX3S can reduce the video image transfer times by a factor of 4-10 while only minimally affecting the expected battery life. ⏮

(150466)

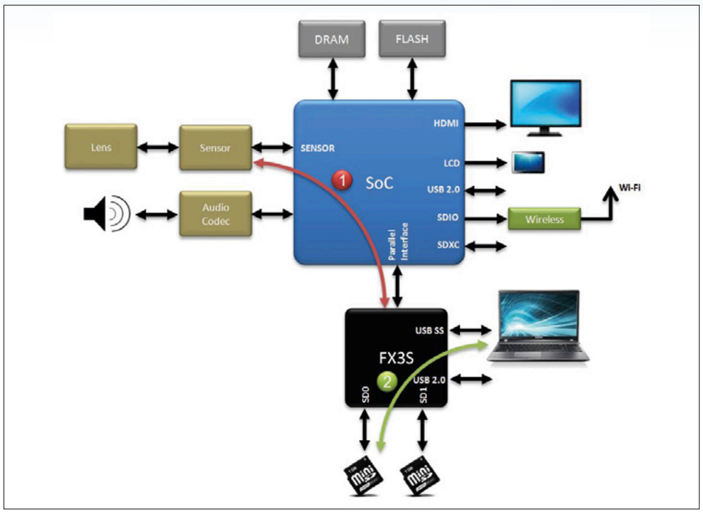


Figure 2. Cypress FX3S SuperSpeed USB 3.0 controller block diagram.

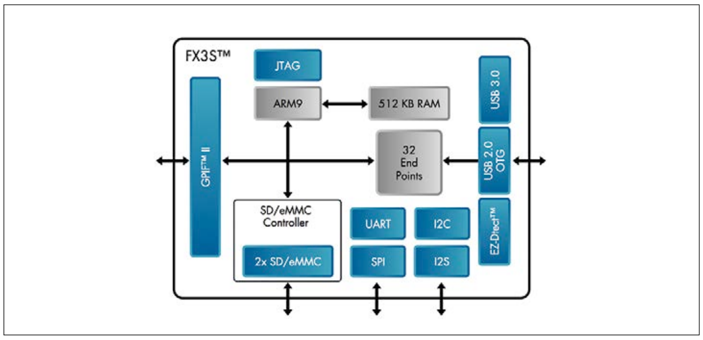


Figure 3. Enhanced video camera architecture using Cypress FX3S.

Web Link

[1] EZ-USB® FX3STM SuperSpeed USB Controller: www.cypress.com/?docID=45991

The Author

Abhishek Gupta is a business analyst for Cypress Semiconductor. He has a B.E. in Electronics & Communications from Maharishi Dayanand University, Haryana, India. He has worked with Agilent Technologies as a Logistics Coordinator (RoHS Specialist) and can be reached at agni@cypress.com

Welcome to Elektor Labs

Elektor Labs is the place where projects large, small, analog, digital, new and old skool are sketched, built, discussed, debugged and fine-tuned for replication by you.

Our offer: Become Famous

Most engineers and budding authors we come across are just too modest. If they do not see the attraction and beauty of a design idea scribbled on a coaster and worked out later at home, others may, and should. Let Elektor Labs help you hone your design to perfection, let the editors assist with text & graphics, and reap the rewards by seeing your name in print in Elektor magazine's celebrated LABS section. Sure, we are happy to negotiate payment, but the actual remuneration will be fame & glory in electronics land, and your name added to the long list of extremely successful e-authors. Our get-u-famous formula also applies to book authors, bloggers and video makers. Students and youngsters: being in publication is a current boost like no other in getting a job!

Our Experts and Designers

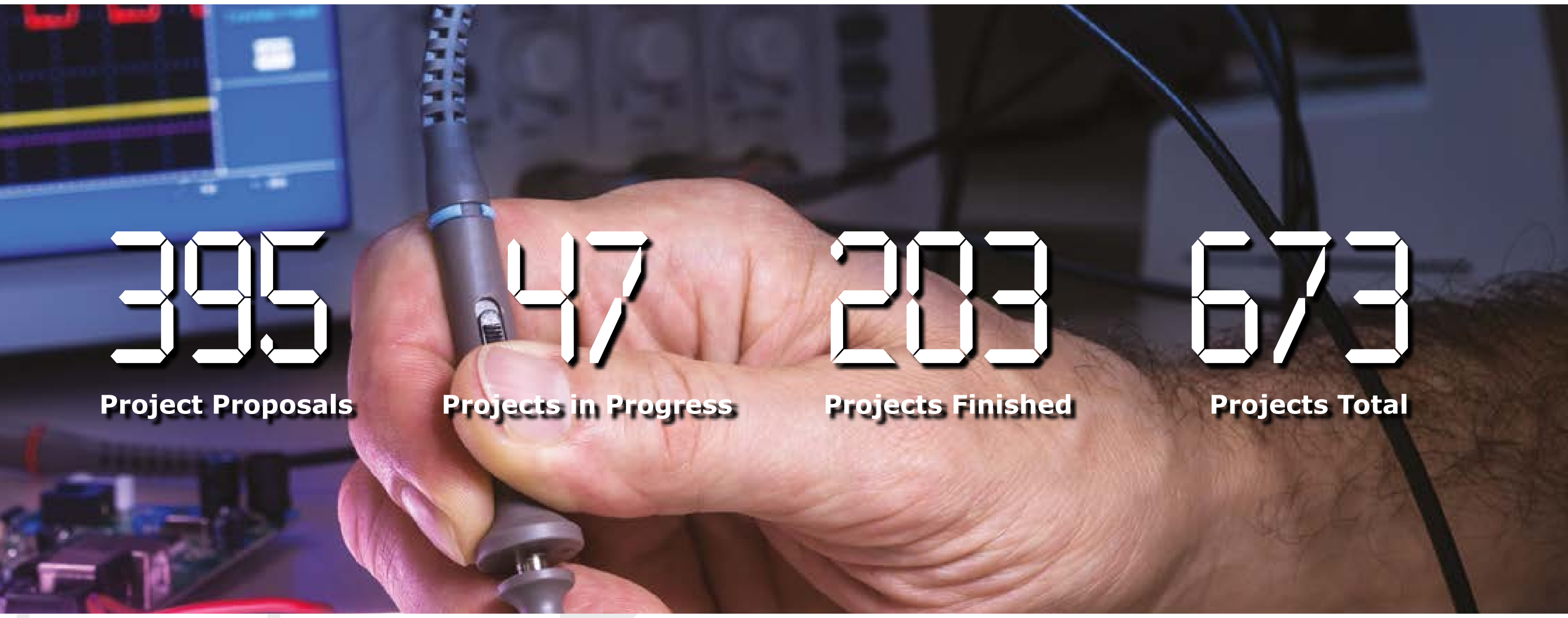
Besides experienced support staff and BSc, MSc qualified engineers with a total working life in electronics of about 200 years, the Lab has access to Elektor's vast network of experts for consultation, critical advice, and assistance with specialized assignments.

Our Facilities

We are sumptuously housed in three spacious rooms at Elektor House where we try unsuccessfully to keep our solder jobs and prototypes off our computer desks. We have water, 218 volts electricity and coffee nearby. PCB milling, prototype assembly, SMD reworking, audio testing, pizza cooking, and mechanical work are deferred to converted cellars in the basement.

Our Standards

All projects and products going through the Labs pipeline are produced to high engineering standards. In practice, prototypes of projects labeled LABS in the magazine must be demonstrated to work to specification on certified, calibrated test equipment available locally. BOMs and schematics must match perfectly. Kits are sampled for completeness. We are ROHS compatible, lead free and comply with electrical safety standards applicable in our location. If engineering errors are found these will be put up for public notification.



395 Project Proposals

47 Projects in Progress

203 Projects Finished

673 Projects Total

Our Products

Our products are visible in Elektor magazine, as well as on the Elektor.Labs and Elektor Store websites. The range includes text write-ups for editors, prototype photography, PCBs including SMD-prestuffed, PCB files, project software, programmed devices, semi-kits, tools, modules, videos, and service desk information.

Our Webinars

The more talkative of our Lab engineers do not stop at testing prototypes, they happily share technology related problems, insights, get-u-going information, and design skills on live camera at Elektot.tv. Labs' webinars are free to attend and extremely interactive. They are announced in Elektor.POST, and webcast live from Elektor House in Holland. Do plug in!

Our History

The origins of Elektor Labs go back to the early 1970s when soldering and writing was a one-man, single-desk job. Ove the years Labs staff have not only witnessed the arrival of the transistor, the IC, the microcontroller, and the SMD, but actually jumped on these parts as soon as they were out of the professional-only woods. Once special branches, Audio Labs, Micro Labs, PCB Labs, and Mechanical have converged back again into a single activity.

Upload your own projects!
On our very own Elektor-Labs website, you can share and showcase your ideas and project proposals to thousands of other electronic engineers. The site also allows you to follow other specialists' activities, supply comment, and so push the projects forward. Here's the best part, the top projects are eligible for processing in our test lab, in preparation for publication in Elektor magazine.

Who's this for?
Although only members can log on to our Elektor.Labs website to publish projects and contributions, anyone can view along with the other projects. If you'd like to see your project published in Elektor magazine, which is published in four languages and read by tens of thousands of electronics specialists all over the world, then become a Green of Gold member (**www.elektor.com/member**) or log in as an existing member of our Elektor community!

Welcome to the DESIGN section

By **Clemens Valens,** Elektor Labs

The fork — more dangerous than a knife

Open source is great. Over the past decade enormous amounts of software — and hardware — have been produced and the associated source files have been made available for anyone to enjoy, use, learn from, modify, etc. The open source movement has undeniably moved the electronics industry forward by rendering technologies accessible to anyone interested, and that's absolutely fabulous. But it could have been so much better. The llustration accompanying this column attempts to visualize the known "forks" (splits, branches) of Linux up to 2013. All these flavors were created by people who thought that they had a better idea of what Linux should be. The amount of time and energy spent on maintaining and improving all these spin-offs is mind boggling. If only the time had been spent on a single distribution, Linux would have been the (best) operating system of the Universe.

The Bible has a story about Linux, in the Old Testament even, which shows how old Linux really is. Originally Linux was called Babel and many people contributed to it. But when it became so powerful that it veered on being usable, God decided to fork it and we know the result.

Last year (2015) we witnessed a major Arduino fork that created arduino.cc and arduino.org. The Bible also tells us how this eventually will end: *So the Lord scattered them abroad from there over the face of all the earth, and they ceased building the system.*

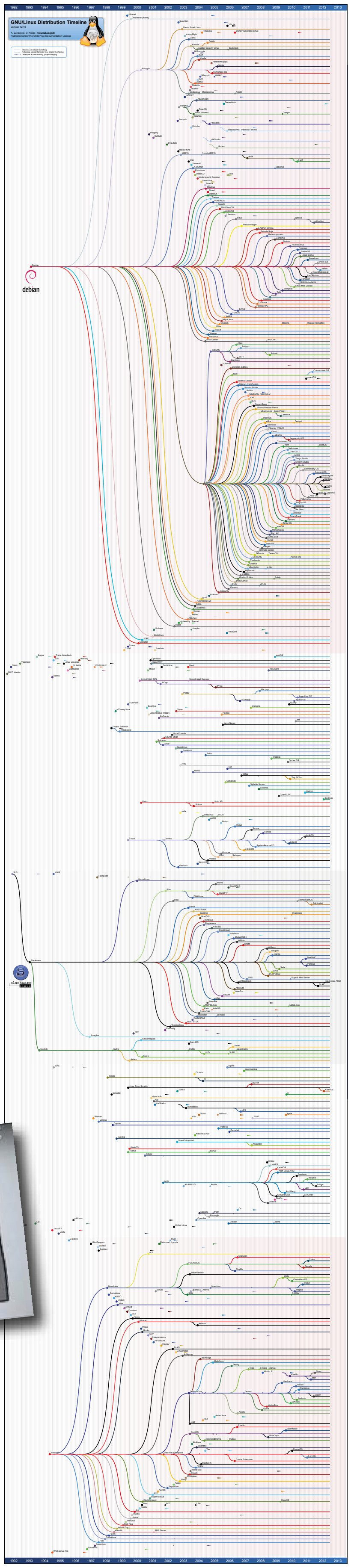
en.wikipedia.org/wiki/Fork_(software_development)

10^{-24} Linux

Meanwhile the industry got fed up with all the Linux forks and so it did what it does best: it created a workgroup. The Yocto Project, as this workgroup is called, has several goals. One of them is to encourage standardization and reuse of existing components for Linux platform development. Another one is to educate the community about the benefits of this standardization and collaboration. Like Linux, the Yocto Project is an open source collaboration project. Will they succeed or are its odds yocto? I will now take bets on the date of the first Yocto fork. ⏮

www.yoctoproject.org

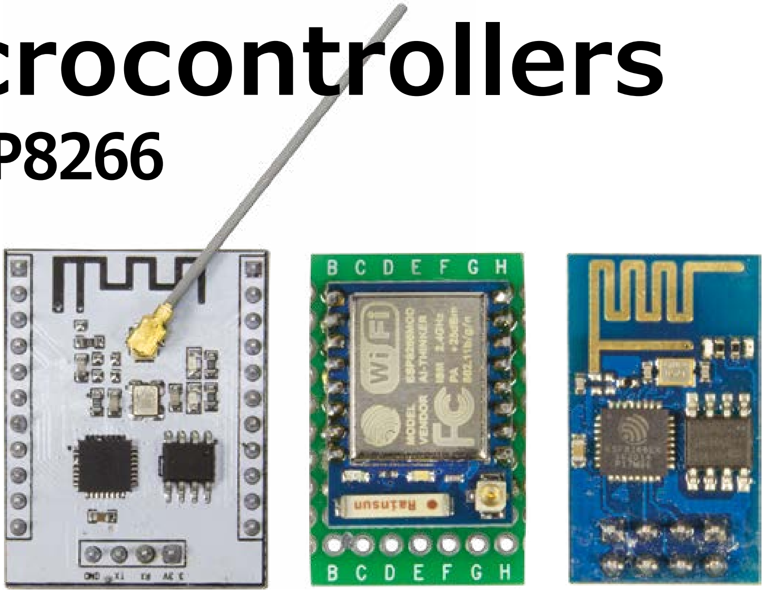
(150559-I)



WLAN for Microcontrollers

Take control with an ESP8266

By Walter Trojan (Germany)



Taking your first steps into the Internet of Things, where your refrigerator talks to the supermarket and the heating responds to a smartphone call may seem a bit daunting for many, but don't despair, the hurdle is not as high as it seems. A low cost WLAN board together with a standard microcontroller plus some software is all you need to build this project which allows you to control LEDs from a PC, tablet or smartphone.

SP8266 Main Features

• Processor	Tensilica L106, 32-bit
• Processor clock	80 / 160 MHz
• RAM memory 3	2 + 80 KB
• Flash memory	Max. 16 MB, typically 512 KB to 1 MB
• Operating Voltage	3.3 V, **not** 5 V tolerant
• Current	10 µA to 170 mA
• GPIO port count	17, including I^{2}C, SPI, UART, PWM multiplexed
• Analog port count	1, 10-bit resolution, input range 0 to 1 V
• WLAN standard	802.11 b/g/n/....
• Simultaneous TCP channels	Max. 5
• Supported functions (Modi)	Station, Accesspoint, Station+Accesspoint

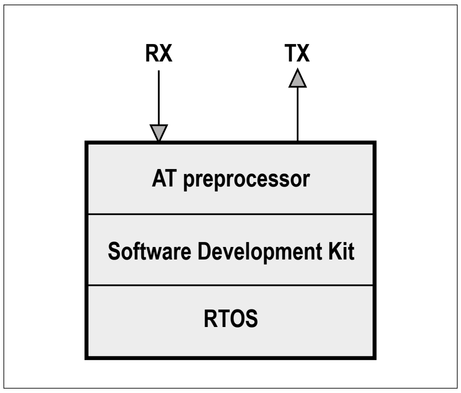


Figure 1. The ESP8266 firmware structure.

Connecting a microcontroller to a WLAN is not a trivial task but thanks to a relatively new chip called the ESP8266 the workload is drastically reduced. Here we will show how you can use this chip as a WLAN front end for microcontrollers such as the Atmel, PIC or ARM family of devices.

The ESP8266...

... is a product developed by Expressif [1] from Shanghai and has been on the market since the beginning of 2014. A System-on-a-Chip (SOC), it combines a 32-bit microprocessor with a WLAN transceiver. There are a number of similar breakout boards available which use this chip; they also have flash memory (0.5 or 1 MB) and a 2.4 GHz Wi-Fi antenna included on the board. The title picture shows one of these breakout boards, they all use the same chips but with slightly different GPIO configuration or antenna and usually retail for around 3 euros. Their main features are listed in **Table 1**.

Table 1. Simple application protocol.

	Command	Action
Button 1	B1, Lof	Red LED off
Button 2	B2, Lon	Red LED on
Button 3	B3, Lbl	Red LED pulsing
Button 4	B4, Lzz	Red LED flash
Slider 1	S1,<value>	Blue LED brightness according to <value>
Slider 2	S2,<value>	White LED brightness according to <value>

The ESP8266 can be used as a station, access point or station + access point. As a station it works as a TCP/UDP server or client, as an access point it forms its own WLAN for neighboring WLAN nodes.

Figure 1 gives a rough outline of the firmware structure. The multithreading real-time operating system freeRTOS takes care of program sequence control. Many of the functions are already available in the Software Development Kit (library). The ESP8266 comes pre-programmed with standard firmware which provides an AT preprocessor for serial communications so you can start communicating with it straight away. The AT command set goes all the way back to the days of the Hayes modem (those with grey hair will surely remember these) and contains command strings with which much can be accomplished. For

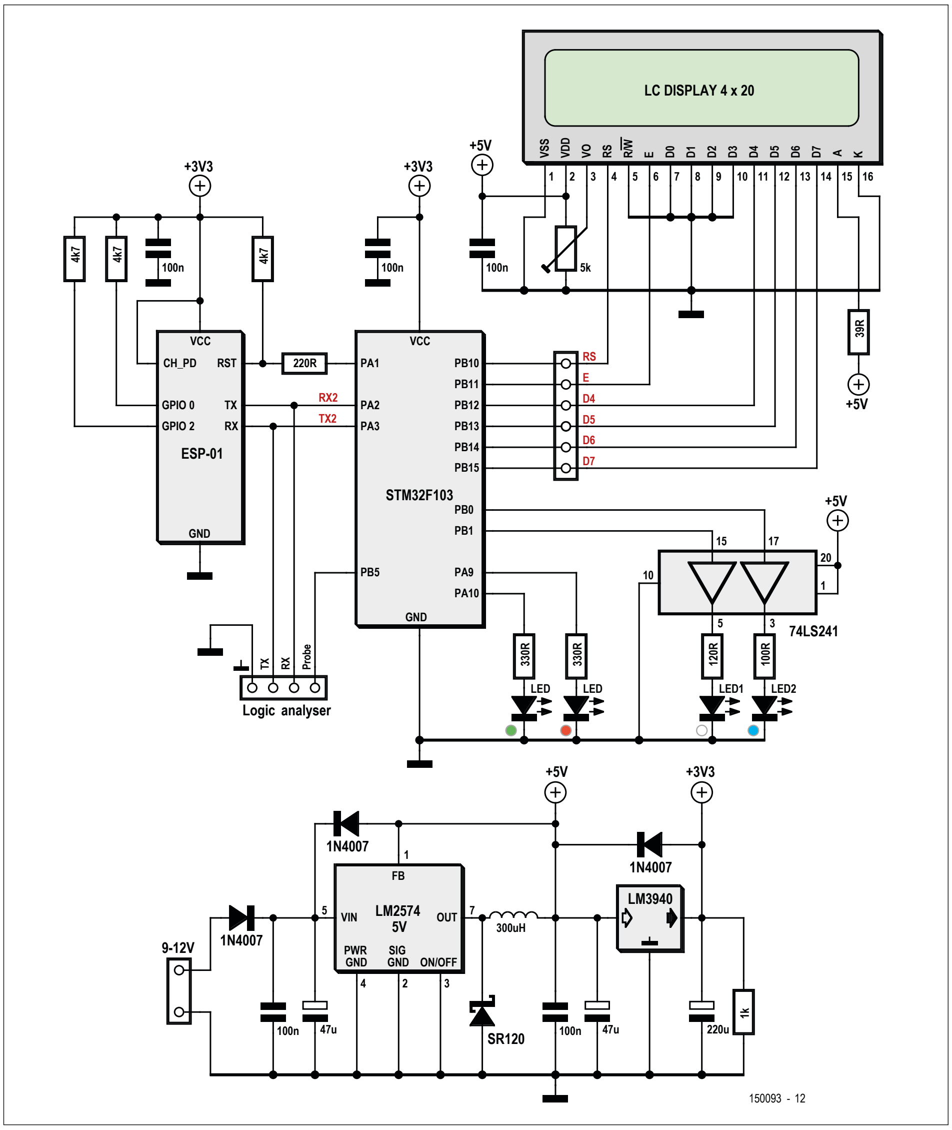
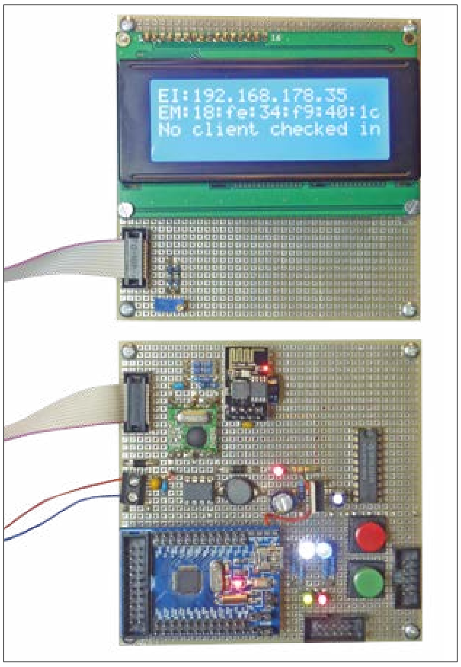


Figure 2. The project schematic.



example, the character string 'AT+CWMODE=1' switches the ESP8266 into Station mode. A UART interface is provided on the breakout board so that the AT commands can be sent serially to the processor.

In the context of this article we will exclusively be using the AT command set. It is also possible to replace the AT preprocessor with your own software interface or make the ESP8266 a completely autonomous control unit without the need for an external processor. The firmware to achieve this can then be loaded into the flash memory using the boot loader via the UART interface.

Figure 3. The test board.

The task

In this introductory project (Software-Download see [7]) we use an ESP8266 as a WLAN Frontend for a STM32 microcontroller. For practical purposes it falls into the 'Hello World' category of applications. It can switch three LEDs (white, blue and red) connected to the ST controller. The commands can be generated by different clients (PC, Android tablet or Apple iPhone) and are relayed over the WLAN. The client user interface consists of four buttons and two sliders (to dim the blue and white LEDs). The author has implemented some commands (along with their actions) shown in **Table 1**.

<value> can have any value in the range 0 to 255 (0 = off, 255 = fully on). When one of the control elements is activated the client sends the corresponding command over the WLAN, which will be executed immediately in the test platform. The STM32 responds with an 'ack\' character sent over the WLAN.

A green LED on the board indicates the WLAN status.

- off = No connection to WLAN
- blinking = Connected to WLAN, but currently no client
- on = Client is connected

The test platform

Figure 2 shows the test platform schematic. The author was able to re-purpose an existing development board containing an STM32 processor PCB and a 5/3.3 V voltage regulator that he had hand-wired for a previous project. Two 10-way connectors provide connections to a 4x20 LC display and a logic analyzer. The neat ESP8266-01 board was quickly integrated onto the board. The finished construction can be seen in **Figure 3**.
An ARM controller type STM32F103C8T6 is used for sequence control. It is clocked at 72 MHz and has 64 KB Flash memory and 20 KB of RAM. This 32-bit powerhouse has lots of power in hand for this relatively undemanding task, any 8-bit Microchip or Atmel controller could have done the job just as well.

Diagnostic information and connection status is shown on the 4x20 LC display. The standard 4-bit data with two control signals from the controller's port B are used to interface to the display. A logic analyzer can provide information by listening to the Rx and Tx traffic exchanged between the controller and the ESP8266. A Probe output signal from a GPIO pin can be toggled by the Probe signal subroutine which produces a pulse each time it's called. The Probe routine can be called anywhere in the program and gives an indication when a process is executed in relation to other events. The author has found this useful for quickly identifying the source of programming errors.

Table 2. Initializing the ESP8266

STM → ESP	ESP → STM	Action
Reset pulse	Unreadable reply	Hardware reset
AT+RST	OK.....ready	Soft reset
AT+CWMODE=1	OK	Station mode
AT+CIPMUX=1	OK	Connect to multiple TCP
AT+CWQAP	OK	End connection TerWLAN
AT+CWJAP=<ssid>,<pw>	OK	Start new WLAN connection
AT+CIFSR	<STAIP><STAMAC>OK	The station's IP and MAC address
AT+CIPSERVER=1,3333	OK	Start TCP-Server, Port 3333

The green LED signals the WLAN status and the red LED is a controllable Object connected directly to the STM32 output. The blue and white LED both have a higher forward conduction voltage and won't operate reliably at 3.3 V. We also operate them with a relatively high current of 20 mA which is getting close to the maximum output current (25 mA) available from a port pin. A good old 5-V 74LS241 buffer has therefore been used to drive these two LEDs.

The ESP8266-01 has only eight connections and is relatively easy to integrate. Communications to the STM take place over RX/TX of UART 2 with a transfer speed of 115,200 bit/s. The reset pin is connected to port PA1, so that the controller can selectively issue a hardware reset. The chip select input connects to V_{dd} = 3.3 V to ensure the ESP is permanently selected. The small ESP-01 PCB has two GPIO ports 0 and 2 available. Both of these are used to define the boot address at power-on. The ESP8266 can be booted from different sources (from flash or UART). In this application we boot from flash memory so both pins must be high.

It's worth emphasizing here that the ESP is **not** 5-V tolerant. It will quickly die if you attempt to power it from a supply higher than the recommended maximum. Apart from this the power supply to the board is not critical. 5 V is required for the LCD and the buffer driving the blue and white LEDs while 3.3 V is used for the STM and the ESP. The test board uses a switched regulator to produce 5 V followed by a low drop-out linear regulator for 3.3 V. When the ESP switches between sleep and active mode its supply current changes from 10 µA to 170 mA which is quite a big difference so it is important to ensure the power supply can cope with the necessary current and produce a stable output. A recommended mains power supply will have an output in the range of 9 to 12 V at a current rating of 0.5 to 1 A. An electrolytic capacitor with a value between 100 and 220 µF fitted to the input supply of the ESP will suppress current spikes and other unwanted effects.

The STM and ESP dialog

Now we can begin with the firmware development. Scheduling is performed by the STM32 which sends AT commands to the ESP8266. **Table 2** shows the dialog that occurs when the ESP8266 is **Initialized**.

The STM32 initializes the ESP8266 with AT commands and the ESP8266 responds with either OK, ready and/or information. After Start a hardware reset pulse is generated (active low, 1 ms). Then follows a Soft-Reset and the ESP responds with some system information and finishes with 'ready'. Using AT-CWMODE and AT-CIPMUX the ESP is configured as a Station with multiple connections. The ESP will automatically log on to the last connected WLAN, for security the old connection should be closed with AT-CWQAP and a new one opened with AT+CWJAP. <ssid> and <pw> are the log in credentials for the WLAN. In response to the AT+CIFSR command the ESP responds with its IP- and MAC-Address, which the MCU writes to the display. Now the TCP-Server is started with AT+CIPSERVER on Port 3333 and the ESP is ready to receive. The **Dialog with a Client** proceeds as shown in **Table 3**.

When a client is logged onto Port 3333, The ESP signals this with 0,CONNECT. The '0' indicates the channel number. Five channels (0 to 4) are available. To check the client identity the AT+CIPSTATUS command will return the IP address which will then be displayed. When a

Table 3. The dialog with a Client

STM → ESP	ESP → STM	Action
	0,CONNECT	Client is logged in on channel 0
AT+CIPSTATUS	...CIPSTATUS:0,"TCP", <client IP adr>... OK	Request Client IP-Addr
	+IPD,0,6:B2,Lon...	Client sends command, 6 characters on channel 0, red LED on
AT+CIPSEND=0,4	AT+CIPSEND=0,4 >	4 characters will be sent, ESP confirms with >
ack\n	ack	Acknowledge for Client
	Wait for next command	
	+IPD,0,6:B2,Lof...	Client sends command, 6 characters on channel 0, red LED off
AT+CIPSEND=0,4	AT+CIPSEND=0,4 >	4 characters will be sent, ESP confirms with >
ack\n	ack	Acknowledge for Client

command is sent by the client the ESP passes it on to the STM.
The channel number and the payload character count is after the +IPD sequence. The character string generated by the author begins after the colon. The acknowledge character string 'ack\n' is now sent to the client. It must be initiated with AT+CIPSEND followed by the channel number and the character count to be sent. The received command is then acted on by the STM firmware. The connection remains available until the server or client closes it. In this example we have used just a small fraction of the dozens of AT commands available.

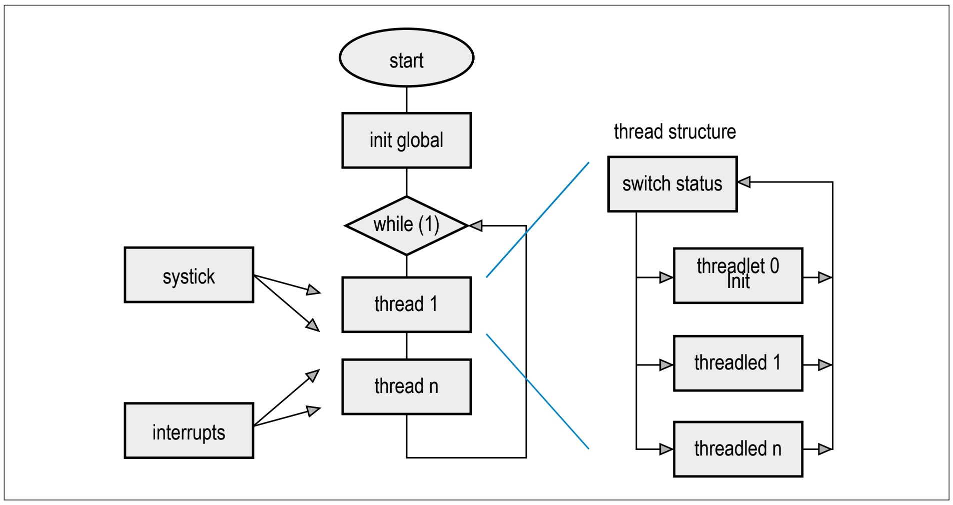


Figure 4. The STM32 firmware Structure.

The STM32 Firmware

Firmware for the STM32 was written in C using the free CooCox Version 1.78 development environment. **Figure 4** shows the structure of this software environment, the author uses this tool for most of his microcontroller projects. It functions as a cooperative multithreading environment. After initializing the globals it enters the main loop where the threads are called sequentially. These are implemented as State Machines, in which the code sequences (called threadlets here) are executed according to Status variables. When a threadlet is successfully executed it changes the Switch variable so that on the next call, the next task will be executed.

This structure prohibits delay loops and polling cycles so as not to impede the execution of other threads. Each thread contains its own Soft Timer (controlled by Systick) which can be used to produce wait loops without blocking. Using this process ensures the threads are called within microseconds, guaranteeing a good response time.
Systick is a prioritized interrupt, handled in the cyclic tasks. In this case the softtimer for the threads is incremented with a 10 ms clock. Peripheral interrupts capture time critical events, set variables and pass them on the relevant thread. Very little processing occurs in the interrupt routines.

The firmware consists of two threads: Doapp2 controls communication with the ESP, Doapp3 controls the LEDs. PWM signals are used to adjust the brightness of the blue and white LEDs. For this Timer 3 with two channels is used to generate a period of 10 ms with a 10-bit resolution. This application is based on the CMSIS libraries. It should be noted that ST have also recently introduced new development tools such as the CubeMX program generator and the HAL libraries.

Clients for PC, Android tablet and iPhone

A PC client with graphical elements allows an intuitive user interface with buttons and sliders. It was built using the free Lazarus development environment together with FreePascal [4]. Lazarus is a Delphi-compatible, platform independent IDE. Not original, but in contrast to Delphi, free!

The PC's TCP/UDP communications were based on the INET add-on component. A small test program is included which demonstrates operation as a Client or Server, IPv4 or IPv6, with or without SSL.

The author expanded on the routine, adding four buttons and two slider controls to make it suitable for use with our ESP/STM setup. Using this we enter TCP/IPv4, Port 3333 and the IP address of the ESP then press the connect button. Now we have control of the LEDs and when we are finished, use 'disconnect' to terminate the connection. The Client can be downloaded from the project website at Elektor; be sure to store the cert und pkey files in the same folder as the exe file.

Compared to the other work required for this project, the creation of the client for a mobile device was really easy. That is basically thanks to the NetIO app developed by David Eickhoff (for Android this costs around 5 euros, for iOS around 11 Euros) [5]. With this you can use the Design-Editor in a browser with intuitive drag-and-drop features. First off you define the target device and the size of

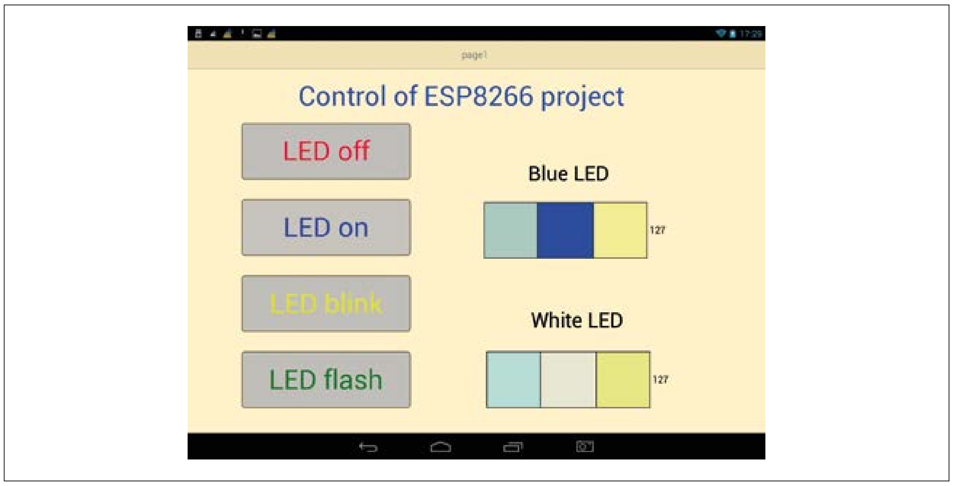


Figure 5. Operation from an Android tablet.

the user window, the global communications parameters, target IP address, port and protocol. After that objects such as buttons, sliders, text boxes labels and so forth are positioned on the surface and their parameters specified. Once complete, you can store your design in the Cloud. **Figure 5** and **Figure 6** show the user interface on an Android tablet and an iPhone.

After the app is started a swipe from left to right opens a menu where you can log in and synchronize with the cloud. Now select Design to start and you are connected immediately to the ESP server. The LEDs can now be controlled as you wish. As we said, a breeze! When the app is closed the connection is terminated.

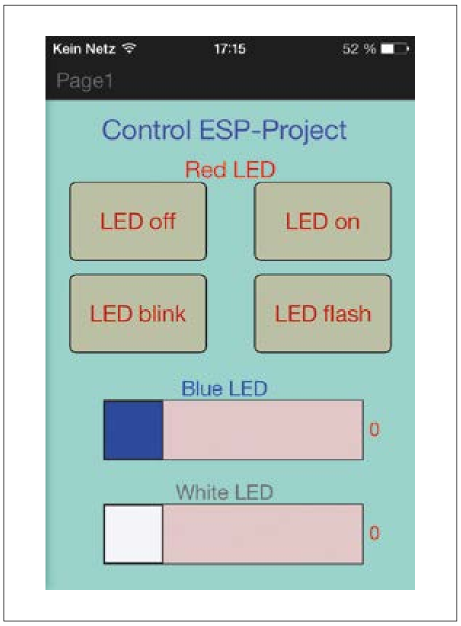


Figure 6. Operation from an Apple iPhone.

Going forward

So what have we achieved with this project? We have added WLAN connectivity to a microcontroller, got to know the low-cost ESP8266 and created clients for a PC, tablet and smartphone. Using the skills you've picked up here you can link up other projects with similar functionality to a WLAN. That's not all; the ESP has an 80/160 MHz 32-bit processor on board together with a lot of peripherals. Why not use all this spare capacity to power an entire application? We will be looking into how you can do this in an upcoming article.

(150093)

Web Links

[1] Manufacturer: http://bbs.espressif.com

[2] C-development environment: http://coocox.org

[3] Elektor project: www.elektor-labs.com/project/wifi-for-mcus-using-esp8266.14347.html

[4] Lazarus: www.lazarus-ide.org

[5] NETIO: http://netio.davideickhoff.de/de/

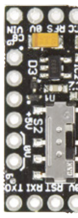
[6] ESP8266 forum: www.esp8266.com

[7] Software download: www.elektormagazine.com/150093

eRIC Nitro

By **Alex Robertson** (Scotland)

In 2015 I entered an "Internet of Things" design contest sponsored by LPRS, a UK manufacturer and supplier of low-power radio solutions, and RS Components' DesignShare. The prize: the winning idea taken all the way from concept through to design and manufacture, supported by LPRS and made a reality by Elektor Magazine. There were many submissions so I was delighted to learn I had the winning entry: eRIC Nitro – a wireless-enabled Arduino board.



My entry to the LPRS Design Contest [1] is based on experience gathered from my home electronics projects. I enjoy the practical aspects of taking my ideas and making them real with a physical board. However, instead of PCBs, these are built using proto boards, point to point wiring and a soldering iron. It can be a lot of fun, but sometimes building a board this way is a time consuming step I could do without. I'm sure other project makers have the same feeling and prefer spending their time on testing out their ideas. Radio communication fascinates me and I have patiently hand built many wireless sensor projects.

There is something magical in being able to transmit and receive data through the air ("aether") without wires and the LPRS eRIC module [2] makes it very easy. For me, having a pre-assembled wireless Arduino board ticks all my design boxes. It reduces the time to test out an idea and for IoT projects it's a perfect fit. The key features of the eRIC Nitro board are given in the inset.

eRIC Nitro Board Key Features

- LPRS eRIC module with a subset of eRIC signals brought out to user accessible pads on the PCB
- PCB form factor similar to Arduino Pro Mini
- Small size: 2.5 x 5.5 cm
- ATmega328 processor with Arduino bootloader with direct serial connection to eRIC module
- 8-MHz operation
- On-board 3.3-V regulator providing power to eRIC module and ATmega328 device
- Header for FTDI friend connector, to allow ATmega328 programming from Arduino IDE and also to provide serial communication interface
- Quickly deployable as a battery-powered low-power programmable RF node
- Total current consumption in low power mode <100 µA
- Open source, open hardware design

Powerful low-power radio

Meet eRIC, the easy radio transceiver module

The eRIC radio transceiver module is the jewel in the eRIC Nitro design. eRIC stands for easy Radio Intelligent Controller and is a 24-pin module (**Figure 1**) containing a Texas Instruments CC430F5137 IC, a mixed-signal System-on-Chip (SoC) that combines an MSP430 microcontroller core with RF circuitry.
The eRIC modules are licence-exempt ISM band radios operating in the 433-MHz, 868-MHz or 915-MHz ISM frequency bands. Two versions of the module are available: eRIC4, operating in the 433-MHz band, and eRIC9 for both the 868-MHz (UK and Europe) and 915-MHz (USA) bands. The modules used in this project come configured as standard 'serial bridge' modules. Send serial data into the eRIC SDI (Serial Data In) pin and the module transmits it out on its RF interface. Data received on the module's RF interface gets transmitted on its SDO pin (Serial Data Out). By connecting a simple wire antenna, a power supply decoupling capacitor and power to a standalone module you can give any project the ability to communicate wirelessly. Any microcontroller with a hardware or software UART can easily communicate with eRIC.
The capabilities of the module don't end at being a serial bridge. What makes eRIC a rock star of the transceiver world is that the existing firmware in the device can be replaced with user MSP430 code (for free!). If you want to get into the internals of the transceiver and replace the firmware, you will need Texas Instruments Code Composer Studio (CCS) to compile your MSP430 application. On the LPRS website a demo CCS project is provided to get you started. The demo code is written in C.

The benefits of going down this road is that it can reduce the component count in your project since the GPIO pins on the transceiver are fully programmable, even supporting A-to-D functionality. LPRS also supply a PC utility to allow you to flash your code into the module. We won't delve further into programming the eRIC but suffice to say, the eRIC Nitro board has all of the connections needed to put the eRIC into bootloader mode to allow the MSP430 firmware to be replaced.

Circuit description

Arduino boards are easily programmable and have the major benefits of a free development environment (IDE), a huge development community and a vast library of user contributed open source software. If you want a software library for a sensor or a peripheral chip, it probably already exists for the Arduino. So basing the eRIC Nitro design on Arduino hardware was an easy choice.
The Arduino section of the eRIC Nitro is derived from the open-source Arduino Pro Mini. There are a few changes though. Looking at the schematic in **Figure 2**, you'll see the addition of inductive filter L1 at the ATmega328 (IC1) power supply, recommended by its manufacturer to improve the performance of the analog-to-digital converter. Another change is the clock source. The original 16-MHz resonator has been replaced by an 8-MHz version (X1). This keeps the MCU's clock speed within the datasheet specification for a 3.3-V supply.

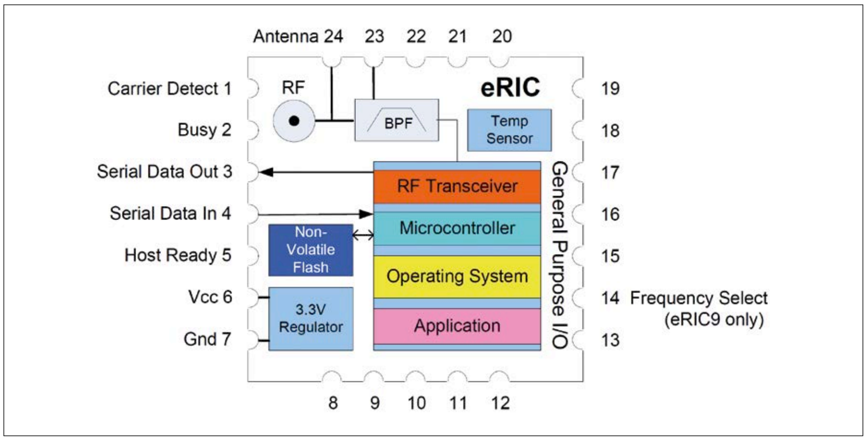


Figure 1. X-ray of eRIC revealing its guts. (Source: LPRS)

It was a design decision to keep the pinout of the board as close as possible to the Pro Mini original. With so many peripheral ICs available with an I²C connection, it makes sense to bring Arduino's I²C pins A4 and A5 out on connector K2, replacing the A2 and A3 of the original Pro Mini design. The eRIC module (MOD1) reset pin (RST) is also made user acces-

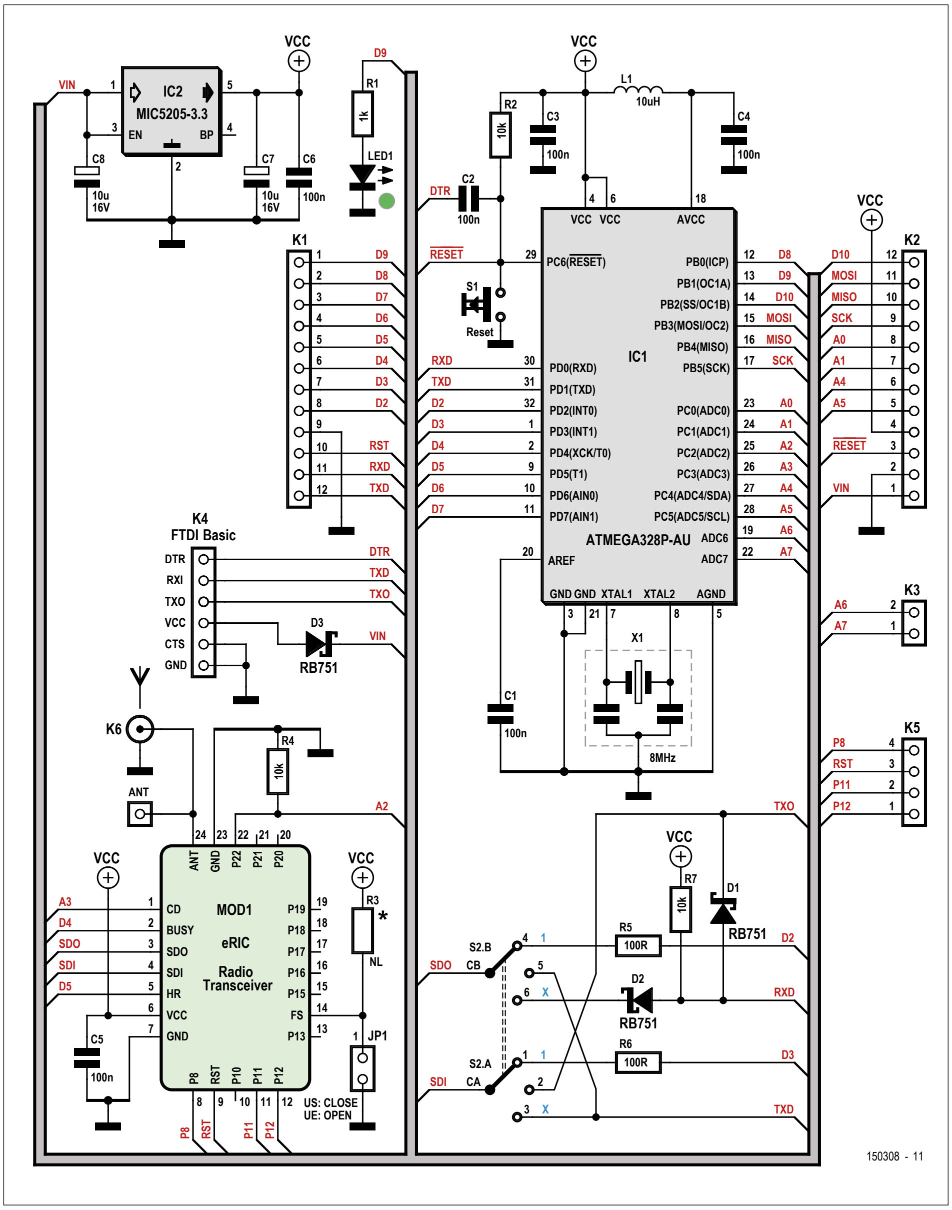


Figure 2. Schematic of the eRIC Nitro board. eRIC sits in the lower-left corner, the rest is Nitro.

sible and is brought out to pins on connectors K1 and K5 to replace one of the original Pro Mini RESET pins. Having the MCU and eRIC reset pins separate gives the user more control of the eRIC module. A user program can reset the eRIC module without having to reset the MCU. Voltage regulator IC2 provides a supply voltage V_{CC} of 3.3 V. IC2 is rated at 150 mA at 3.3 V output. However, the output current capacity is directly related to the input voltage V_{IN} on connectors K2 and K4. If you need this level of current then please refer to the MIC5205 datasheet for more information on calculating the maximum output current for any given V_{IN} since a high V_{IN} voltage and a high load current on the IC2 OUT pin could end up with IC2 overheating possibly causing permanent damage to IC2 and to the eRIC Nitro board.

The eRIC Nitro does not have the 'power present' LED of the Pro Mini. This was removed to reduce the board current consumption — an important requirement for low-power environments. It's always convenient to have an LED under program control and on the eRIC Nitro it's connected to pin D9 on connector K1.

Switch S2 and the passive components D1, D2, R5, R6 and R7 all play important roles in connecting the eRIC SDI and SDO pins to the MCU. These components permit SDI and SDO to be connected to the MCU hardware UART RXD and TXD pins, or software UART using pins D2 and D3 or to bypass the MCU completely while remaining accessible via the TXO and RXI pins of connector K4.

R3 and JP1 are only needed when an eRIC9 module is used. JP1 is needed to select the module's frequency (open for 868 MHz, closed for 951 MHz). R3 is not really needed as the module includes a pull-up resistor; it is on the board just in case.

Sending data using eRIC

Under the bonnet of the eRIC module is where data received on SDI gets encoded before transmitting the SDI data over the RF interface. The eRIC firmware adds all necessary preamble, message length and message CRC information. Similarly, the firmware strips the same preamble, message length and CRC encoding data from RF data received by the module before transmitting it out serially on SDO.

The default serial SDI/SDO pin data rate is 19,200 baud, with an over-the-air (OTA) data rate of 38,400 baud. The SDI/SDO pin data format for one character is one start bit, eight data bits, no parity and one stop bit (19200n81). Data received on the SDI pin is automatically put into an internal data buffer (maximum 250 bytes). The module transmits the buffer contents over the RF inter-

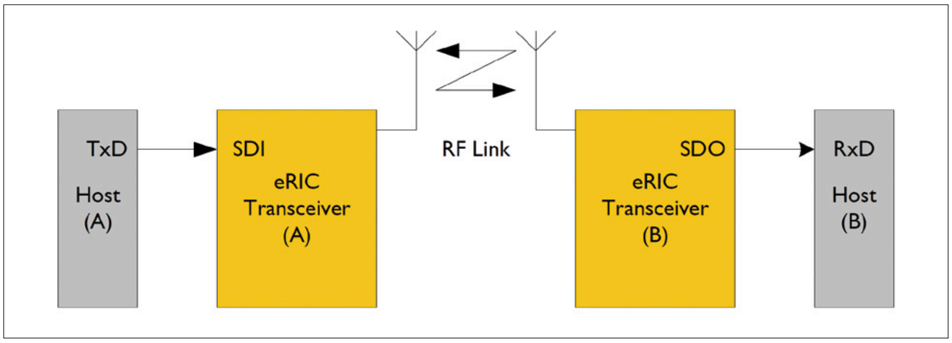


Figure 3. Serial bridge system overview. (Source: LPRS)

face (emptying the buffer in the process) when it detects no data has was received on the SDI pin for a period equal to two characters (**Figures 3 and 4**).
Sending and receiving data is easily done with no need for a hardware handshake between eRIC and the MCU. Receive and transmit using the SDI and SDO pins is easily achievable just using software delays (for the two-character transmit delay on the SDI pin). If a hardware handshake is needed then the eRIC can support this using Clear to Send (CTS) and Request to Send (RTS) signals on the respective BUSY and HOST READY (HR) pins. The BUSY output pin goes low when the module is ready to receive data on the SDI pin. It will not send any RF data it has received if the HOST READY input pin is driven high by the MCU. By default the eRIC firmware disables the handshaking functions BUSY and HOST READY. If a user wants to enable the handshaking signals they can send the string data `ER_CMD#A51` to the module. It will trap this data string, recognizing it as a valid command and enable the hardware handshake signals. BUSY and HOST READY are connected to Arduino's D4 and D5 pins. If hardware handshaking is not required then D4 and D5 are free for Arduino program use.

eRIC modules have many neat features, one being the ability to detect if a radio frequency band is currently occupied by another source transmitting data. The module has a Carrier Detect (CD) pin which swings High to flag an RF carrier detected within the radio receiver bandwidth. In the eRIC Nitro the CD pin is connected to Arduino's A3 pin. eRIC command `ER_CMD#T8` is also useful for a similar purpose as it returns the Received Signal Strength Indication (RSSI) of the last packet received.
The eRIC module firmware permits changing many of the radio transceiver parameters such as transmit power, receive sensitivity, low power modes, SDI/SDO baud rate, and radio transmit and receive frequency. The module's datasheet describes these in more detail.

Antenna connection and radio range

As might be expected, the eRIC Nitro needs an antenna to transmit data using the RF interface. There are three antenna possibilities:

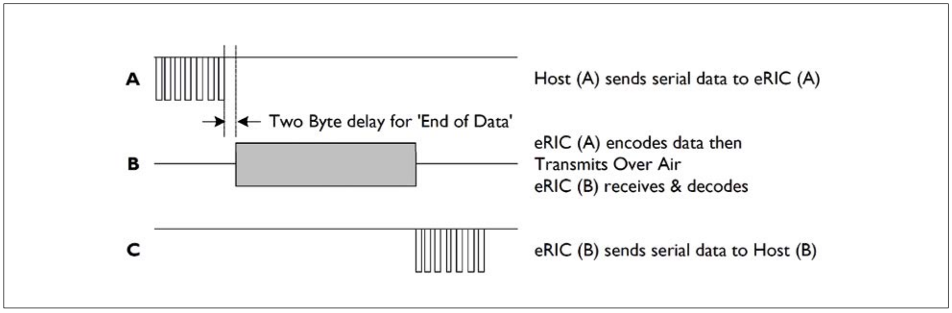

A
B
C
Two Byte delay for 'End of Data'
Host (A) sends serial data to eRIC (A)
eRIC (A) encodes data then
Transmits Over Air
eRIC (B) receives & decodes
eRIC (B) sends serial data to Host (B)

Figure 4. Serial bridge timing diagram. (Source: LPRS).

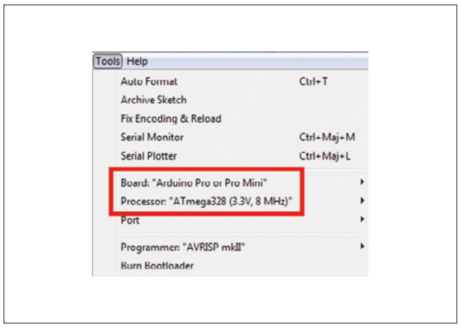

Tools
Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Maj+M
Serial Plotter Ctrl+Maj+L
Board: "Arduino Pro or Pro Mini"
Processor: "ATmega328 (3.3V, 8 MHz)"
Port
Programmer: "AVRISP mkII"
Burn Bootloader

Figure 5. Choosing the right board in the Arduino IDE. It may look slightly different depending on the version of the IDE.

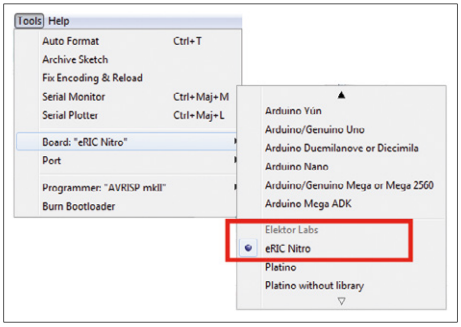


Figure 6 – It is even easier if you have the Elektor.Labs Arduino board package [4] installed.

- an SMA antenna on edge connector K6;
- an external antenna connected to the eRIC module UFL antenna connector;
- a minimum cost ¼-wavelength wire antenna.

When using a wire antenna, solder the relevant length of (stiff) wire to the ANT pad on the PCB (approximately 171 mm for the eRIC4 and 82 mm for the eRIC9 operating at 868 MHz). It goes without saying that when using a wire antenna, there is no need to have the SMA connector on the board and the SMA position on the PCB can be left unpopulated. If using the eRIC9, leave jumper JP1 open when operating at 868 MHz. JP1 needs to be closed to TX/RX at 915 MHz.
The transmit/receive range depends on many factors. These include the radio power (max 10 mW for eRIC4) and sensitivity, obstacles in the path of the radio signal, antenna type and also radio frequency. SMA and UFL antennas should give better performance than a wire antenna, but a wire antenna will still give good results. The physics of path losses say that the range at 433 MHz exceeds that at either 868 MHz or 915 MHz.

Keep in mind the legal requirements of using an ISM radio, both in terms of transmit power and transmit duty cycle [3].

Programming

Programming the eRIC Nitro couldn't be easier as the good folks at Elektor have pre-installed an Arduino bootloader into the board's ATmega328 chip. This is a derivative of the standard Arduino Uno Optiboot bootloader, conveniently flashing LED1 during programming. The board can be programmed using the Arduino IDE (version 1.0.6 or greater), and selecting board "Arduino Pro or Pro Mini (3.3 V, 8 MHz)" or "Arduino Pro or Pro Mini" plus a processor "ATmega328 (3.3 V, 8 MHz)" (**Figure 5**, this depends on the version of the IDE). An Arduino board file [4] is also available for extra ease of use. Add this into the IDE (v1.6.6 and up) and the eRIC Nitro appears in the list of supported boards (**Figure 6**).
Just as with Arduino Pro Mini boards, header K4 has the same pinout as the 6-pin end connector of FTDI basic breakout boards and FTDI TTL-232R cables. The FTDI chip converts USB signals to serial signals and allows programming the ATmega328 from the Arduino IDE. The

USB-to-serial function is also useful when you need to communicate to the eRIC Nitro from a PC. Always use a version of the FTDI chip having 3.3-V TTL signal levels and not the 5-V TTL version. The Nitro board will be damaged if you don’t use the 3.3-V version. And always make sure you are connecting the FTDI pins to the matching pins of K4. The green and black wires match up with the “G” and “B” text on the Nitro PCB. Get this wrong and the Nitro board will be damaged.

When using the Arduino IDE to program the ATmega328, put switch S2 in the “upper” position (switch button closest to connector K1). While in this position eRIC’s SDI-SDO pins are connected Arduino’s D3-D2 pins. When switch S2 is in the ‘lower’ position (switch button closest to connector K2), eRIC’s SDI-SDO pins are linked to Arduino’s UART TXD-RXD pins and will interfere with the programming process.

When using the easyRadio Companion PC utility (available from the LPRS website) to talk directly to the eRIC via K4 (bypassing the ATmega328), put S2 in the ‘middle’ position. Before doing so, you maut force the ATmega328 into reset by connecting K2 pin /RESET to 0 V (with a jumper for instance).

Example software

Elektor.Labs have kindly released example Arduino hardware and software files [5] to get you started with the eRIC Nitro board. Software example `eRIC_blink.ino` is a good starting point to get you up and running with the Nitro board. With two boards you can begin two-way communication and example `eRIC_bridge.ino` is a great template for your future eRIC projects. Expect to see further software examples appear in the future.

You 2 can build your own

This project is open hardware and software and all the files needed to build and program your own eRIC Nitro board can be downloaded from GitHub [5]. The PCB design files are available as DesignSpark PCB and Eagle files. Note that assembling the board can be challenging as some components are rather hard to solder manually. The resonator is one of those; the Schottky diodes are so small that it’s hard to see their pads. For this reason pre-assembled boards can be bought from the Elektor Store [6]. The eRIC module is not mounted and must be ordered separately because you must choose the right one. Anyone with lesser soldering skills should be able to mount the radio modules manually.

Component List

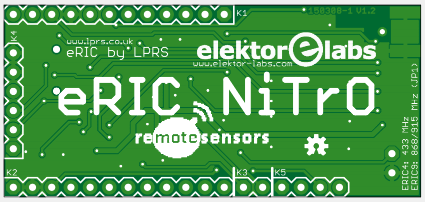
Resistors
All 5%, 50V, 0.1W, 0603
R5,R6 = 100Ω
R1 = 1kΩ
R2,R4,R7 = 10kΩ
R3 = 10kΩ

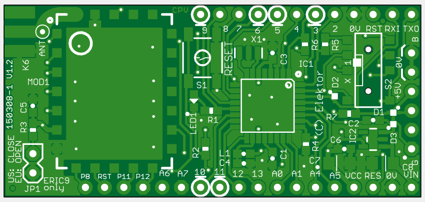
Capacitors
C1–C6 = 100nF, 0603
C7,C8 = 10µF 16V, tantalum

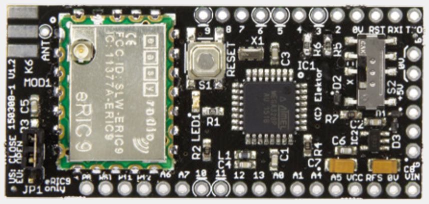
Inductor
L1 = 10µH, 0603

Semiconductors
IC1 = ATmega328P-AU
IC2 = MIC5205-3.3
D1,D2,D3 = RB751
LED1 = LED, green, 0603

Misc.
X1 = 8MHz resonator, SMT
K1,K2 = pinheader, 1x12, 0.1’’ pitch
K3 = pinheader, 1x2, 0.1’’ pitch
JP1 = pinheader, 1x2, 0.1’’ pitch w. jumper
K5 = pinheader, 1x4, 0.1’’ pitch
K4 = pinheader 1x6, 0.1’’ pitch
K6 = SMA edge connector
S1 = tactile switch
S2 = DP3T slide switch
MOD1 = eRIC4/eRIC9 SoC Radio Transceiver
PCB 150308-1 v1.2







Design, Make, Use

The eRIC Nitro board is a fantastic platform for Arduino projects requiring wireless connectivity. The Internet is full of sensor, actuator and Wi-Fi breakout boards just waiting to be connected to this board. And if you want to progress further to programming in the native environment of the eRIC module, this is supported too. What project will you create? ◄

(150308-I)

Web Links

[1] www.lprs.co.uk/smart-eric-iot-competition/

[2] www.lprs.co.uk/easy-radio/eric/

[3] www.ti.com/lit/an/swra048/swra048.pdf

[4] https://github.com/ElektorLabs/Arduino

[5] https://github.com/ElektorLabs/150308-eRIC-Nitro

[6] www.elektor.com/150308

[7] www.elektor-labs.com/eric-nitro

Fridge Thermostat Calibration

Using the Elektor Android I/O board

By **Elbert Jan van Veldhuizen** (Netherlands)

If you have an Android I/O board, you don't need any fancy measuring equipment when you have to calibrate a new fridge thermostat. The only things you need to add are a resistor and an NTC. An app on an Android mobile phone accurately tracks the temperature during the calibration process.

Fridges have the unwelcome tendency to go wrong at times. Fortunately, the fault is usually not caused by the expensive compressor. More often than not, it's the cheaper thermostat that's broken. Replacing a thermostat will be a piece of cake for the typical Elektor reader. However, it will be more difficult to find exactly the same type of thermostat. The author found that the specific thermostat required for the fridge would cost five times as much as a universal thermostat. Furthermore, the supplier couldn't guarantee that it was an exact replacement for that type of fridge. The author decided to buy a cheap universal thermostat and calibrated it using the Android I/O board and a homemade app. It wasn't really worth the effort to build a separate device for a job that only needs doing once every five to ten years. But if you have a spare Android I/O board lying about, it won't take long to make a fridge thermostat calibrator!

Thermostat operation

Before we look in detail at the fridge thermostat calibrator, we first have a look at how a fridge thermostat works and what the differences are between the various types.

A fridge thermostat has a capillary tube in which a liquid evaporates (often the same type of liquid that is used in the fridge for cooling). When it warms up, more gas is created. This increases the pressure, which forces a diaphragm to move, closing a switch that turns on the compressor. When it becomes cooler, the gas condenses, which causes the diaphragm to retract, turning off the switch.

The electrical circuit of the fridge is shown in **Figure 1**. This circuit also shows a heating element used for automatic defrosting, although this may not be present in all fridges. When the compressor is switched on, the heating element is shorted and will therefore be off. When the compressor is turned off, a small current flows through the resistor in the thermostat and through the heating element, all via the motor of the compressor. However, this current is too small to make the motor turn. The heating element warms up the cooling element of the fridge to above freezing, which causes any ice on it to melt and to drain away through a small hole in the fridge.

Different types of thermostat will have different values of resistors. However, you can easily reuse the resistor from the old thermostat.

The thermostat measures the temperature of the cooling element (rather than

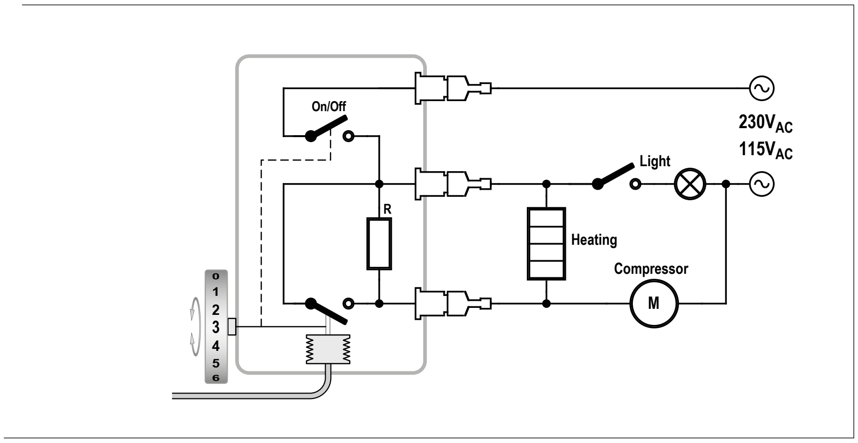


Figure 1. Circuit diagram for the thermostat and the wiring in the fridge.

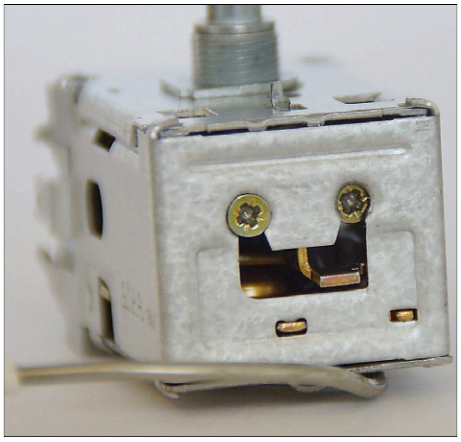


Figure 2. The external (left) and internal (right) adjustment screws on the thermostat.

the temperature in the fridge!). In a combination fridge the cooling element cools down to a temperature between −15 and −25 degrees Celsius (turn-off point). The thermostat then switches off and lets the temperature rise all the way to above the freezing point, which starts the automatic defrosting (turn-on point). The compressor will then be turned on again. The temperature control on the thermostat determines the turn-off point. The screen dump in **Figure 4** shows the graph for a complete cycle.

The measuring circuit is extremely simple

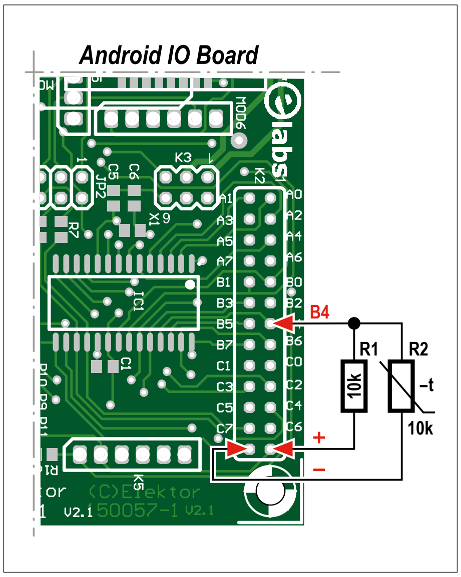


Figure 3. The 'measuring circuit' consists of a resistor and an NTC.

Table 1. Effect of the adjustment screws on the switching temperature.

	turn-off temperature	turn-on temperature
1 x CW external	+3 °C	+2 °C
1 x CW internal	−2 °C	+2 °C

There are two screws on the back of the thermostat (**Figure 2**). One screw is fixed to the casing (from here on called the external screw) and one screw is inside the thermostat (the internal screw). The adjustment of both these screws determines the turn-on and turn-off temperatures.

To start with, the turn-off temperature has to be low enough to make it cold enough inside the fridge. But if the turn-off temperature is set too low, the fridge will never be able to reach this temperature and the compressor will remain turned on, causing the freezer compartment to freeze solid. The turn-on temperature should be about 4.5 degrees Celsius for most fridges.
In **Table 1** you can see what effect the screws have in the thermostat of the author, to get an idea what to expect. CW means clockwise, 1 turn to the right and '+' indicates that the switch point is at a higher temperature.

Calibration

To calibrate the thermostat you have to set the temperature control to its maximum setting (usually on '6'). Turn the fridge on and adjust the thermostat so that the compressor turns off after a long time (between 1 to 2 hours, for example). When it is warming up you should adjust the turn-on temperature. The fridge will then cool again, when you should adjust the turn-off temperature. You should repeat this process several times until you're happy with the result. You can now set the temperature control to a central position and check that the fridge settles down to a normal temperature (about 3 to 4 degrees Celsius).

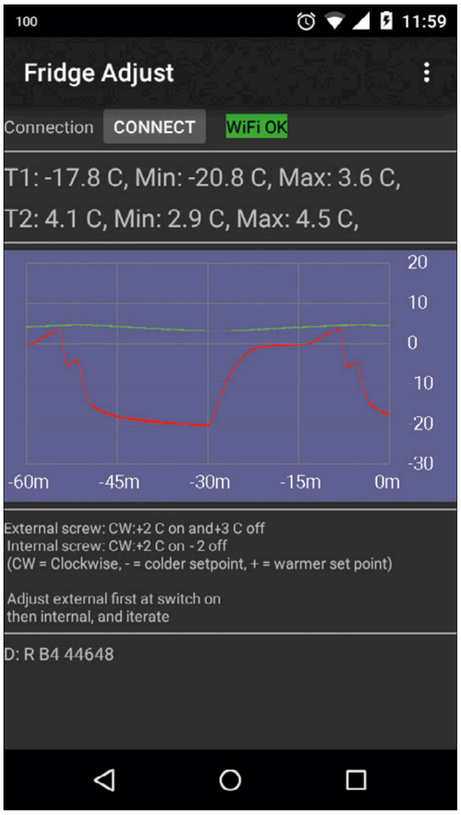
It is important to keep a log of all events during this process. Which temperature did you measure? Which screws did you turn, and how much by and in which direction? It will be useful if you make a mark on the screwdriver, so you can easily see how much you turn it by. You can determine from the log if you're going in the right direction, and it will be easy to reverse any changes made, if necessary. It is important to note that it will take several cycles before the fridge settles down to a stable temperature, so you will need some patience.

The circuit

The circuit diagram is shown in **Figure 3** and is extremely simple. It consists of a potential divider with an external NTC, which is connected to pin B4. The Android I/O board already has an NTC on board, which is connected to pin B3. When the Android I/O board is put in the fridge (Bluetooth and WiFi can go through a metal fridge), then B3 will measure the fridge temperature. The external NTC has to be mounted at the end of the capillary tube. In some fridges this is easy because the cooling element in the fridge sticks out a little bit. In other fridges the capillary tube has to be pushed through a whole to get to the cooling element. In that case you should use an SMD for the NTC, which is connected via two thin wires. This NTC can then be mounted to the end of the capillary tube using some duct tape. As an alternative, the NTC can be mounted onto the cooling element in the fridge (also using duct tape) and covered with some isolation foam. In that case you won't measure the exact cooling temperature, but it will be close enough for the calibration.

The app

A screen dump of the app is shown in **Figure 4**. 'T1' is the NTC connected to pin B4 and T2 is the NTC mounted on the Android I/O board (pin B3). The app displays the measured temperature, including the minimum and maximum measured temperatures. It also gives a graphical representation of the temperature over the last hour. From the menu you can reset the minimum and maximum temperatures, and clear the graph. The software in the app is based on part 2 of the Android I/O board Etch Control example (Elektor November & December 2015). In onCreate() the GUI is defined; a loop in a thread takes care of the regular transmission of commands to the Android I/O board, and a message handler receives the measured data and passes it on to the GUI.



T1: -17.8 C, Min: -20.8 C, Max: 3.6 C,
T2: 4.1 C, Min: 2.9 C, Max: 4.5 C,

External screw: CW:+2 C on and+3 C off
Internal screw: CW:+2 C on - 2 off
(CW = Clockwise, - = colder setpoint, + = warmer set point)

Adjust external first at switch on
then internal, and iterate

D: R B4 44648

Figure 4. Screendump of the calibration App.

What is different here is the graph. An imageView is created in the GUI for the space taken up by the graph. When the graph is created, a bitmap is made in memory that has the same dimensions as the imageView. A Canvas is defined in de bitmap, where the graph is made using the drawLine, drawPoint and drawText commands. The color is defined by the Paint object. Once everything has been drawn via the Canvas in the bitmap, this bitmap is copied to the GUI and it becomes visible to the user. In **Listing 1** you can see (an abstracted version of) the code for this.

Listing 1. An example for creating the graph.

```
public void makegraph() {
      Paint paint;
      Bitmap bg;
      Canvas canvas;
      int color;

      double xm=imGraph.getWidth();
      double ym=imGraph.getHeight();
      bg = Bitmap.createBitmap(xm, ym, Bitmap.Config.ARGB_8888);
      canvas = new Canvas(bg);
      paint = new Paint();

      // draw line
      paint.setColor(Color.parseColor("#808080"));
      // ...
      // x1,y1 x2,y2 begin and end point
      canvas.drawLine(x1,y1,x2,y2, paint);

      // draw text
      int textsize=(int)(xm/24);
      paint.setColor(Color.WHITE);
      paint.setTextSize(textsize);

      // draw point
      paint.setColor(Color.parseColor("#FF0000")); //red
      canvas.drawPoint(x1,y1, paint);

      // Make graph visible
      // in onCreate(), imGraph is defined as:
      //    ImageView imGraph;
      //    imGraph = (ImageView) findViewById(R.id.idGrafiek);
      imGraph.setImageBitmap(bg);
}
```

As usual, you can download the apk file and the source code for this app from the magazine website [1].

Good luck with calibrating your fridge thermostat!

(150516)

Web Link

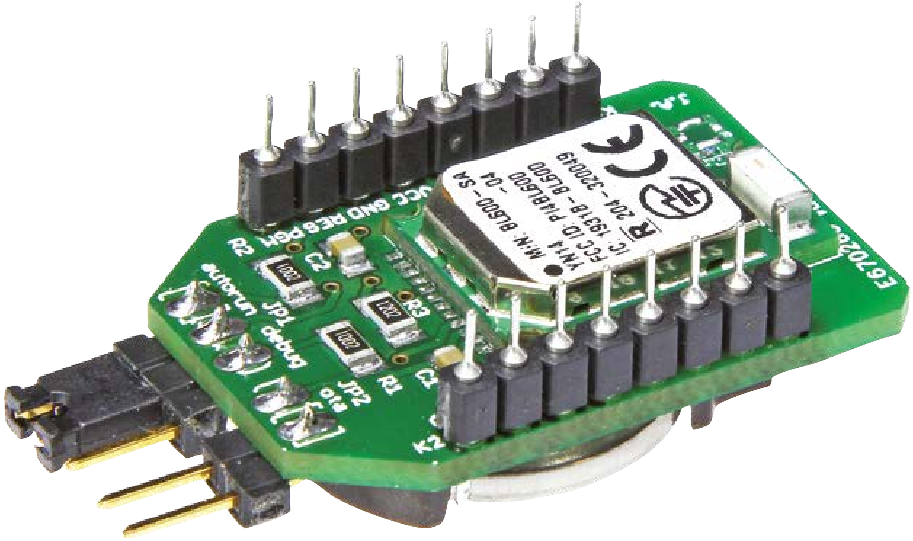
[1] www.elektormagazine.com/150516

BL600 e-BoB

Bluetooth Low Energy Module

(part 6)

- Low Energy, 5 µA
- The TIMER event
- Data exchange events
- The connection event



By **Jennifer Aubinais** (France) elektor@aubinais.net

In the previous installments, we have examined together the basics of programming our BL600 e-BoB BL600 and shown how to implement it. In this the sixth and final installment in the series we take a look at its low consumption (Low Energy) which is precisely its strong point! Then you'll be flying solo.

Low Energy, 5 µA

In deep-sleep mode, the consumption is only 0.4 µA, but in this mode it can only be woken up by resetting the module or by a change of state on an input. Instead, we're going to be using the stand-by mode, with a consumption of 5 µA, and module will be woken up by our program, e.g. by using TIMERs. To illustrate these explanations, we're going to be using just a single LED. If you connect other devices to your BL600, you'll need to adapt your program so as to maintain the 5 µA in stand-by mode. To highlight the module's low consumption, we're going to make it operate alternately in normal and stand-by mode. Connect your (micro-) ammeter as per the diagram: when the LED is out, the current is 5 µA.

The source code for the *bl600-6.sb* test program shown below is available from the Elektor Magazine website [1]. If you want to alter it, refer to the previous installments which give all the details on this subject.
The stages:

- **Stopping the Bluetooth:** the Bluetooth is stopped using the *BleAdvertStop* function and woken up again by the *BleAdvertStart* function. The alternation will be achieved using TIMERs (*TIMER* paragraph for the description of the code in red and green).

- **LED off:** Everything connected to the module consumes power. So our LED will be taken LOW by the *GPIOWrite* function. We'll turn it on when the Bluetooth is running and turn it off when it is stopped (using functions in the TIMER – see below).

- **UART stopped in NON DEBUG mode**: The module's serial communication facility also consumes power. We'll turn it off using the *UartClose* function, the TX and RTS ports will be set to outputs with *High* and *Low* states respectively. Two modifications are needed here to avoid a ~~~*default*~~~ error:

  - If we close the serial port when the program is started, , the serial port buffer in the MAIN program is still full. Hence for it to be emptied, we shall wait 1000 ms (an arbitrary duration) before closing it, using a TIMER in the mauve code.
  - Once the serial port is closed, the HandlerLoop handler transfers the data received by the Bluetooth to the serial port and vice-versa using the *BleVspUartBridge* function. We'll put this function as a comment and will compose our own code in orange (see *Data exchange events*).

- **Measuring:** The high internal resistance of some current meters can hinder the measurement in awake mode. For example, with a resistance of 100 Ω, for a current of 10 mA (LED lit) the voltage drop across the meter will be 1 V: so the module won't work anymore… Select the 2000 µA range rather than 200 µA.

The TIMER event

The TIMER, covered in the May 2015 article on the light-chaser [4], is used to create alternating stops and restarts of the Bluetooth. To do this, we need to code:

- the declaration of its handle: *OnEvent EVTMR1 Call FuncTimer1*
- the function: *Function FuncTimer1()*
- the trigger for the TIMER: *TIMERSTART(1,4000,0)*

In order for the module's low consumption to show up on an meter, we enable the Bluetooth (LED on) for 25 ms (*bleadvertstop* function, code in red), and stop it (LED off) for 4000 ms (*bleadvertstart* function, code in green).

```
n = Bleadvertstart(0,Adrt$,25,4000,0)
```

The TIMER will be used again to delay the closing of the UART port.

Data exchange events

The data received by the Bluetooth and on the serial port are processed by a handler, just as we did, for example, to intercept the data coming from your phone to control a 3-colour LED [4]. Here, this poses us a problem, as we have closed the UART port; this is why we have deleted the *BleVspUartBridge* code (put into a comment). To illustrate this example, the text received in Bluetooth will be sent back in Bluetooth, but with an offset of one character (I'll come back to this later).

The connection event

In the previous articles, we've already tackled the *EVBLEMSG* event in order to be able to handle connection and disconnection events. In the thermometer program in the January/February 2015 article [1], when connecting, the first step is to recover the value from the NTC, the second is to send this to the smartphone, and the last, to disconnect the module from the phone. This time, we're going to light an LED on output 2 of our module when it is connected; this will be turned off when disconnected.
Don't forget the initialization of port 2 when exiting the main program. We're going to copy the Bluetooth message handler *EVBLEMSG* from the *cli.manager.sblib* library and create our own handler, renaming it *MyHandlerBleMsg* (code in blue).

Creating our program

To end this list of the elements to be recycled in order to create our *Low Energy* program here, we must lastly mention the *upass.vsp.sb* program used in other articles as the basis for the UART services [1][4][6]. This will be reworked and will be used to modify the data received by our module. To do this, we rename it *BL600-6.sb* and change two parameters in this first version:

- ENABLE_DEBUG_PRINTS: 0 (*debug* mode disabled)
- DEVICENAME: "JA_TEST"
- You can test the operation of your program by transferring it to your BL600 e-Bob. It will run automatically, otherwise enter the command *AT+RUN* [3]. You can now test your program's communication using the *SERIAL* program from *TOOLKIT* on your smartphone [2].

The two TIMERs:

- Code in green:
  - declaring its handle: *OnEvent EVTMR1 Call FuncTimer1*
  - the function: *Function FuncTimer1()*
  - turning the LED on: *GpioWrite(2,1)*
  - starting the Bluetooth: *BleAdvertStart*
  - the trigger for the next TIMER: TIMERSTART(2,25,0)

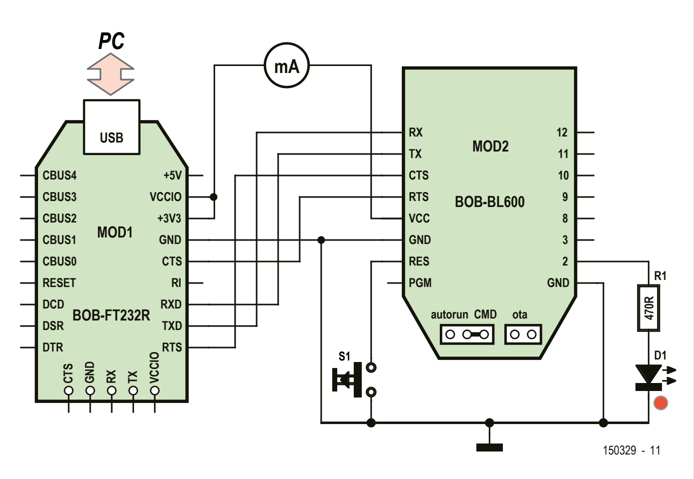


Figure 1. As usual, we'll be using the Elektor USB-serial FT232 bridge [8] for communicating with the BL600.

Component List

Resistors :
R1 = 470Ω

Semiconductors
D1 = LED, 3mm (select color)

Miscellaneous
KS1 = pushbutton
MOD1 = FT232 e-BoB, ready assembled, Elektor Store # 110553-91 [7]
MOD2 = BL600 e-BoB, ready assembled, Elektor Store # 140270-91 [7]
or printed circuit board # 140270-1 [7]

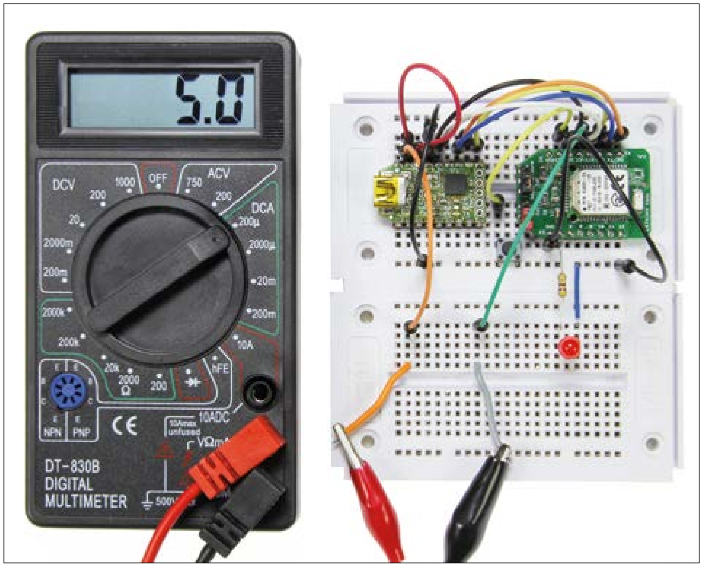


Figure 2. Photo of the experimental assembly on a prototyping board.

- Code in red:
  - declaring its handle: *OnEvent EVTMR2 Call FuncTimer2*
  - the function: *Function FuncTimer2()*
  - turning the LED off: *GpioWrite(2,0)*
  - stopping the Bluetooth: *BleAdvertStop*
  - the trigger for the next TIMER: *TIMERSTART(1,4000,0)*

The *BleVspUartBridge* function must be deleted if we want to close the UART and hence consume less power. To show the module is functioning, we send back the received text with an offset of one character: a becomes b, d becomes e, etc. (code in orange).

- We've taken the code for the *HandlerLoop* function from the *cli.upass.vsp.sblib* library. To avoid duplicates that would cause a compilation error, let's rename our function ***My****HandlerLoop*.

- The four handlers that call the *HanderLoop* function:
  - data arrive at the module's UART port
  - data arrive at the module's Bluetooth port
  - the module's Bluetooth port buffer is empty
  - the module's UART port buffer is empty

- The *BleVspUartBridge* function creates a loop between the phone and the UART port. We put this line of code into a comment.

- The *BleVSpRead* function lets us read the data received by the Bluetooth. Attention, the manufacturer has imposed a limit of 20 characters.

- The *STRPOS* function makes it possible to know if the text received ends with a carriage return.

- To send back the received text with an offset of one character, we're going to use the two functions *StrGetChr* (convert a character at a given position in a string to a byte) and *StrSetChr* (change the value of a character by the value of a byte at a defined position).

- To end, the text will be transmitted in Bluetooth via the *BleVspWrite* function.

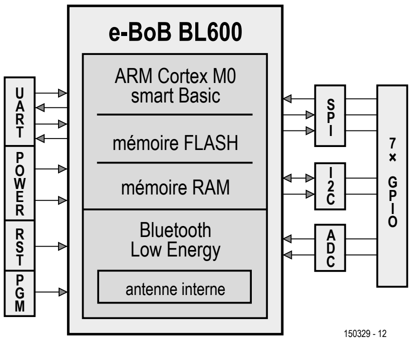
e-BoB BL600

Reminder of the BL600 module specifications:

- UART, I²C, SPI interfaces
- 28 general-purpose inputs/outputs (GPIO)
- 6 analog inputs (10-bit ADC)
- consumption:
- 5 µA in stand-by / 0.4 µA in deep-sleep mode
- 10 mA during transmission
- easy programming in smartBASIC
- nRF51822 chip from Nordic Semiconductor.
- compact: 19 × 12.5 × 3 mm
- free-field range: up to 20 m

Bluetooth in BASIC!

The BL6000-SA's smartBASIC (event-oriented) programming language simplifies the incorporation of Bluetooth into your applications by making it easier not only to manage sensors connected directly to the module, but also to transmit the measured values to any Bluetooth v4.0 receiver (smartphone or tablet, computer, bridge, etc.)



Specifications of the Elektor e-BoB breakout board using the BL600-SA:

port K1	serial port used for loading the program into the BL600 power pins (3.3 V) reset line (RESET) PGM pin (for possible updates to the module's firmware)
port K2 7 logic input/ outputs	2 × 10-bit analog inputs (pins 2 and 3) I²C port (pins 8 and 9) SPI port (pins 10, 11, and 12)
jumper JP1	cmd: AT commands autorun
jumper JP2	OTA (Over The Air)

Android application and program

On the BL600 manufacturer's website [7], you will find the source code for the *Toolkit* application for Android (and also iOS) which includes the following services:

- BPM (blood pressure)
- HRM (heart rate)
- *Proximity*
- MTH (medical thermometer)
- *Serial* (UART)
- OTA (Over The Air)
- *Batch*

To thrive, connected object networks need low power consumption wireless communication. This breakout board for the Bluetooth Low Energy module is the dream accessory for exploring the IoT.

The program will no longer send data to the UART port, so we can close the UART port, after an arbitrary delay of 1000 ms (code in **mauve**):

- declaring its handle: *OnEvent EVTMR0 Call FuncClose*
- the function: *Function FuncClose()*
- closing the port: *UartClose()*
- the TX and RTS ports will be set to outputs with *High* and *Low* states respectively: *GPIOSetFunc*
- triggering the TIMER in the MAIN program: *TIMERSTART(0,1000,0)*

We want to intercept the status of the connection so as to turn the LED on when connecting or off when disconnecting (in reality, the LED will flash as a result of TIMER1 and TIMER2).

The *HandlerBleMsg* function code in blue comes from the *cli.upass.vsp.sblib* library:

- We've taken the code for the *HandlerBleMsg* function from the *cli.upass.vsp.sblib* library. To avoid duplicates that would cause a compilation error, let's rename our function ***My****HandleBleMsg*.
- When connecting via BLE_EVBLEMSGID_CONNECT:
  - the TIMERs are stopped: *TIMERCANCEL*
  - the LED is turned on: *GpioWrite(2,1)*

We have offered you for download [6] the extremely simplified source code for an Android phone application using just the UART service. We have used Android Studio, available under Windows, MAC OS and Linux [7]. A reminder that the manufacturer has created a library `laird_library_ver.0.18.1.1.jar` in order to speed up development of Android applications in normal Bluetooth and Bluetooth Low Energy.

The functions described in this series of articles:

inputs / outputs:
- configuring the pin on the BL600 (nFunction = 1 or 2): `rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)`
- reading the signal 0 or 1: `rc = GPIOREAD(nSigNum)`
- writing (nNewValue = 0 or 1): `rc = GPIOWRITE(nSigNum, nNewValue)`

analog/digital converter:
- configuring the pin on the BL600 (nFonction = 3) : `rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)`
- reading the signal (0 to 1023): `rc = GPIOREAD(nSigNum)`

I²C port:
- opening the port: `rc = I2COpen(nClockHz, nCfgFlags, nHandle)`
- writing an 8-bit value: `rc = I2CWriteReg8(nSlaveAddr, nRegAddr, nRegValue)`
- reading an 8-bit value: `rc = I2CReadReg8(nSlaveAddr, nRegAddr, nRegValue)`
- closing the port: `I2Close(handle)`

SPI port:
- opening the port: `rc = SpiOpen(nMode, nClockHz, nCfgFlags, nHandle)`
- reading / writing: `rc = SpiReadWrite(stWrite$, stRead$)`
- closing the port: `SpiClose(handle)`

Tools for compiling and transferring

All the tools and examples are available for download [7]. When you apply to open an account on the manufacturer's website, specify: "Elektor reader". All you have to do is download and then unzip the document `Firmware Files version 1.5.70.0 – Revision 5 i` which contains:

- program examples in the `smartBASIC_Sample_Apps` directory
- the `UwTerminal.exe` program in the `smartBASIC_Sample_Apps` directory
- the library in the `smartBASIC_Sample_Apps/lib` directory
- a number of examples (`UserManualExampleCode`)
- the special Notepad++ configuration for smartBASIC (`smartBASIC(notepad++).xml`)

Normal commands

- AT I 0: BL600 revision number
- AT I 3: BL600 firmware version
- AT+DIR: list of the programs in the BL600
- ATZ: reset BL600
- AT&F 1: clear memory and restart BL600
- AT+RUN "xxxx": run program xxxx

- When disconnecting via BLE_EVBLEMSGID_DISCONNECT:
  - the LED is turned off: *GpioWrite(2,0)*
  - the first TIMER is started again: *TIMERSTART(1,10,0)*

Now all the elements are there for you to create your own `BL600-6.sb` program. After checking the low consumption of our module, it's desirable to modify the value of the TIMER defined as 1 from 4000 ms to e.g. 300 ms for a better connection with your smartphone: *TIMERSTART(1,300,0).* The moment has come to launch yourself into programming *your* BL600, on *your* own e-BoB, for *your* ANDROID application; the information you need is all in last month's article [6].

Open conclusion

With this summary of what we have described since the initial Bluetooth thermometer project, we are closing this 6-installment article with double satisfaction: our efforts to facilitate the implementation of this module have been recognized and appreciated by not only the Elektor readers who are buying the e-BoB, but also the manufacturer of the BL600-SA. Other projects based on the BL600 are waiting in the author's and Elektor's laboratories. What's more, the BL600 has a cousin, the BL620, another Bluetooth module, but with the role of Master, which makes communication even easier. You'll be seeing it soon in Elektor Magazine…

(150329)

```
'//**************************************************
'// Laird Technologies (c) 2013
'// Jennifer AUBINAIS (c) 2015
'// ++++++++++++++++++++++++++++++++++++++++++++++++

'//**************************************************
'// Definitions
'//**************************************************
#define AUTO_STARTUP                        1
'//Set this to 0 to disable all debugging messages
#define ENABLE_DEBUG_PRINTS                 0
#define DEVICENAME                          "JA_TEST"
#define DEVICENAME_WRITABLE                 1
***** code here *****

'//*************************************************
'// Library Import
'//*************************************************
#include "lib\cli.upass.vsp.sblib"
'//*************************************************
'// Global Variable Declarations
'//*************************************************
DIM text$
'//*************************************************
'// Function and Subroutine definitions
'//*************************************************

'//=================================================
'// CLOSE UART
'//=================================================
Function FuncClose()
  DIM rc
  UartClose()
  rc = GPIOSetFunc(21,2,1) '// TX
  rc = GPIOSetFunc(23,2,0) '// RTS
  TIMERSTART(1,10,0)
ENDFUNC 1
```

```
'//=================================================
'// Receive data
'//=================================================
function MyHandlerLoop()
  //BleVspUartBridge()
  DIM n, rc, tempo$, tx$
  DIM pos, return$
  // Wait return from received data
  tx$ = "0D"
  return$ = StrDehexize$(tx$)
  tempo$ = ""
  n = BleVSpRead(tempo$,20)
  text$ = text$ + tempo$
  pos = STRPOS(text$,return$,0)
  IF ( pos >= 0 ) THEN
    DIM i, x
    pos = pos - 1
    FOR i = 0 TO  pos
      x = StrGetChr(text$,i)
      rc = StrSetChr(text$,x+1,i)
    NEXT
    rc = BleVspWrite(text$)
    text$ = ""
  ENDIF
endfunc 1

'//=====================================================
'// TIMER 1
'//=====================================================
FUNCTION FuncTimer1()
  dim rc, Adr$
  Adr$ = ""
  // led on
  GpioWrite(2,1)
  rc = bleadvertstart(0,Adr$,25,25,0)
  TIMERSTART(2,25,0)
ENDFUNC 1
```

Web Links

[1] **Elektor January & February 2015**
www.elektormagazine.com/140190
Bluetooth Low Energy wireless thermometer
Remote temperature display on your smartphone

[2] **Elektor March & April 2015**
www.elektormagazine.com/140270
BL600 e-BoB: Part 1:
Wireless communication on a plate

[3] **Elektor May & June 2015**
www.elektormagazine.com/150014
BL600 e-BoB | Part 2:
Editing, compiling, and transferring a program using the Bluetooth Low Energy module

[4] **Elektor July & August 2015**
www.elektormagazine.com/150129
BL600 e-BoB | Part 3:
smartBASIC programming for the Bluetooth Low Energy module

[5] **Elektor September & October 2015**
www.elektormagazine.com/150130
BL600 e-BoB | Part 4:
The I²C port and the temperature sensor

[6] **Elektor November & December 2015**
www.elektormagazine.com/150272
BL600 e-BoB | Part 5:
SPI port & digital/analog converter
Android application

[7] https://laird-ews-support.desk.com/?b_id=1945

[8] www.elektor.com/ft232r-usb-serial-bridge-bob-110553-91

```
'//=====================================================
'// TIMER 2
'//=====================================================
FUNCTION FuncTimer2()
  // led off
  GpioWrite(2,0)
  rc = bleadvertstop()
  TIMERSTART(1,4000,0)
ENDFUNC 1
'//=====================================================
'// This handler is called when there is a BLE
message
'//=====================================================
function MyHandlerBleMsg(BYVAL nMsgId AS INTEGER,
BYVAL nCtx AS INTEGER) as integer
'// Inform libraries
ConnMngrOnBleMsg(nMsgId,nCtx)
AdvMngrOnBleMsg(nMsgId,nCtx)
select nMsgId
  case BLE_EVBLEMSGID_CONNECT
    TIMERCANCEL(2)
    TIMERCANCEL(1)
    DbgMsgVal(" --- Connect : ",nCtx)
    ShowConnParms(nCtx)
    // set at High
    GpioWrite(2,1)
  case BLE_EVBLEMSGID_DISCONNECT
    DbgMsgVal(" --- Disconnect : ",nCtx)
    // set at Low
    GpioWrite(2,0)
    TIMERSTART(1,10,0)
  ***** code here *****
endselect
endfunc 1
'-----------------------------------------------------
'// TIME OUT = nothing
'-----------------------------------------------------
Function MyHandlerTimOut() as integer
  'NOTHING
EndFunc 1
'//***************************************************
'// Handler definitions
'//***************************************************
ONEVENT EVTMR1 CALL FuncTimer1
ONEVENT EVTMR2 CALL FuncTimer2
OnEvent EVTMR0  Call FuncClose
OnEvent  EVBLEMSG           call MyHandlerBleMsg //
EVBLEMSG indicate when signifficant BLE event occurs.
OnEvent  EVBLE_ADV_TIMEOUT  Call MyHandlerTimOut
//all events have the same handler
OnEvent  EVVSPRX            call MyHandlerLoop //
EVVSPRX is thrown when VSP is open and data has
arrived
OnEvent  EVUARTRX           call MyHandlerLoop //
EVUARTRX  = data has arrived at the UART interface
OnEvent  EVVSPTXEMPTY       call MyHandlerLoop
OnEvent  EVUARTTXEMPTY      call MyHandlerLoop
‘//***************************************************
‘// main
‘//***************************************************
text$ = ““
// pin 2 output at low
rc = GPIOSetFunc(2,2,0)
rc = bleadvertstop()
IF (ENABLE_DEBUG_PRINTS == 1) THEN
  UartRsp(0)
  TIMERSTART(2,10,0)
ELSE
  PRINT “low Energy”
  TIMERSTART(0,1000,0)
ENDIF
WaitEvent
```

Active Crossover for Active Loudspeakers

With steep 3-way filters

By Alfred Rosenkränzer (Germany)

Although active loudspeakers are significantly more complicated than their passive counterparts, they have clear advantages in terms of audio technology and the resulting sound. That may sound familiar to attentive audio enthusiasts, and rightly so. Here we present a crossover network from the author of the previously published high-end amplifier project for active speakers. This gives you everything you need (in electronic terms) to build your own high-end loudspeakers.

Specifications

- Balanced input: 0.25’ inch phone jack or XLR
- Typical input signal level: 0.5–1 V_{rms} (line level)
- Input impedance: 100 kΩ
- THD + noise per channel (BW = 22 kHz):
  - Bass: 0.0003% at 100 Hz
  - Midrange: 0.0003% at 2 kHz
  - Treble: 0.0005% at 7 kHz
- Supply voltage: ±16 to ±25 V
- Current consumption: 65 mA typical

The author’s ‘High-End Amplifier for Active Speakers’ [1] project was published in the March 2014 edition of Elektor. As the name suggests, it is more oriented to high quality (excellent specs) than to high power. The amplifier uses Hitachi MOSFETs with low transconductance and relatively high channel resistance. These devices have seen use in many high-end power amplifiers over several decades and are still readily available. The design provides small to medium output power with very low distortion, which is perfectly complemented by the top-end active crossover design presented here.

You might question whether 30 watts is enough for an active loudspeaker, since some people say you need at least 100 watts. That’s a question you have to answer for yourself, but if you want to listen to music at home with exceptionally good quality and are not looking for a sound system to fill a gymnasium, all you really need is a few dozen watts. Power madness is a known characteristic of people nowadays who are less than expert in audio matters and let themselves be distracted by big numbers that are not really relevant.

Why active?

You need more than just a good amplifier to build a good active loudspeaker,

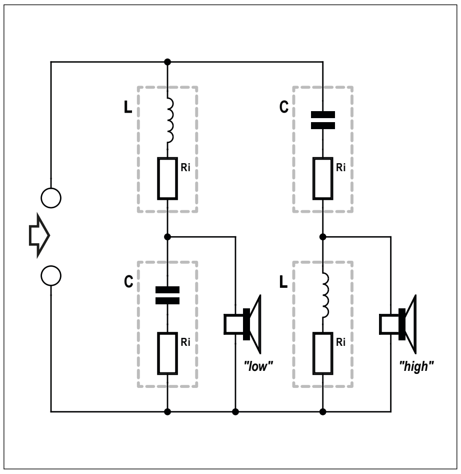


Figure 1. Circuit diagram of a passive two-way crossover, including the internal resistances of the coils and capacitors.

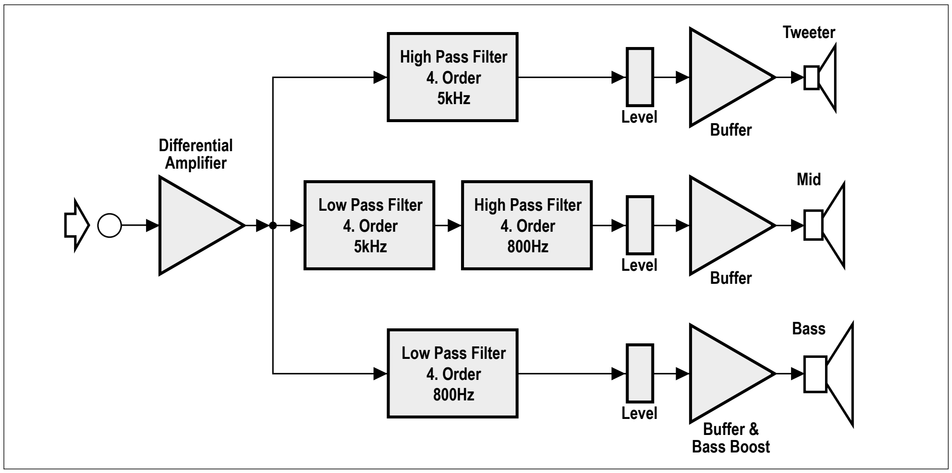

Differential Amplifier
High Pass Filter 4. Order 5kHz
Low Pass Filter 4. Order 5kHz
High Pass Filter 4. Order 800Hz
Low Pass Filter 4. Order 800Hz
Level
Level
Level
Buffer
Buffer
Buffer & Bass Boost
Tweeter
Mid
Bass

Figure 2. Block diagram of the active crossover.

at least if you want to take full advantage of the concept. The idea of mounting a good amplifier in a cabinet with good speakers and then distributing the audio power to the tweeter, woofer and possibly a midrange unit through a passive crossover is actually a bit absurd. To see why, consider the simple case of a passive two-way crossover as illustrated in **Figure 1**. Passive crossover networks consist of coils and capacitors. These components have internal resistances (R_i) in series with the speakers that reduce the damping factor. With the author's high-end amplifier, for example, the very high damping factor of 400 or more can only be achieved if the speaker is connected directly to the amplifier output with the shortest possible leads. With a woofer, a coil resistance of just 0.1 Ω in the crossover reduces the damping factor by a full order of magnitude. Capacitors also have an effective series resistance, even if it is lower than the coil resistance. The AC impedances of the coils and condensers, which must be in the ohms range for proper crossover operation, also come into play in the transition regions near the crossover frequencies. That makes the damping versus frequency characteristic nonlinear.

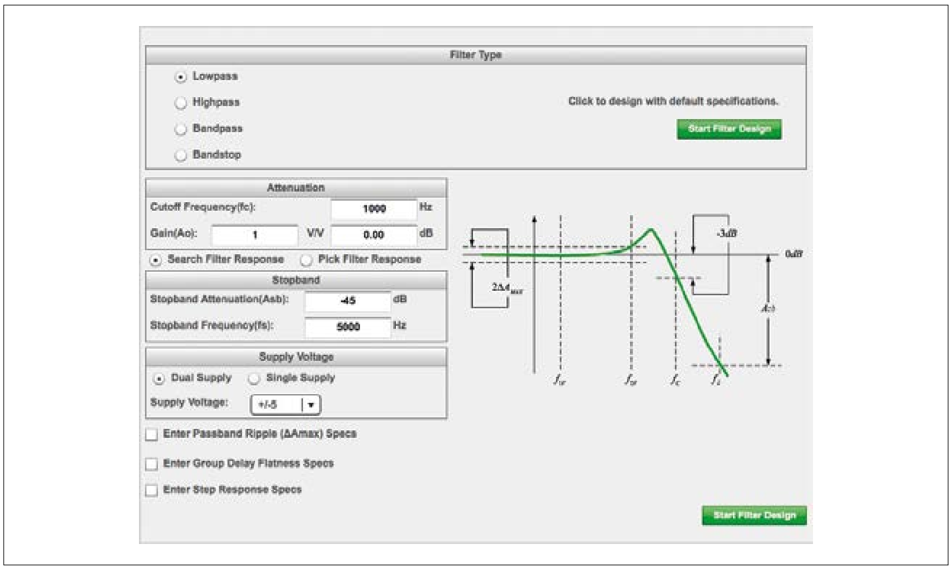
On top of this you have nonlinearities from the core materials of low-cost coils, cheap electrolytic capacitors and other low-grade components. Components with better characteristics, which means high-inductance air-core coils and high-capacitance film capacitors, are expensive and soak up some of the cost savings compared to a more complex active design. And that's by no means all: another factor is that filters with steep edges cannot be implemented with simple passive circuits, so you need even more coils and capacitors. This makes them even more expensive, and it increases the negative impact of the passive components because there are more of them in the signal path. With active filters, steep filter edges — which means stronger separation of the different frequency ranges — can be implemented easily and economically The active crossover described here has filter slopes of 24 dB per octave, which is virtually impossible with any reasonable passive network.

Another significant drawback of passive crossovers, although by no means the last, is that they are difficult to adapt to speakers with different sensitivities. If you look at the manufacturer's data sheets for various speakers, you will see a wide range of SPL ratings. A difference of just 1 dB/W/m has a perceptible effect on the sound and is by no means negligible. The sound, as expressed by the frequency response, also depends on the specific location of the speaker and the furniture in the room. With a passive system, the only way to match the SPLs of the speaker units in the cabinet is to put resistors in series with the speakers to reduce the power fed to them. This amounts to replacing one evil with another because it not only degrades the damping but also alters the frequency response of the filter arrangement. All in all, building a good passive crossover network is pretty complicated when you consider all the details. By contrast, active crossovers are a lot easier to design and the speaker SPLs can be matched very simply without interactions between the channels.

The circuit

Although passive loudspeakers with good sound are certainly available (at a price), from an engineering perspective it is unquestionably clear that a good active crossover is better than a very good passive crossover. If you want superior sound, there's no getting around the extra complexity. In this case, extra complexity means that you need a separate amplifier for each speaker. For a two-way system, you therefore need two amplifiers, and for a three-way system you need three amplifiers. And of course you also need suitable circuitry to distribute the audio signals. **Figure 2** shows the block diagram of high-end active three-way crossover.

With modern analog electronics and currently available high-quality components, building a high-end active crossover is not especially difficult. You need to be able to assess the characteristics of the components, in particular the opamps for the filter circuits, and you need to be familiar with filter design. However, nowadays you do not have to be a math genius to calculate the right component values for the desired frequency responses, since there are user-friendly programs available to do the job for you. The author used the Filter Pro Desktop tool from Texas Instruments for this purpose. If you enjoy doing this sort of calculation, we recommend its successor WEBENCH Filter Designer



Filter Type

- ● Lowpass
- ○ Highpass
- ○ Bandpass
- ○ Bandstop

Click to design with default specifications.

Attenuation

Cutoff Frequency(fc): 1000 Hz

Gain(Ao): 1 V/V 0.00 dB

● Search Filter Response ○ Pick Filter Response

Stopband

Stopband Attenuation(Asb): -45 dB

Stopband Frequency(fs): 5000 Hz

Supply Voltage

● Dual Supply ○ Single Supply

Supply Voltage: +/-5

□ Enter Passband Ripple (ΔAmax) Specs

□ Enter Group Delay Flatness Specs

□ Enter Step Response Specs

Figure 3. Screenshot of WEBENCH Filter Designer, an online tool for filter calculations.

[2] from the same source. The latter is a web app (see **Figure 3**), so no software installation on your computer is necessary. The author wanted a filter skirt slope of 24 dB per octave and opted for the well-behaved Sallen-Key filter topology [3] with the Linkwitz characteristic, which corresponds to a cascade of two normal Butterworth filter stages. Now let's look at the implementation.

ICs and power supply

The most striking aspect of the circuit diagram in **Figure 4** is the large number of opamps. The opamps here are readily available NE5532A dual opamps [4], which have very low noise characteristics and are an outstanding choice for high-quality audio filters. All of the circuitry is powered by balanced ±12 V supply voltages from a pair of conventional voltage regulators IC8 and IC9. Capacitors C45 to C49 ensure stable, low-noise supply voltages everywhere on the board, while capacitors C31 to C44 decouple the individual ICs.

The crossover can be powered directly from the power supply for the power amplifier(s) thorough connector K5. This fits well with the ±25 V supply voltage level for a high-end amplifier with 34 W output power. In that case diodes D5

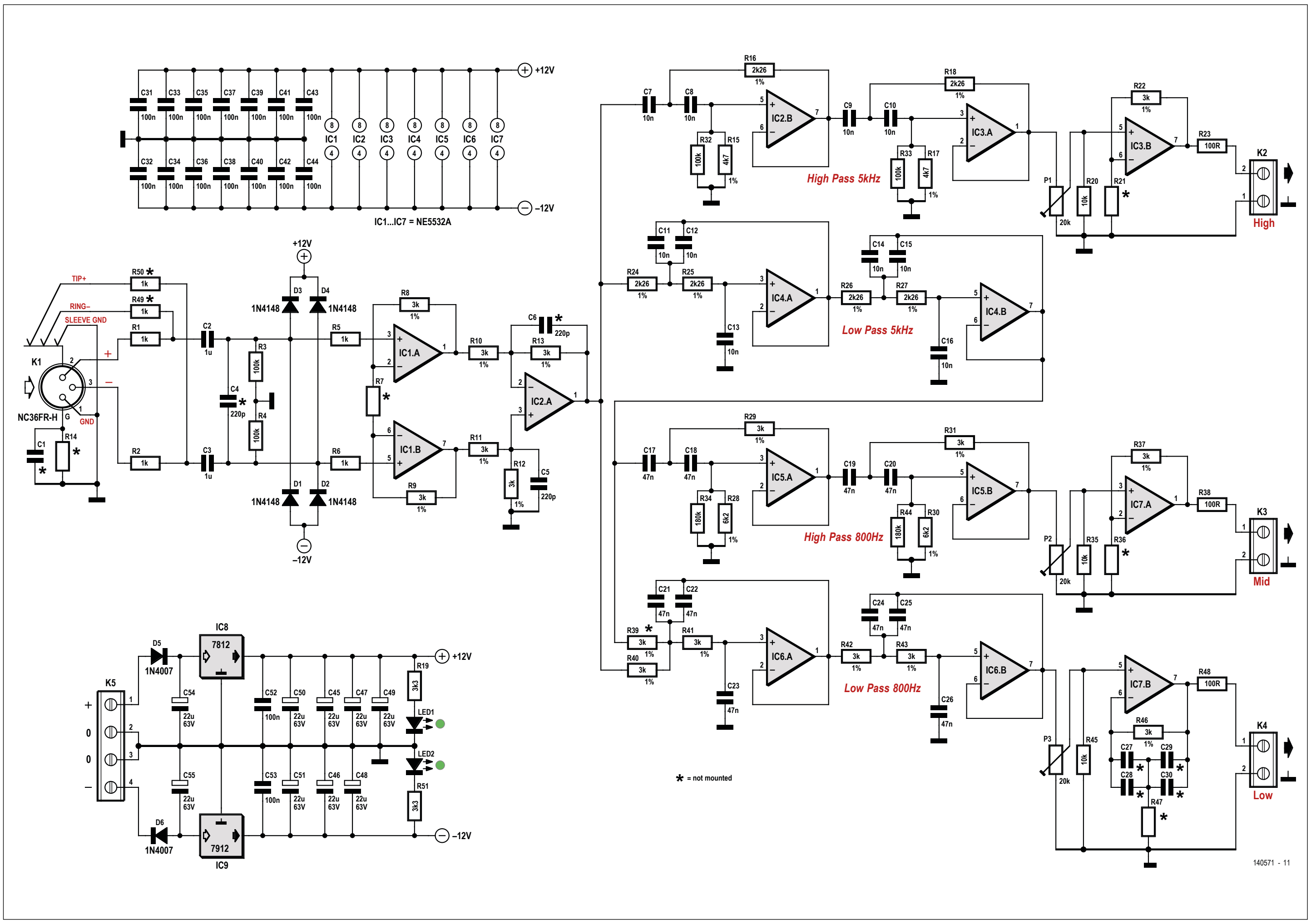


Figure 4. The main components of the active crossover circuit are opamps.

and D6 simply provide protection against reverse polarity connection. With a typical current consumption of ±65 mA (maximum ±100 mA), voltage regulators IC8 and IC9 require heat sinks at relatively high input voltages, and the maximum input voltage rating of the 7812 is 35 V. In that case a better solution is a separate power supply consisting of a 4 VA power transformer with dual 12 V secondaries and full-wave rectification and filter capacitors rated at least 220 µF , 35 V, which will deliver approximately ±17 V unregulated. The author runs his own high-end amplifiers at ±48 V and gets a bit more than 100 W out of them with proper cooling. A separate power supply for the crossover is essential in that situation.

Input stage

The signal input at K1 has a few special features. The input is balanced and is equipped with a special combination connector for K1 that can accept a 6.3 mm phone plugs (for stage use) as well as a conventional three-pin XLR plug. The XLR option is the preferred choice, and you can omit resistors R49 and R50 if you use this option. The housing of the XLR connector can be grounded for audio signals through R14 (a few tens of ohms) and C1 (a few tens of nanofarads) to improve shielding. Compared to the simpler approach of direct grounding, these components help to avoid hum-inducing ground loops. The balanced audio signal passes through R1, R2, and DC blocking capacitors C2 and C3 to the inputs of a conventionally configured instrumentation amplifier built around IC1.A, IC1.B and IC2.A.

The four diodes D1–D4 protect the opamp inputs against voltage spikes. The network formed by C4, R1 and R2 blocks any high-frequency noise at the input. The gain of the instrumentation amplifier with R7 not fitted is 1 (inverting), but the magnitude of the gain can be increased if necessary by fitting R7. For example, a value of 3 kΩ for R7 raises the gain to 3. A few words about the resistor values are in order here. For ideal common mode suppression, resistors R10 to R13, and in addition R8 and R9 if R7 is fitted, must be matched as closely as possible. Metal film resistors are therefore not an unnecessary luxury, and they have the additional advantage that they produce less noise than less expensive carbon resistors. You should also bear in mind that low resistance values increase the load on the opamp outputs and thus increase distortion, while high resistance values cause higher noise levels. The best compromise is somewhere in the single-digit kΩ range; here the author opted for 3 kΩ. Capacitors C5 and C6 limit the frequency range to roughly 240 kHz and are not absolutely necessary (the same is true of C4).

Filters

The output of IC2.A provides an unbalanced audio signal with low source imped-

Component List

Resistors

All 1% = metal film
R1,R2,R5,R6,R49,R50 = 1kΩ 5%
R3,R4,R32,R33= 100kΩ 5 %
R7,R14,R21,R36,R47 = optional *
R8..R13,R22,R29,R31,R37,R39–R43,R46* = 3kΩ 1%
R15,R17 = 4.7kΩ 1%
R16,R18,R24,R25,R26,R27 = 2.26kΩ 1%
R19,R51 = 3.3kΩ 5%
R20,R35,R45 = 10kΩ 5%
R23,R38,R48 = 100Ω 5%
R28,R30 = 6.2kΩ 1%
R34,R44 = 180kΩ 5%
P1,P2,P3 = 20kΩ preset, horizontal *

Capacitors

C1,C4,C5,C6,C27–C30 = optional *
C2,C3 = 1µF 63V, PET, 2.5 or 5mm pitch
C7–C16 = 10nF 5%, PET, 5 or 7.5mm pitch *
C17–.C26 = 47nF, 5%, PET, 5 or 7.5mm pitch *
C31–.C44,C52,C53 = 100nF, X7R, 5mm pitch *
C45–C51,C54,C55 = 22µF, 63V, radial, 7mm diam., 2.5mm pitch

Semiconductors

D1–D4 = 1N4148
D5,D6 = 1N4001
LED1,LED2 = LED, green, 3mm
IC1–IC7 = NE5532A, DIL case
IC8 = 7812
IC9 = 7912

Miscellaneous

K1 = 3-way XLR, 6.3mm, socket, Neutrik NCJ6FA-H
K2,K3,K4 = 2-way PCB screw terminal block, 0.2'' pitch
K5 = 4-way PCB screw terminal block, 0.2'' pitch
Thermal insulation washer and collar for IC9
Platine 140571-1 *

* see text

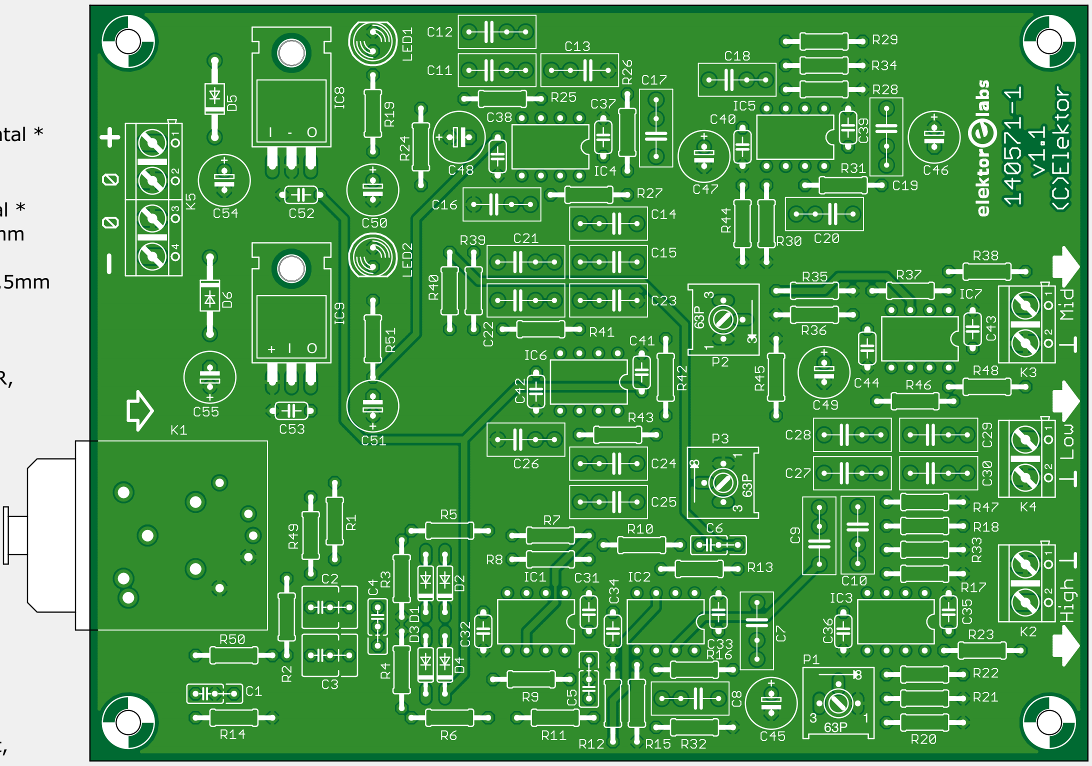


Figure 5. Circuit board layout for the active crossover. The biggest component is the combined phone jack and female XLR connector.

ance for splitting into several frequency ranges. First let's look at the high-pass filter for the treble channel.

A Sallen-Key filter requires two resistors, two capacitors and an opamp. It achieves a skirt slope of 12 dB per octave. To obtain the required steepness of twice this value, you have to connect two of these filter stages in series (IC2.B and IC3.A). The author calculated the filters so that the capacitors in both stages have the same, commonly used values. That's because capacitors are not generally available with the same finely graduated values as resistors and they are more expensive than resistors, so the necessary non-standard values are obtained by connecting resistors in parallel. The filter stages are followed by a buffer stage with adjustable gain, which is discussed later.

For the midrange channel there is first a low-pass stage with the same corner frequency as the high-pass filter of the treble channel, so the signal levels add to form a smooth frequency characteristic in the transition region. The low-pass stage is followed by a high-pass stage with a corner frequency corresponding to the crossover frequency between the woofer and the midrange speaker.

For the bass channel we only need a pair of low-pass stages with a corner frequency corresponding to the crossover frequency. Resistor R39 is normally not fitted. The output stage of the bass channel has another special feature.

Output stages

As already mentioned, the signal level can be adjusted for each speaker with the three trimpots P1, P2 and P3. The output stages built around IC3.B, IC7.A and IC7.B normally have unity gain, but the gain can be increased if necessary by fitting resistors R21, R36 and R47. However, this should only rarely be necessary. There is another special feature in the bass channel: you can boost the bass by fitting C27–C30 and R47, which increase the signal level to counter the 6 dB per octave rolloff of the woofer amplitude below its corner frequency, which depends on how the woofer got installed in the cabinet and other factors. The frequency range is determined by the ratio of the C27–C30 values to the resistor values, while the amount of bass boost is determined by the ratio of R46 to R47. See the "**Bass boost**" inset for information about suitable values for usable frequencies and boost levels.

Particularly with relatively small loudspeakers, this measure can significantly improve bass reproduction. Of course, you can't get something for nothing. If you implement bass boost, you use a significant part of the amplifier output power to compensate for the low-frequency rolloff. This means you trade off usable volume range for better sound, since the amplifier as well as the woofer reach their limits at lower signal levels. With larger loudspeakers or when you want higher volume levels, you are therefore better off to avoid bass boost.

Construction and connection

Thanks to use of dual opamps, the component density on the circuit board (see **Figure 5**) is a lot less than you would expect from the schematic diagram. The PCB is available from the Elektor Store, and the layout files can be downloaded for free from the project web page [5]. To simplify board assembly and manual soldering, we avoided using SMT components. You should first mount the six diodes and all of the resistors, followed by the IC sockets (high recommended), ceramic capacitors and film capacitors. IC8 can be screwed directly to the board as shown in the picture because almost all of top side of the two-sided board is covered by a copper ground plane. This is not possible with IC9 because the metal pad of the package is connected to the input lead. You therefore need an insulator and a shoulder washer for the screw. Finally, solder the LEDs, electrolytic capacitors, screw terminal blocks and input connector K1 to the board. The components marked with an asterisk (*) are optional.

After checking for cold solder joints and shorts, you can apply power to the board. Both LEDs should light up now. To be on the safe side, you can measure the supply voltages (±12 V) on an IC socket to verify that they are in spec. Now switch off the power, insert the ICs in their sockets (pay attention to the right orientation), and switch on the power again. If the circuit does not draw more than about ±100 mA, you can assume that everything is okay and the board is ready to be installed in

Bass Boost

If you want to boost the loudness of the woofer in the low-frequency range, where there is usually a natural amplitude roll off of 6 dB per octave due to physical dimensions and installation parameters, you can simply increase the gain in this low-frequency range. That can extend the low end of the loudspeaker frequency response by as much as two octaves. Of course, this only makes sense if you are building fairly small loudspeakers, since reasonably large loudspeakers are linear down to 25 Hz without any extra help. However, a bass boost at 73 Hz is certainly suitable for really small shelf loudspeakers about the size of satellite loudspeakers. Even if the data sheet of you woofer provides useful information, it's worth remembering that listening experiments are a good substitute for highly complex measurements.

If you want to implement bass boost, you can use the accompanying table for guidance. First choose a boost frequency *f* that is 1 to 1.5 octaves below the corner frequency of the woofer. Then the individual columns correspond to the desired boost in dB. In the circuit, change the value of R46 to 6.8 kΩ. The values in the table are the capacitance (in microfarads) of C27/C28 in parallel or C29/C30 in parallel. This way you can obtain non-standard values by combining capacitors with standard values from the E12 series.

Table 1

f	8.1 dB	6.1 dB	4.4 dB
(Hz)	*R47 = 2.2 kΩ*	*R47 = 3.3 kΩ*	*R47 = 5.1 kΩ*
73	0.56	0.47	390
60	0.82	0.56	0.47
50	0.82	0.68	0.56
40	1.00	0.82	0.68
33	1.20	1.00	0.82
28	1.50	1.20	1.00

the loudspeaker cabinet and wired up. It should go without saying that with audio signals the wires and cables should be kept as short as possible. The best option is to mount the board flush with the metal rear face of the plug-in amplifier module, so that K1 peeks out thorough a suitable hole in the cabinet.

Adjustments and settings

The component values shown on the schematic diagram in Figure 4 are dimensioned for crossover frequencies of 800 Hz and 5 kHz. If you want to use different frequencies, you must change the values of the frequency-determining components accordingly. If you do this, you should either change the capacitor values or the resistor values, but not both. To reduce the crossover frequency by a factor of 2, for example, a good option is to essentially double the values of C7–C16 to 22 nF. The same applies to C17–C26; here a value of 100 nF would be suitable. This results in crossover frequencies of 380 Hz and 2.3 kHz. Of course, you can also work with the filter design tool [2].

The crossover does not have to be built as a three-way type. To build a two-way version, you can simply omit the components for the midrange channel, and of course the corner frequencies of the treble and bass channels will have to be the same. With this option you do not need components C11–C20, R24–R31, R36, R38, P1, IC4, IC5 and K3.

If desired, you can also reduce the steepness of the filter skirts. If 12 dB per octave is sufficient, simply omit the second stage of the high-pass and low-pass filters. In that case you do not need components C14–C16, C24–C26, R17, R18, R30, R31, R33 and R44, and components C9, C10, C19, C20, R26, R27, R42 and R43 must be replaced by wire bridges.

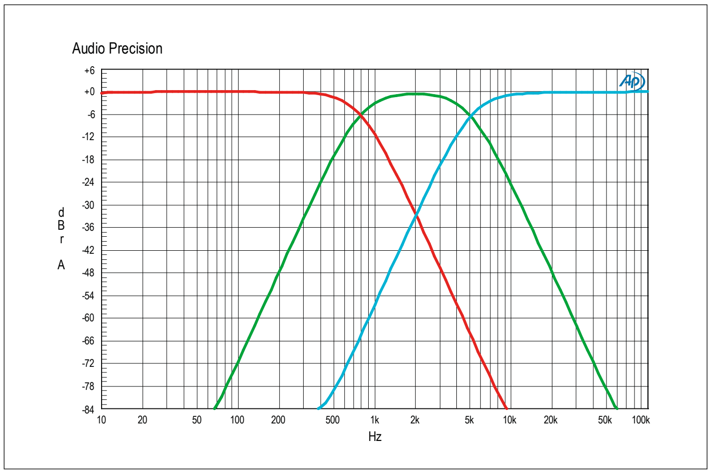
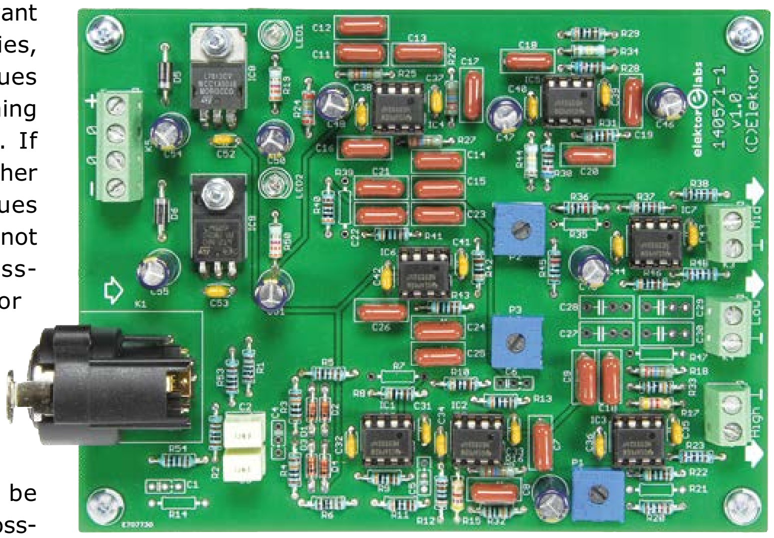


Figure 6. Frequency response of the active crossover.



About the Author

Alfred Rosenkränzer has been working as a design engineer for 31 years, initially in the field of professional television engineering. He has been developing and testing high-speed digital circuits and analog circuits for IC testers since the late 1990s. Audio is his personal passion.

Web links

[1] High-End Amplifier for Active Speakers, Elektor March 2014: www.elektormagazine.com/130007

[2] WEBENCH Filter Designer (TI): http://bit.ly/1SjFWCA

[3] Sallen-Key filters: http://en.wikipedia.org/wiki/Sallen–Key_topology

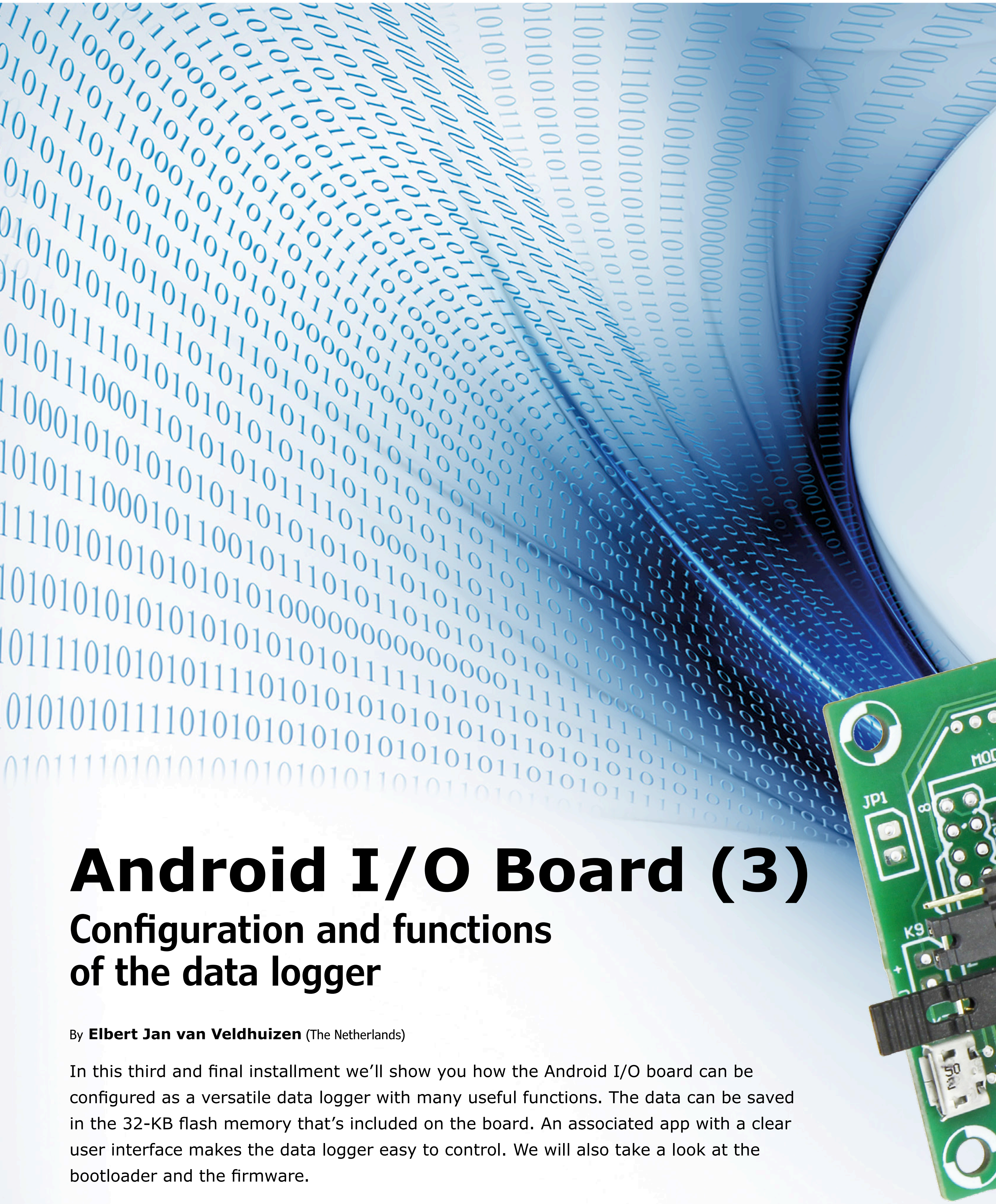
[4] NE5532A data sheet: www.ti.com/product/ne5532a

[5] Active Crossover project web page: www.elektormagazine.com/140571

If your preamplifier does not have a balanced output, you can also operate the crossover with an unbalanced input signal. In that case the input signal is connected to pin 2 of the connector, with pins 1 and 3 joined together.

Figure 6 shows the measured frequency response of the crossover, which is picture perfect but nevertheless real. If necessary, you can use trimpots P1, P2 and P3 to adjust the relative levels of the three speakers. Set the trimpot of the speaker with the lowest SPL rating to the maximum value. Then adjust the other two trimpots to obtain the same sound level. For the technically inclined, this adjustment can be made using test signals (pink noise, for example) and a microphone at the listening location to measure the sound pressure level, but that takes a certain amount of effort. Making the adjustments by ear with music should actually yield good results, since good sound is ultimately determined by what sounds good to you. With this approach, start by setting all three trimpots to the maximum level and then adjusting them a bit at a time until you do not hear any more improvement. Trimpots have a number of undesirable properties, including ageing and noise. You can either buy expensive high-quality types, or you can unsolder them after completing the adjustment and measure their resistance values. Then you can alter the values of R20, R35 and R45 as necessary and solder suitable fixed resistors between the pads for the wipers and top ends of the trimpots. ⏮

(140571)



Android I/O Board (3)

Configuration and functions of the data logger

By **Elbert Jan van Veldhuizen** (The Netherlands)

In this third and final installment we'll show you how the Android I/O board can be configured as a versatile data logger with many useful functions. The data can be saved in the 32-KB flash memory that's included on the board. An associated app with a clear user interface makes the data logger easy to control. We will also take a look at the bootloader and the firmware.

The Android I/O board includes a 32-KB serial flash memory. This can be used to store standalone measurements. For example, if two analog channels are read once per minute, there is enough memory to store the data for well over a week.

There is also a mode where statistical functions are carried out on the data over a longer period of time, with only the results being stored. As an example, it could be set up to take a measurement every 0.9 of a second. Once per hour the average, the standard deviation, the minimum and the maximum of the 4096 measurements are calculated and saved. When measuring the temperature, for example, this provides more than enough information to keep track of all significant events. There is enough memory to store about nine months' worth of data when one analog channel is used.

You can log a maximum of eight different pins at the same time, for example, four ADCs, the counter, a CPS and two digital inputs. When reading the data, you can use any of the settings for the pins, as described in Part 1. When reading data, you can use any of the settings for the pins, as described in Part 1. For example, you can reduce noise of the ADC by using several measurements for each reading, or you can change the reference voltage.
The firmware can be made to stop when the memory fills up, or it can continue and overwrite the memory in a loop. The data logger has been designed to continue at the memory location where it left off in the event of a power-cut.

The data logger is controlled via the S0, S1 and S3 status registers.

To log a pin, bit 7 in the status-1 register for that pin has to be set. As mentioned earlier, other settings can be used as well. For example, to use pin B2 as ADC with a 2.048 V reference voltage, you first have to send *s b2 5#s b21 129*. To start the data logging, the value 1 should be written to register S3. Bit 0 of the status S0 register will then be set. To stop logging the value 0 should be written to S3. Bit 0 of S0 will then be reset as well. To make the data logger continue from where it left off, the value 2 or 3 should be written to S3. With a value of 2, the data logger will continue in the 64-byte block (see below) where it left off (this will be at the beginning if the microcontroller was reset). With a value of 3, the data logger always starts in a new 64-byte block.

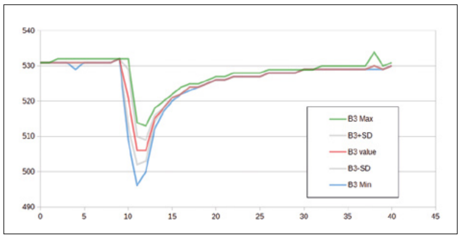
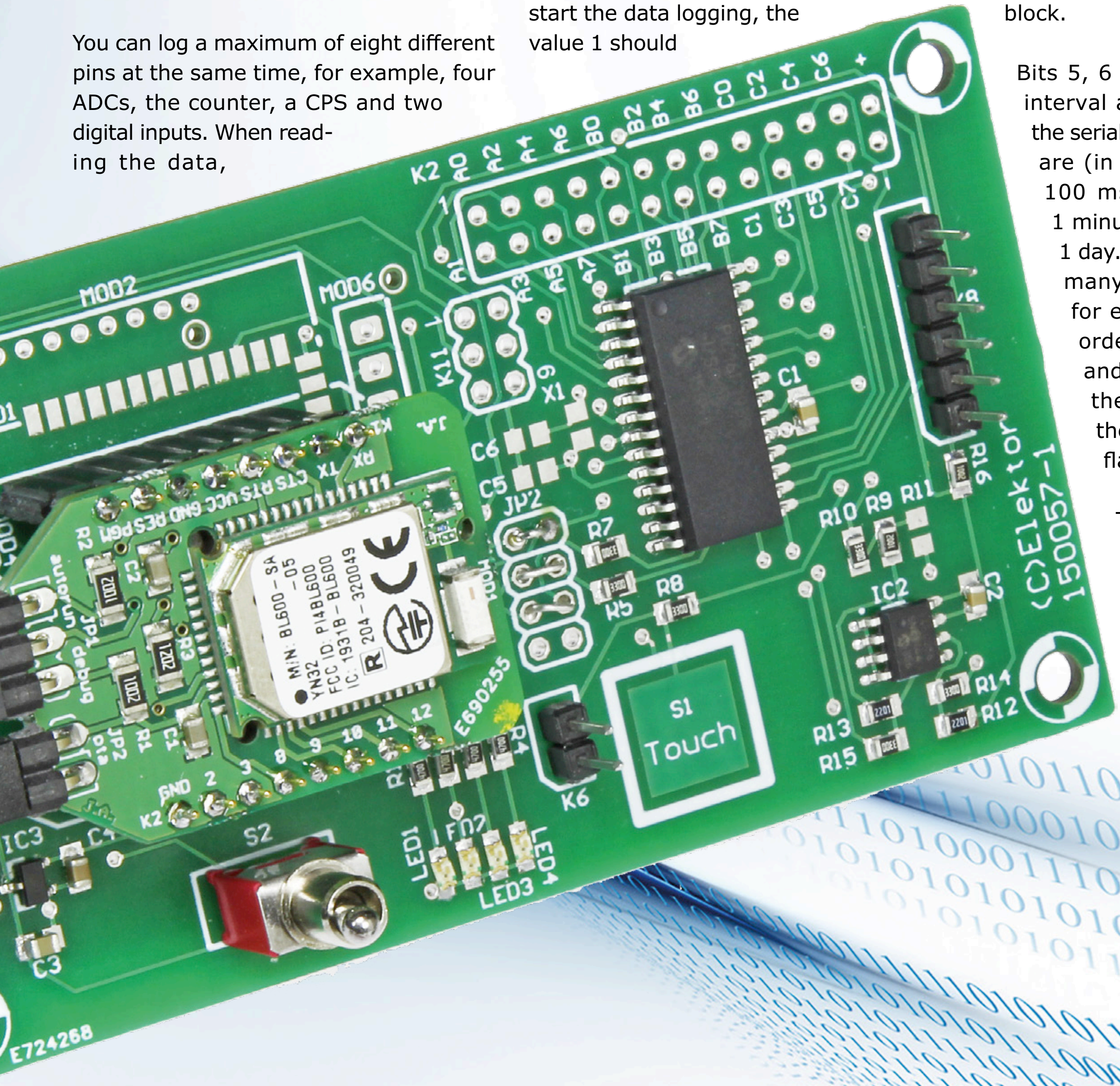


Figure 1. The available statistical functions.

Bits 5, 6 and 7 are used to set the interval at which data is written to the serial flash memory. The intervals are (in increasing order): 10 ms, 100 ms, 1 second, 15 seconds, 1 minute, 15 minutes, 1 hour and 1 day. Bits 3 and 4 determine how many measurements are taken for each reading. In increasing order, these are: 1, 64, 4,096 and 4,194,304. When bit 7 of the S1 status register is set, the LED on pin B0 is made to flash for each measurement.

The data logger can store the average value of the measurements (the sum of all the measurements, divided by the number of measurements) or the cumulative value (the sum of all the measure-



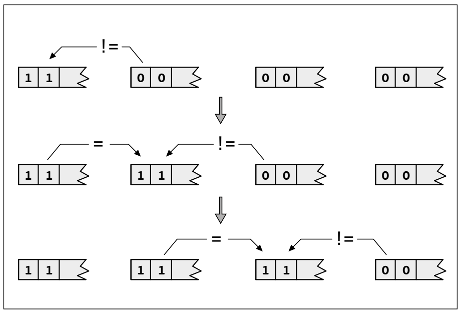


Figure 2. The first two bits in the RunID.

Byte n	9	8	7	6	5	4	3	2
Byte n+1	1	0	9	8	7	6	5	4
Byte n+2	3	2	1	0	9	8	7	6

Figure 3. Compacting the bits; this example is for bits 9 – 0 of the ADC measurements.

ments). It can also store the minimum and maximum values of the measurements (see **Figure 1**). A final function lets it calculate and store the standard deviation of the measured values. The first five bits of status-1 register S1 are used to set this up for all pins. There is a special setting for cases where the input value consists of a single bit (a digital input or the CPS). The average (as an integer value) doesn't make much sense here. When both bits 0 and 1 of S1 are set, the sum of the measurements is saved for one-bit values, and for values of more than one bit it's the average.

It is not always efficient to store the average value, maximum value, minimum value and the standard deviation. For data with a normal spread, the maximum and minimum values are usually two times the standard deviation plus or minus the average value. When you're expecting the data to have a normal spread, it's sufficient to store just the average value and the standard deviation. For example, when the temperature in a fridge is measured, where you mainly expect peaks of higher temperatures, the average and maximum values could be stored (for an NTC this would be the lowest resistance value). This would also give an indication of the length of the peak; the longer the duration of the peak is, the closer the average is to the maximum.

Not all settings will give good results. This is partly due to the number of bits used for some of the variables. The sum of the measurements is stored in 32 bits (maximum value: 4,294,967,295) and the minimum and maximum values use 16 bits (65,535). When only the average or the sum is stored from the ADC in the data logger, it's possible that up to 4,194,304 measurements are taken for each data item. The standard deviation is calculated by taking the square of the sum of the measurements that are stored in a 32-bit register. When the standard deviation of the 10-bit ADC is stored, it can only deal with up to 64 measurements. In that case, the average value of the ADC measurements must also be calculated (via the status-1 register of the ADC).

The time taken for all these operations should also be taken into account. Storing a block of 64 bytes takes 5 ms. Calculating the standard deviation takes 2.5 ms per channel, with the measurements themselves taking a fraction of a ms. As an example, when a write period of 10 ms is set up for 4 channels, with 64 measurements and standard deviation, the effective data rate will be less than half that, with a lot of jitter. The presence of this jitter can be detected (see below). When an automatic restart is configured via bit 2 of the S0 status register, it is important that this setting and the settings for the pins are saved in the EEPROM of the microcontroller (using command 2 in the Z3 status register). Otherwise these settings will be 'forgotten' when the microcontroller restarts. In addition, bit 2 of the Z0 status register should be set, so that these settings are indeed read. The data logger app reads the raw data from the serial flash memory. It also needs to know the structure used for storing the data in this serial memory.

Serial flash memory

We decided to use a 24FC256 for the flash memory. This IC is accessed via I²C and can be controlled via two pins by the microcontroller. The I²C protocol uses two pins (C3 and C4) that have 'open collector' outputs (the line is pulled to GND or kept high via a pull-up resistor to Vcc). This method avoids the possibility of short circuits when two ICs try to send data at the same time. One line is used for the clock, the other for data. The state of the data line is read when the clock changes from low to high. The maximum data rate is 1 Mb/s.

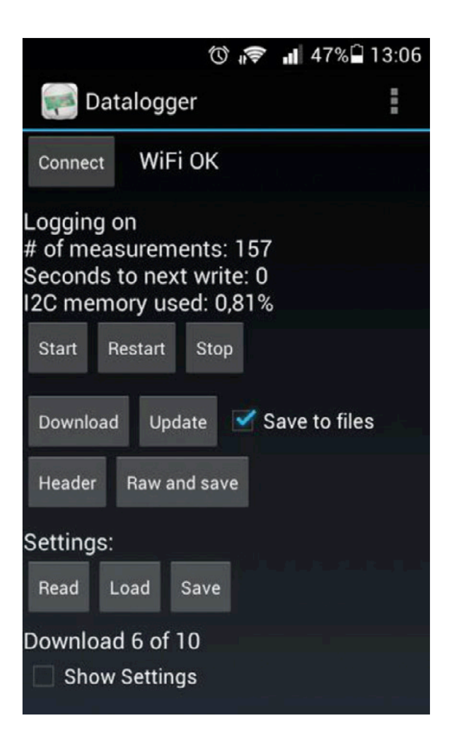
When using the 24FC256, you have to bear in mind that the flash memory wears out when data is written. When you write just a single byte, a complete row of 64 bytes is first saved locally. The whole row is then erased, after which the new data is written. If you were to store each byte in a 64-byte row individually, then all bytes in that row would be erased 64 times. The flash memory has an operating life of at least 1 million erase/write cycles. With this method you would therefore reduce the life of the 24FC256 to 15,625 write operations per device.
A better way to store the data is to make the microcontroller buffer the 64 bytes for one row, and then write all 64 bytes to the flash memory in one operation. This improves the operating life of the 24FC256 by a factor of 64 times.
To further reduce the wear of the flash memory, the current write position of the data is not stored in the flash memory itself, but in the RAM of the microcontroller. However, this does give rise to a problem when the power goes off. You would then no longer know at which location in the flash memory the last data was written to. A clever trick has been used by the author to recover the position of the last data written. It uses the first two bits of every 64-byte block as a type of run identifier (runID). It works as follows: The first bit of the 64-byte block is given the same value as the second bit in the previous 64-byte block. The second bit equals the inverted value of the first bit in the following 64-byte block. **Figure 2** shows this method in action. The result of this is that all data that belongs together has bit 6 of the 64-byte block and bit 7 of the following block the same, whereas the bits will be different for old data. Using this method, you can find out where the data logger wrote the last block.

The third to the fifth bits in the same byte are used as flags. The third bit indicates that all of the memory has been written and that it's now being written for the second (or more) time. The fourth bit indicates that this block is the start of a new series (for example after a power-cut) and that there could be a discontinuity in the data. The fifth bit indicates that the measurement wasn't taken in the same interval, because the microcontroller was too busy with other tasks. This is important to note, since the measurements don't have a time-stamp associated with them. You can normally assume that the measurements were saved at the interval that was set up. But when the fifth bit is set, this is no longer the case, and some jitter occurred. The fifth bit is only reset when a new datalog run is started.

The data is compacted bit-wise before it is stored. For example, when two 10-bit values are stored, the first byte contains the highest eight bits (9 – 2), with the lowest two bits ending up in the highest two bits of the following byte. The next 10-bit value has its six highest bits (9 – 4) stored in the lowest six bits of the second byte, with the remaining four bits stored in the highest bits of the third byte, etc. This is shown in **Figure 3**. It is not possible to store more than 63 bytes per measurement. In theory, it is possible to set up the Android I/O board so that a measurement takes up more than 63 bytes. The Android I/O board then returns an error by setting bit 1 of the status byte (this can be read using R C4) and stops with logging.

It is important that the app knows exactly how many bits have been used for each channel, so that it can decipher the raw data from the serial flash memory. For this reason, the first 64-byte block of the serial memory is used to store these details. It tells us which pins have been read, what data has been stored and how many bits have been used. The exact format is shown in **Table 1**.

The flash memory can be read using the Read command on the C3 register along with the address (for example, r c3 128). The controller then returns 32 bytes. These fit exactly in the write buffer of the controller. When more data is written, the controller stops (and therefore stops logging data) until this data has been sent via the 9600-baud connection. The app must therefore wait until the 32 bytes have been received before requesting the next set of 32 bytes, otherwise jitter will occur and measurements will be missed. When addresses 32768 and 32800 are requested, the current buffer is read, instead of the serial flash memory. This is particularly useful with longer periods (one measurement per hour, for example, where the buffer may be saved only once per day), since it lets the app read the latest measurements.


47% 13:06
Datalogger

Connect WiFi OK

Logging on
# of measurements: 157
Seconds to next write: 0
I2C memory used: 0,81%

Start Restart Stop

Download Update ☑ Save to files

Header Raw and save

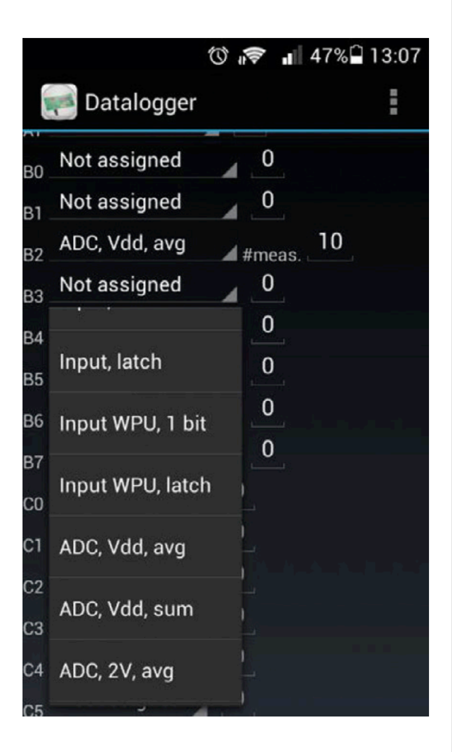
Settings:

Read Load Save

Download 6 of 10

□ Show Settings

Figure 4. The app designed for use with the data logger.


47% 13:07
Datalogger

B0 Not assigned 0
B1 Not assigned 0
B2 ADC, Vdd, avg #meas. 10
B3 Not assigned 0
B4 0
B5 0
B6 0
B7 0
C0
C1
C2
C3
C4
C5

Input, latch
Input WPU, 1 bit
Input WPU, latch
ADC, Vdd, avg
ADC, Vdd, sum
ADC, 2V, avg

Figure 5. From a dropdown menu you can select any possible option for the pins.

The status of the measurements can be requested by reading the C4 register. The hexadecimal data returned gives the status, the position (in 64-byte blocks) in the flash memory, the position in bytes and bits in the buffer, the number of measurements remaining until the next data is written to the buffer, and the number of milliseconds remaining until the next measurement. The most important bits in

Table 1. Contents of the header

Byte	Contents (first byte is LSB)
0	RunID
1	Status S0 register
2	Status S1 register
3	Number of extra bits in Sum
4,5	Total number of bits per measurement (all channels)
6,7,8,9	Number of measurements per write operation
10,11,12,13	Number of milliseconds between measurements
16	Pin number of the first channel (255= no pin)
17	Number of bits for the first channel
…	
30	Pin number of the eighth channel (255= no pin)
31	Number of bits for the eighth channel

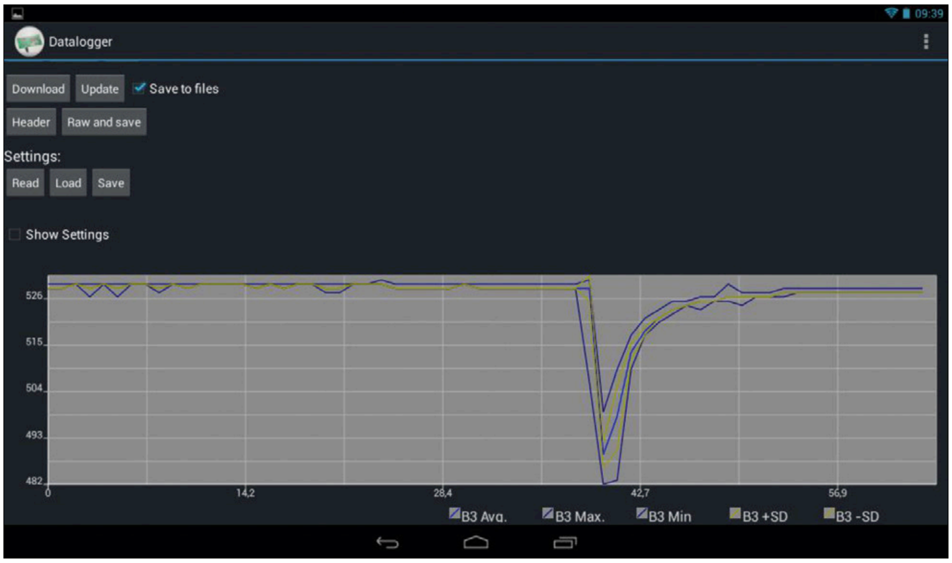


Figure 6. A graph showing the statistical functions on the measurements.

the status are bit 0 (data logging active or not), bit 2 (jitter) and bit 4 (the data has filled up the memory at least once).

Data logger app

The Android app designed for this project (**Figure 4**) can be used to configure the settings for the data logger: the frequency, the number of measurements, whether measurements loop round the buffer, and whether the data logger immediately starts measuring at power-up. We can also specify what is going to be stored: the sum or the average, the maximum value, the minimum value and the standard deviation. All the pins can also be configured here. From a drop-down menu (called a 'Spinner' in Android-talk) you can select any of the possible options for all the pins (**Figure 5**). For certain options there is a second parameter (with the ADC it's the number of measurements and with the CPS it's the threshold value). These parameters are also set up here.

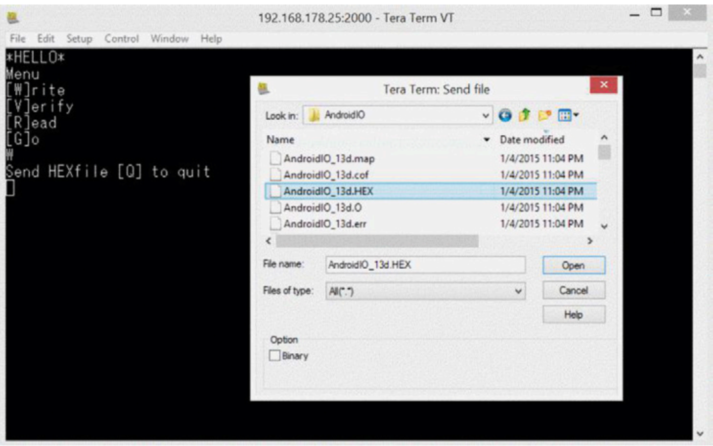
The data logger can be started and stopped via the app. It's also possible to restart the logging after it has been stopped, but only if all the functions and parameters for the pins haven't been changed. This is why the parameters aren't stored during a restart, even if different parameters were selected in the app.

The app also shows the status of the data logger: if it is running, how much of the memory has been written to, and how long it takes until the next measurement and write-action.

The data can be read in several ways: a complete download, or an update of the data since the last reading. This data is shown in a graph, and is optionally stored in a CSV file. This CSV file can be read by Microsoft Excel or LibreOffice Calc. A configuration file is written at the same time, which contains a number of important details regarding the data set, such as the pins that were read and the parameters used. These files are stored in the 'data logger' directory on the Android device. It is also possible to just read the header block, so the app can display the correct information, or to download the raw data from the serial flash memory and save it in a file.

The structure of the app is similar to that for the etch controller, which was described in Part 2. In 'onCreate' the graphical elements, the background loop and the message handler are started. The spinners for the pin functions haven't been defined in the XML files, but are created in Java in a loop. The whole layout has been put in a 'ScrollView', which automatically adds a scroll function, without needing any further programming. The same code has been used for the communications with the Android I/O Board as that used in the app described in Part 2.

This app is also quite complex because of the asynchronous transmission of commands to the Android I/O board and the reception of replies received The result is that part of the functionality is in the thread, controlled via variables that act as flags (if necessary, set from within a 'clicklistener'). Information is sent back via the message handler that updates the graphical elements. This means that the functionality for a task has been split between two or three sections of the code. This does make the program less clear, but this is inherent in Java



```
*HELLO*
Menu
[W]rite
[V]erify
[R]ead
[G]o
W
Send HEXfile [Q] to quit
```

Look in: AndroidIO

Name	Date modified
AndroidIO_13d.map	1/4/2015 11:04 PM
AndroidIO_13d.cof	1/4/2015 11:04 PM
AndroidIO_13d.HEX	1/4/2015 11:04 PM
AndroidIO_13d.O	1/4/2015 11:04 PM
AndroidIO_13d.err	1/4/2015 11:04 PM

File name: AndroidIO_13d.HEX

Files of type: All(*.*)

Option

□ Binary

Figure 7. Uploading a file to the I/O board via a terminal program on the PC.

programming.

This app also has a buffer with a built-in delay for the sending of commands. The various pin functions are effectively sent at 10 ms intervals to prevent the buffer of the Android I/O board from overflowing and losing some of the commands. This function can also be incorporated in your own apps. When data is read from the flash memory, the successive blocks are read with a delay of 120 ms in between. When programming an app it is always advisable to check if there are any libraries available that could be used. The author made use of the open-source AndroidPlot library for the drawing of the graphs. However, there are other libraries that do the same thing. If we want to edit the source in Eclipse, we must download the jar file *androidplot-core-x.x.x.jar* [1] and save it in the lib/ directory. AndroidPlot takes care of the drawing of graphs, along with axes, labels and legends, and includes a zoom function (**Figure 6**). More information about AndroidPlot can be found at [2].

Bootloader

For this project, the author has written a bootloader for the microcontroller. This bootloader is stored in a protected part of the memory, which means that the firmware cannot damage the bootloader. Thanks to this bootloader it becomes easy to write new firmware for the microcontroller and to try it out.

The protected area of the microcontroller has room for only 512 instructions. The author had to use several tricks to make the bootloader fit into this limited space, and to keep it user-friendly. For example, the text is stored as two characters per 14 bits, and the variables and code have been put together in such a way that a minimum number of bank-select statements had to be used.

The bootloader is activated by connecting pin B2 to ground when the microcontroller is powered up (either via a push-button or a jumper on K6). LED3 on RB0 will then turn on. Using a terminal program that is connected to the board via a Bluetooth, WiFi or USB connection, you will be greeted by a menu (if necessary, press the Enter key to display the menu). There are several available options (see **Figure 7**). Press on 'W' (Write) to upload a program to the microcontroller. The bootloader then asks for the hex file of the program. In, for example, TeraTerm Pro you should use 'file/send file...' to select the hex file. The hex file will then be uploaded. Since this happens at 9600 baud, this can take up to a minute and a half, depending on the file size. 'Q' is used to return to the main menu from any of the sub-menus.

With 'V' (Verify) you can use the same method to check that the hex file has been programmed correctly. With 'R' (read) we obtain a printout of the program memory from addresses 0x0200 to 0x3FFF. If there are more than four empty locations (value 0x3FFF), then the next non-empty location will be shown. For clarity, the last address will always be shown.

With 'R' (Run) the firmware is started. You can return to the bootloader by writing either 20 or 21 to Z3 in the firmware. In the first case the bootloader will only start if the bootloader key is pressed. In the second case the bootloader will always start. The bootloader can also be forced to start by sending a capital Q when the debug mode is turned on. In your own firmware you can jump to either 0x0000 or 0x0019. In both cases you must remember to first turn off interrupts and to set PCLATH and BSR to zero.

Modifying the firmware

It's possible for you to modify the firmware. There are several reasons for wanting to do this. The main one is that you may want to add some default settings to the code. This causes the ports to be set up automatically, so there is no need for them to be sent to the Android I/O board and saved in its EEPROM. Another reason is that you will need extra firmware for specific I²C and SPI ICs. Or you could add software-based support for certain formats (e.g. IR remote controls or 433 MHz modules).

You will require some knowledge of Microchip 14-bit assembly language to modify the firmware. You can use the MPLAB IDE or MPLAB X to edit the program, compile it and test it. The required source files (AndroidIO.asm and ppmacros.asm) can be downloaded from the Elektor Magazine website [1].

The structure of the firmware is shown in **Figure 8**. There are two loops. The interrupt-routine is called for all actions that require a real-time response: update the PWM, measure the CPS, read the counters, and keep time-related variables up to date, such as the flags for the ADC, the clock, the watchdog and the time itself. The text for the serial port isn't generated inside the interrupt. This always happens inside the main loop, to prevent the interrupt routine from stalling when the buffer becomes full. The main loop writes into a buffer, from where the data is fetched one character at a time by the interrupt routine and sent to the serial port (the Android device) when the UART is empty. The main loop contains a parser for dealing with the commands and (if set up) the storing of any changes (input and CPS),

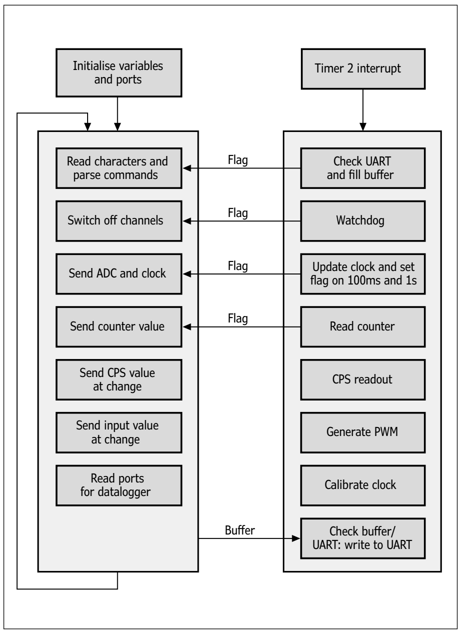

Initialise variables and ports
Timer 2 interrupt
Read characters and parse commands
Flag
Check UART and fill buffer
Switch off channels
Flag
Watchdog
Send ADC and clock
Flag
Update clock and set flag on 100ms and 1s
Send counter value
Flag
Read counter
Send CPS value at change
CPS readout
Send input value at change
Generate PWM
Read ports for datalogger
Calibrate clock
Buffer
Check buffer/ UART: write to UART

Figure 8. Structure of the firmware.

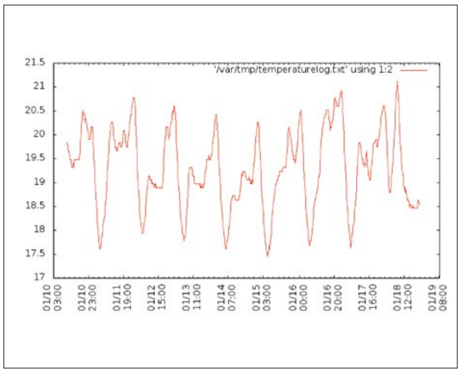


Figure 9. A graph showing the measured temperature. This was sent via email by an RPi.

Listing 1. Script for periodically measuring the temperature and for creating a graph.

```
#!/bin/bash
IOcmmd () { sleep 5; echo; echo s b3 3#; sleep 1; echo s b32 10#; sleep 1; echo r b3#;sleep 3;}
txtfile='/var/tmp/temperaturelog.txt'
giffile='/var/tmp/temperaturelog.gif'
loop=0
beta="3900"
r00='echo "10000*e(-$beta/298)"|bc -l'
while true
do
  adcvalue1='IOcmmd | telnet 192.168.178.27 2000 | grep "R B3" |tail -n 1 | awk '{ printf $3 }''
  adcvalue=$(if [ "$adcvalue1" -gt "1" ] 2>/dev/null; then echo $adcvalue1; else echo 1; fi)
  rntc='echo "10000/((1024/$adcvalue)-1)"|bc -l'
  temp='echo "$beta/l($rntc/$r00)-273" | bc -l'
  printf "%s %.2f\n" 'date +%Y-%m-%d_%H:%M' $temp >> $txtfile
  if [ "'echo $loop%24 |bc'" -eq "0" ]; then
    xlines='cat $txtfile |wc -l'
    nofxtic='echo $xlines*900/10|bc'
    printf "set terminal gif size 800,480\nset output \'$giffile\'\nset timefmt \"%%Y-%%m-%%d_%%H:%%M:%%S\"\
     nset bmargin 4\nset xdata time\nset xtics rotate 270\nset xtics $nofxtic\nplot \'$txtfile' using 1:2
     with lines\n" |gnuplot
    echo Temperature |mutt -s "Temperature" xxx@gmail.com -a $giffile
  fi
  loop='echo "$loop+1"|bc'
  sleep 900
done
```

or periodically displaying the ADC and the time. The data logger is also controlled from the main loop.

More detailed instructions for modifying the firmware, such as how to add default settings and what to watch out for, are shown in the source code itself.

Other systems

This article has focused mainly on controlling the Android I/O board via an Android device. But the board also works perfectly well with a PC or laptop under Windows or Linux, or with devices such as the Raspberry Pi.

The author used an Android I/O board with WiFi to keep a record of the temperature during the holidays, which was then transmitted by a Raspberry Pi via email. It is based on a simple bash script that uses the telnet command to make contact with the Android I/O board every quarter of an hour, converts the result into degrees Celsius and adds that value to a file. The data is then converted four times per day by GnuPlot into a graphical GIF format, and then emailed (**Figure 9**). The script is shown in **Listing 1**; It assumes that the programs bc, gnuplot and mutt have already been installed.

It is also possible to connect the Android I/O board (via jumper block JP2) directly to the serial port of the Raspberry Pi, which means that no communications module is required. The Android I/O board then functions as an I/O extender for the Raspberry Pi. Pin 6 (Gnd), 8 (TXD) and 10 (RXD) on the Raspberry Pi should be connected to the first three pins on the right-hand side of JP2 (TXD connects to R

Bugs in the PIC16F1938

The first two versions of the PIC16F1938 (revisions A and B) have a bug, which prevents the ADC being read reliably at clock frequencies over 16 MHz (the controller can even lock up), see [4]. The version number can be read via the Z3 status register, command 11.

There is a workaround for reading the ADC in these versions reliably. Part of the workaround includes turning off the interrupt during the reading and temporarily lowering the clock frequency.

This can introduce some jitter in the PWM signal and the counter. Because of this, two versions of the firmware have been made available: a ‘normal’ one for the C revision (and later ones in the future) of the PIC16F1938 and a version including the workaround (with ‘_AB’ in its name).

and RXD to T). You can also connect the 3.3 V of the Raspberry Pi to the fourth pin, so that the Android I/O board is powered by the RPi. In that case you must ensure that the current consumption is less than the maximum that the Raspberry Pi can supply.
In the software the serial port can be written to and read from via */dev/ttyAMA0*. The baud rate can be configured using *stty –F /dev/ttyAMA0 speed 9600*. Since the serial port is used by default to send logging information, this should be turned off. This can be done by changing */etc/inittab* and */boot/cmdline.txt*. In the first file you have to add a # (has sign) indicating the line is comment:

```
#T0:23:respawn:/sbin/getty -L
ttyAMA0 115200 vt100
```

In the second file you have to remove

```
console=ttyAMA0,115200
kgdboc=ttyAMA0,115200
```

(150310)

Web Links

[1] http://androidplot.com/download/

[2] http://androidplot.com/

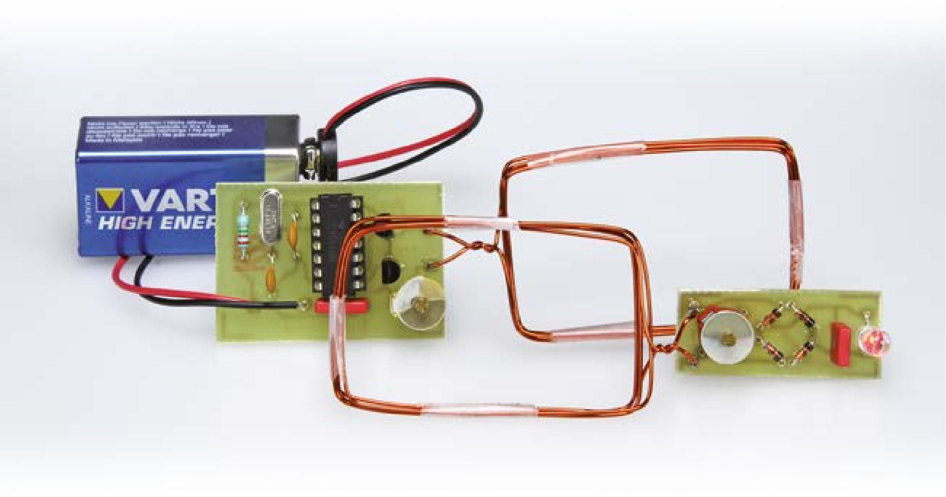
[3] www.elektormagazine.com/150057

[4] http://ww1.microchip.com/downloads/en/DeviceDoc/80000501F.pdf

Wireless Power Transfer
Experiments using few parts

By **Wouter Eisema** (Netherlands)

Wireless energy transfer is the future! With this small experimenting set you can explore this technology yourself! Very little is required and over a distance of about 5 to 10 inches you can transfer a (small) amount of energy.



The wireless charging of all kinds of equipment will probably become very common in the next few years. This technology usually involves two coupled coils, in which energy is transferred from one coil to the other in an electromagnetic field. A nice example of this can be found in the design of the propeller clock, which is described in this issue and the previous one. In that particular example this technology is used to supply power to the rotating part.

Simple schematics

The transmitter circuit (**Figure 1**) consists of an oscillator built around four of the inverters inside a 4069 CMOS-IC. The oscillator frequency is set at RFID frequency 13.56 MHz (X1). This frequency is also available at a pin on the circuit board and can be used for other experiments.

The crystal oscillator itself comprises IC1.A and X1, together with resistor R1 and the two load capacitors C1 and C2. The signal is buffered by IC1.B and subsequently goes to the push-pull output stage consisting of IC1.C and IC1.D, which in turn drive small-signal MOSFETs T1 and T2. Connected to the output is an LC circuit made from a coil with 5 turns that you wind yourself and a trimmer capacitor. This series-resonant circuit will be tuned later to the crystal frequency, so that the coil will generate the maximum possible output. To prevent interfering with other equipment, the output power of this transmitter is limited and the frequency that is used is the same frequency at which many RFID readers operate (this frequency is in one of the so-called ISM bands).

The design of the receiver (**Figure 2**) is even simpler, it is entirely passive. It consists of an LC network with the same physical dimensions as those of the transmitter, supplemented with a bridge rectifier consisting of D1 through D4 and a small filter capacitor C2. The load consists of a (red) high-efficiency LED. Here too the LC circuit is tuned with the trimmer capacitor to 13.56 MHz after it has been built.

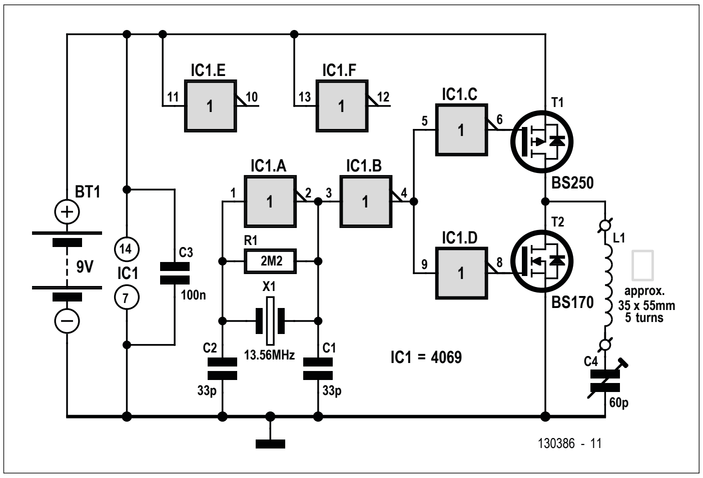


Figure 1. Schematic for the transmitter. This consists of a crystal oscillator and a push-pull driver stage for the LC-circuit.

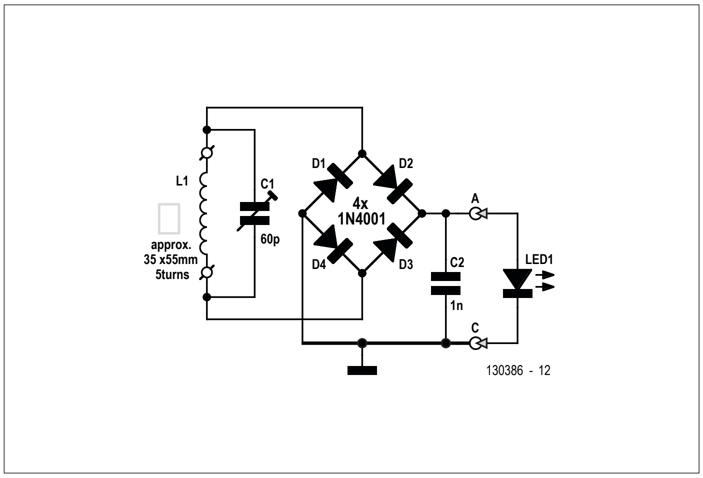


Figure 2. The receiver is entirely passive. The signal received by the LC-circuit is rectified and sent to the LED.

Component List

Transmitter

Resistors

R1 = 2.2MΩ

Capacitors

C1,C2 = 33pF
C3 = 100nF
C4 = 60pF trimmer

Inductor

L1 = 5 turns enameled copper wire, size approx. 35x55 mm

Semiconductors

T1 = BS250
T2 = BS170
IC1 = 4069

Miscellaneous

X1 = 13.56MHz quartz crystal
9-V battery with clip-on lead

Receiver

Capacitors

C1 = 60pF trimmer
C2 = 1 n

Inductor

L1 = 5 turns enameled copper wire, size approx. 35x55 mm

Semiconductors

D1-D4 = 1N4001
LED1 = LED, red, high efficiency

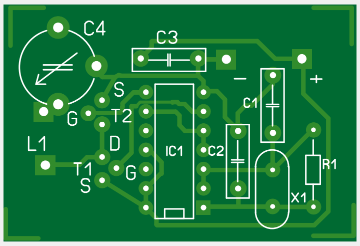


Figure 3. The circuit board of the transmitter...

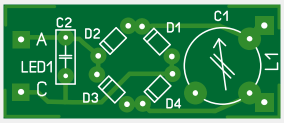


Figure 4. ...and the even simpler circuit board for the receiver.

Construction

The design of both circuits is so simple that they can be built easily an a couple of small pieces of prototyping board. But to make life even easier for you, we have designed two circuit boards, the layout of which (in the Sprint Layout 6 format) is available from [1].

A few remarks regarding the construction. On the transmitter board all the components are soldered straight on the circuit board, the 4069 can optionally be fitted into a socket. Take care when mounting the two MOSFETs. The flat side of the package faces the component designator (T1 and T2) printed on the board. Both of these FETs are offered in 2 different versions that have different pinouts. Should you have any of these in your bins among your old components then check the datasheets from the manufacturer to make sure what the pinout is for the actual parts that you have.
A 9-V battery will suffice as the power supply for the transmitter. This will last for quite some time, particularly when you are experimenting only every once in a while.
The coil consists of 5 turns of enameled copper wire (magnet wire) with dimensions of about 3.5 x 5.5 cm (wind it around a match box or something similar). While you are at it, you can make two of those, since you need one for the transmitter and another one for the receiver.

The construction of the receiver is even easier. As usual note the polarity of the diodes and the LED (the shorter lead is the cathode).

Now both circuit boards need to be tuned. First make sure that all parts are mounted in the correct position and are soldered properly. Connect the 9-V battery. Place the transmitter and receiver about 4 inches apart. The LED will probably already light up a little. If this is not the case, it is possible that the position of the trimmer capacitor on both the transmitter and receiver is just a little wrong. Turn one of the trimmer capacitors a little.

Using a non-metallic trimmer spanner or screwdriver (it may also be a small wooden stick with the end cut into the shape of a screwdriver blade), first slowly turn the trimmer on the transmitter until you find the point where the LED is the brightest. After that do the same on the receiver, the LED will become even brighter at the optimum position of the trimmer.
If necessary, repeat the adjustment with a different distance between the coils.

You could carry out this tuning with a metal screwdriver, but the metal will influence the oscillator circuit. When the screwdriver is removed the frequency will change a little.

Ready to experiment

You can now experiment with the combination of these two circuits by positioning the transmitter and receiver closer together or further apart and observe how the LED brightness changes when the coils are held in a different orientation with respect to each other. It is also possible to experiment with other coil shapes and sizes (but do not forget to tune the trimmers again).
Using the test model, the LED at optimum tuning was still clearly visible at a coil distance of about 10 inches.
Enjoy your experimenting! ◀

(130386)

Internet Link

[1] www.elektormagazine.com/130386

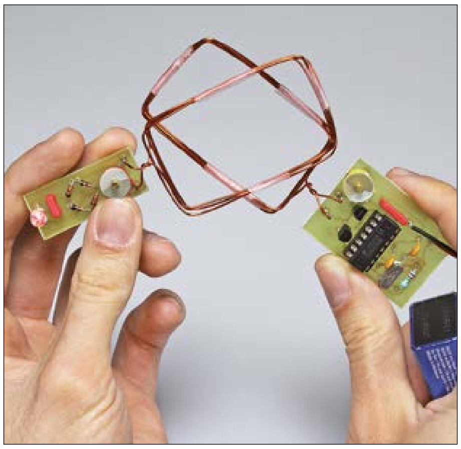


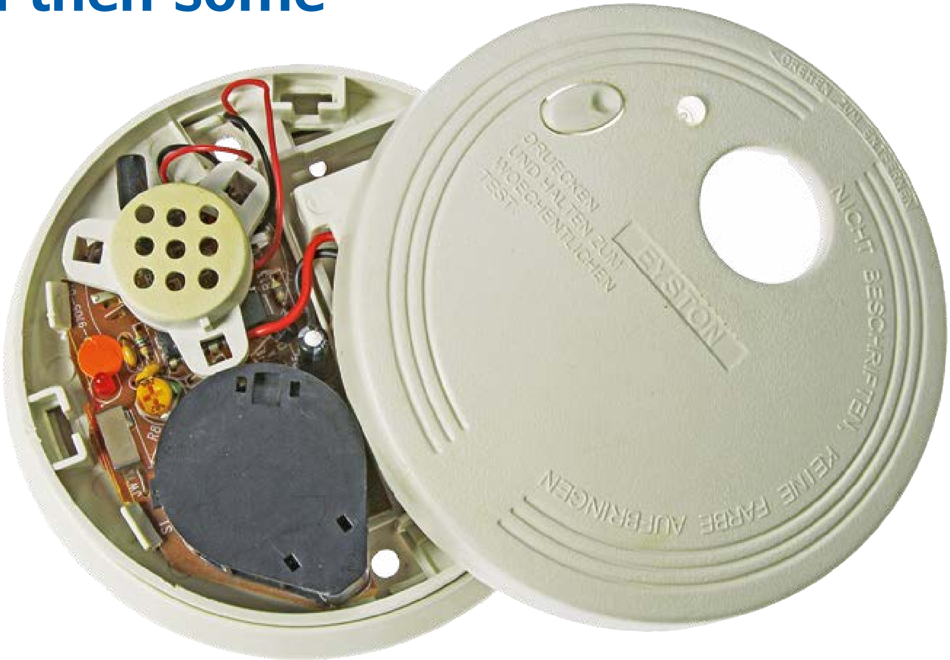
Figure 5. The energy transfer between the coils is strongly dependent on the relative position of the coils.

Universal Alarm

Smoke alarm and then some

By **Jürgen Friker** (Germany)

Just add a few external components to a standard low-cost smoke alarm and turn it into a universal alarm sounder. You can easily hook up the outputs from any number of sensors or detectors to trigger the alarm and the smoke detector still functions as normal.


DRUECKEN UND HALTEN ZUM WOECHENTLICHEN TEST
EI SYSTON
NICHT BESCHRIFTEN, KEINE FARBE AUFBRINGEN

Down in our cellar we have a sump pump to automatically pump out any ground water that collects there. Should the pump fail and the water level start to rise, I need to know, before the cellar floods. The best solution is to fit a water level detector in the sump recess with its output connected to an alarm buzzer or sounder that can be heard throughout the house.

Waking the dead

I tried out a whole range of different alarm sounders to find the best solution. Mains powered horns are certainly loud enough but if the power goes down, so will the horn along with the pump. It is also easy to unintentionally blank-out this type of continuous alarm sound unless you are standing in the same room. For some inexplicable reason my brother, who likes to play tuba, wasn't keen to take on the job of unpaid 'alarm sounder' on a permanent basis… The best solution I came up with is a plain vanilla smoke detector. The alarm sound is really penetrating and it works from battery power. Another nice feature is its built-in battery voltage level alarm. The detector is also quite cheap; the mini DIN plug and socket I bought from the electronics store for hooking up the sensor actually cost about the same as the complete smoke detector.

Every smoke detector I looked at uses a Freescale (previously Motorola) chip. Sometimes the chip type differs but they all have a 'networked' I/O pin 7 (**Figure 1**). This allows a number of smoke alarms to be linked together using a 2-wire bus so that when one alarm triggers, it sets off all the other alarms. As a stand alone smoke detector this network port is not used, but can of course be used as a general purpose alarm input. You won't need to make any changes to the smoke detector circuit (or the test button) and the smoke detector will still work as normal. The I/O pin7 has an internal pull-down resistor to ground. When the alarm is triggered this pin is pulled high and raises the bus to the positive supply voltage level. Input signals in the range of –0.25 V to VDD+10 V (this equates to +19 V for a 9 V PP3 type battery) are tolerated on this I/O. The switching threshold is +3.2 V. The MC145012 (and similar Chips) employ an internal current sink to give good immunity from interference. The LED D4 allows you to determine which smoke detector caused all the alarms to sound in a networked setup. Its LED will flash at half-second intervals while LEDs in all the other detectors remain off.

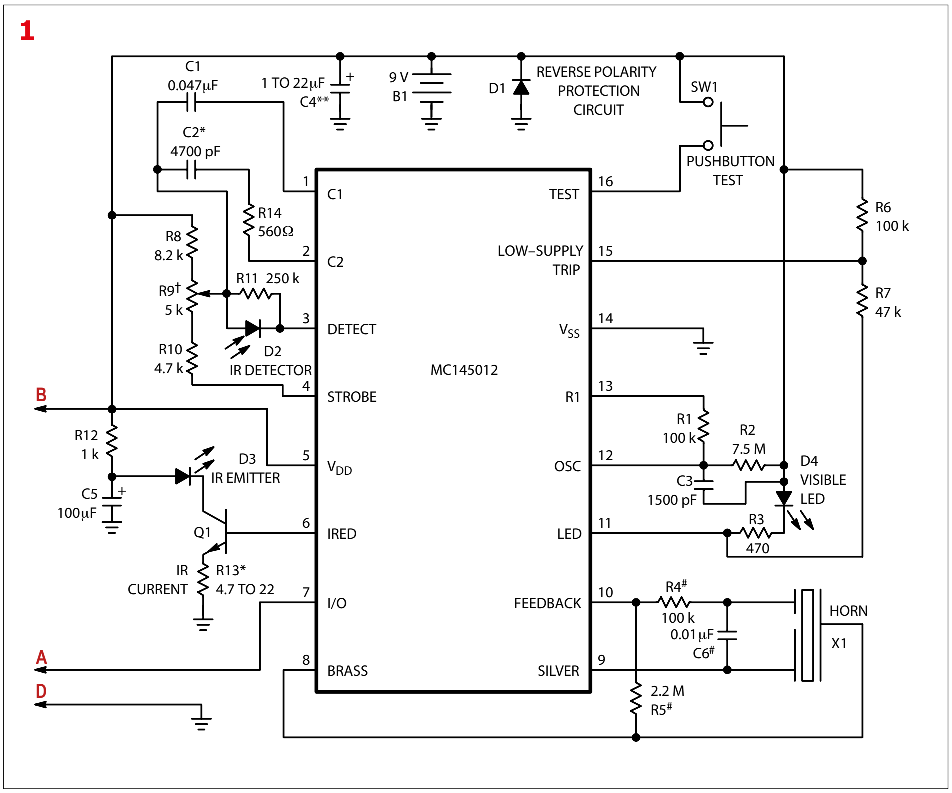
Incidentally it's not a good idea to connect an external alarm signal to the test input pin 16. This input is not designed for the connection of long wires. When test is pressed it sets the smoke sensing circuit amplification to a maximum so that stray light in the smoke chamber is sufficient to trigger the alarm. This could also occur if interference or spikes are picked up by a long wire connected to this input pin. Apart from this, the pin only tests the smoke detection function, if the test is negative then the alarm won't sound.

Warm beer alert

The smoke detector can be used as a general purpose alarm; it doesn't matter if the high alarm signal is generated by a fridge door sensor, a beer cooler over-temperature detector or by an intruder alarm. You can connect the high-going output from all sorts of sensors (within the defined limits) to the 'A' input of the circuit shown in **Figure 2**. You can also combine the outputs from several detectors as a wired-OR input. As long as the voltage level on the 'A' input remains below 1.5 V then the alarm is not triggered. For the water level detector I used a float switch to drive this input high but there is no reason why you couldn't use a transistor or more complex circuit to do the same job.

The float switch is actually a tilt switch, these days they don't contain mercury but more environmentally acceptable steel balls rolling together to achieve the electrical contact. The switch should not trigger the alarm when any water is present in the sump. It needs to switch when the water rises above the danger level (above the pump but before it flows into the cellar). For this switching to occur reliably you need a mechanism or arrangement that allows free movement of the float. Here you can experiment to your hearts content, you can use a tube and cable ties or make use of a counterweight. With this set up its important that the counterweight is heavy enough to overcome the float buoyancy and make sure the weight or cable cannot slip. The free end between the weight and float should not be too short otherwise cable stiffness may hinder movement.

The smoke detector circuit shown in Figure 1 is taken from a Motorola application and was used in the smoke detector shown (with R13 = 6.8 ohms).


1

A risky supply

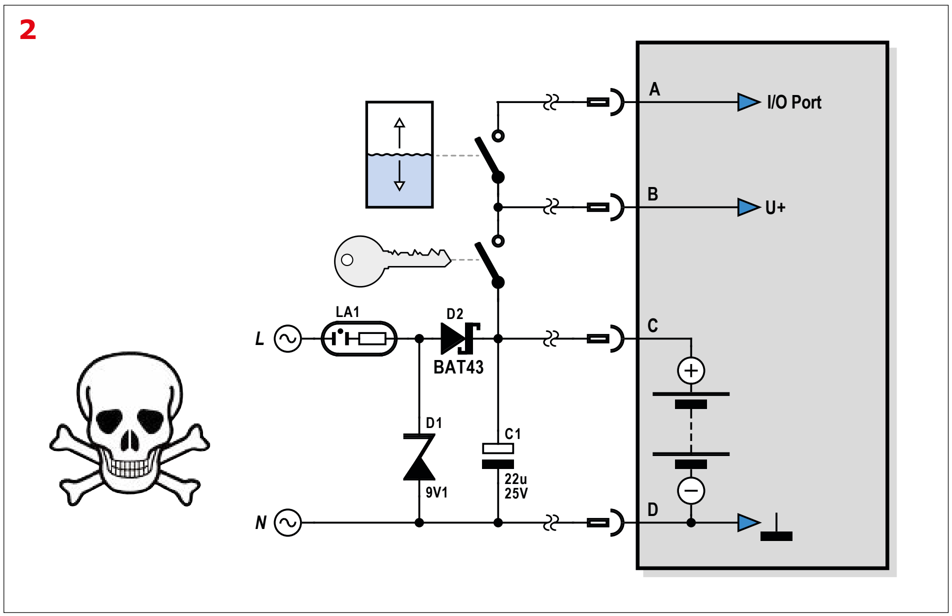
The circuit in Figure 2 shows a simple mains-powered charger to keep a rechargeable NiMH battery fully charged. The neon lamp LA1 is an E10 indicator lamp with a built-in series resistor giving a current flow of 1.5 to 1.9 mA at 230 V. The low level of charge current is just sufficient to compensate for self-discharge in the battery and can never overcharge it. The lamp indicates mains power is available (the alarm doesn't need it but the pump does). The Zener diode D1 produces a low DC voltage and conducts in both directions so that the lamp is not dimmed. D2 prevents the battery discharging through D1. C1 is used as a reservoir capacitor and is effectively in parallel with capacitor C4 on the detector PCB. Diode D1 on the detector board protects against reversed battery connection and doesn't otherwise interfere with circuit operation.

The value of C1 should not be made any larger otherwise it disrupts the battery check function of the alarm: Every 30 s the chip pulls down the LED output on pin 11 for around 8 ms and measures the battery voltage level at the input from the divider network R6 and R7 (a test under load). A larger capacitor with more stored charge will give a false confidence level of the remaining battery capacity; the detector will now only discover too late there is insufficient energy in the battery to sound the horn.

The charging circuit is really simple but also **potentially lethal!** There is no galvanic isolation from the AC line and it **must not be used** if you are able to accidentally touch any part of the circuit. Any external components like the key switch, sensor and wiring **must be fully insulated**. The charger design rules out the use of a sensor with bare wires to detect the rise of flood water.

The switch is used to turn off the water alarm, ideally use a key switch here so that it can't be turned off by any unauthorized person. In one experiment I replaced the battery with a very large capacitor which was OK in normal operation but quickly ran out of charge when the alarm triggered. With the alarm installed I sleep more soundly, I know that if it sounds the cellar is on fire or is about to flood, either way, I will take a bucket. ⏮

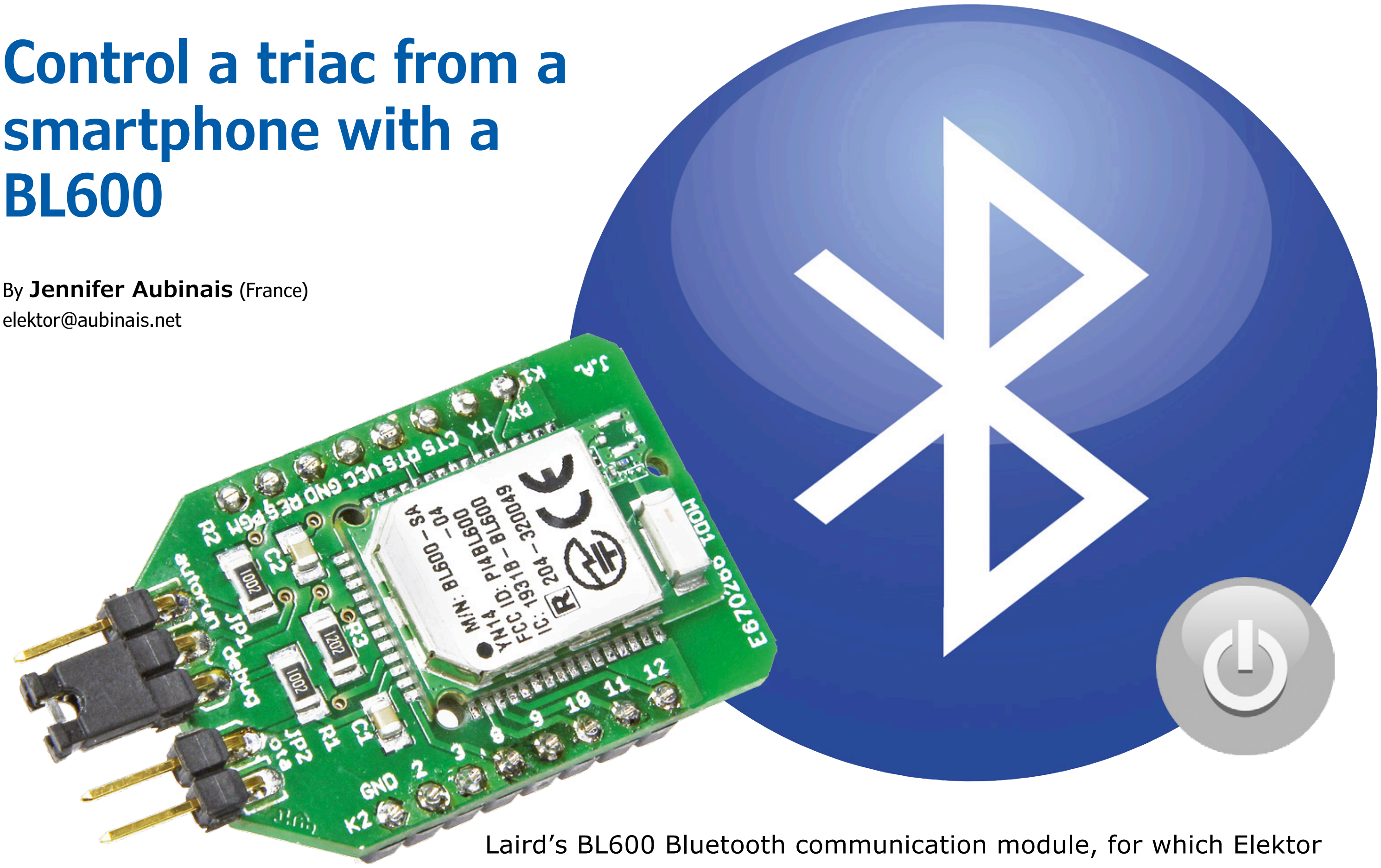
(140184)


2

Line AC Switch Controlled by Bluetooth Low Energy

Control a triac from a smartphone with a BL600

By **Jennifer Aubinais** (France)
elektor@aubinais.net



Laird’s BL600 Bluetooth communication module, for which Elektor offers the famous e-BoB breakout board, is used here in a complete new and autonomous project to switch mains voltage, for example switch a light on or off from an Android smartphone.

This article describes the electronics, the BL600 program and the Android program; the last being an Android smartphone application with which your mains switch circuit may be controlled wirelessly. You may notice an unusual detail (for these pages) — the transformerless power supply — and that is where we will begin (**Figure 1**).

The electronics

We can separate the schematic into three parts:

- The power supply: C1, R1, R2, R7, R8, D1 to D5, C2, IC1;
- The *Bluetooth Low Energy* module: MOD1, R4, LED1;
- The switching: T1, IC2, R5, R6, TRI1, R3, LED.

The power supply is notable in that to save space, it does away with a voltage transformer and thus poses a **risk of electrocution**, because there is no galvanic isolation between this circuit and the AC line (230 V or 115 V). This demands extra vigilance and care, and strict respect for the rules for construction and use of this type of circuit.

The BL600 module (MOD1) is powered at low DC voltage by a 3.3-V regulator (IC1), the input voltage of which is limited by a 6.8-V zener diode (D2). To correctly dimension the power components we must estimate the total consumption of the Bluetooth switch components:

- BL600 module: 6 mA while transmitting;
- 2 LEDs: 3.3 [V] – 1.6 [V]) / 1000 [Ω] = 2 [mA] so 4 mA for the two LEDs (estimate);
- MOC3041 optocoupler (3.3 [V] – 1.3 [V]) / 220 [Ω] = 9 [mA];
- LP2950 regulator current max.: 12 mA;

Giving a total of some 31 mA. This modest consumption allows us to dispense with a power transformer and instead use a capacitor (C1). It is this capacitor which is the main part of the power circuit that reduces the voltage from the AC line (230 VAC or 115 VAC) applied to K1. In effect, the capacitive reactance of a capacitor, designated X_c and expressed in Ω, is, under ideal conditions, a **resistance to alternating current**, inversely proportional to the frequency of the AC

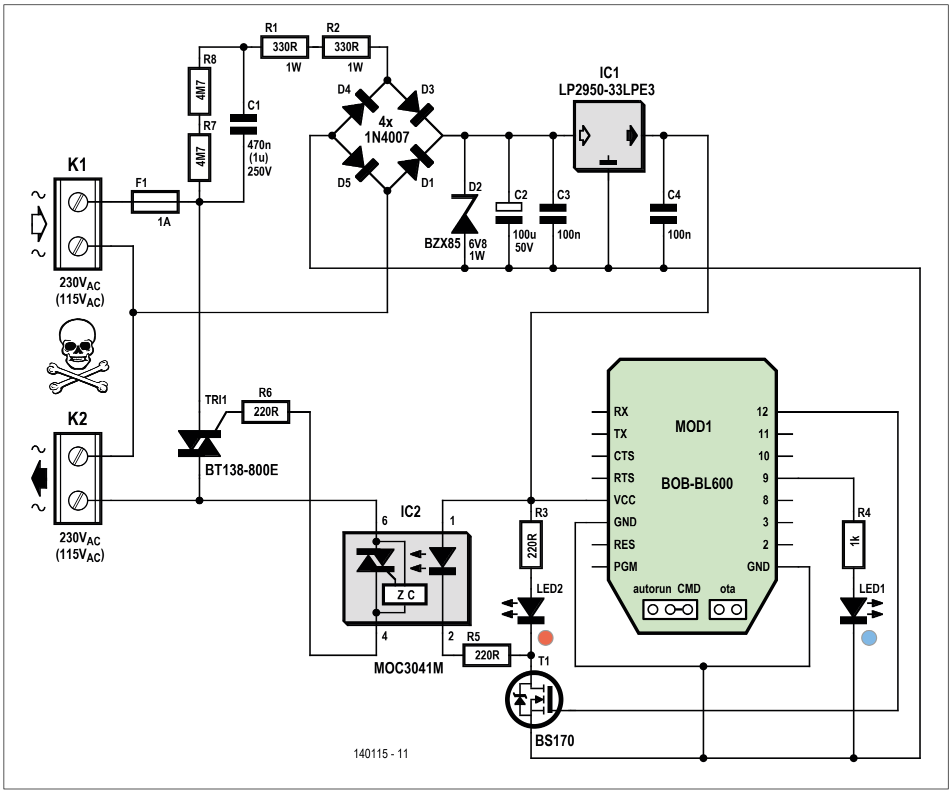


Figure 1. The absence of a transformer is (partially) compensated for in the schematic by a skull and crossbones. Don’t let your fingers or other body parts get anywhere near this circuit!

voltage. Here this frequency is that of the power line, 50 Hz in Europe, 60 Hz in the USA. From this we can calculate the capacitance required as follows:

$X_c = 1 / (2 \pi f C)$

With
f: frequency in Hz
C: capacitance in farads
$C = 0.031 / (2 \times 3.14 \times 50 \times (230 - 6.8)) = 0.44\ \mu F$
We can use **470 nF** for 230 VAC line voltage, or **1 µF** for 115 VAC line voltage.

Calculation of resistors

On the application of power, C1 is discharged and effectively is briefly seen as a short circuit; to reduce the peak current that may result, we insert two resistors in series (because of the possible maximum of 325 V which most resistors are not rated for). The total resistance of 650 Ω (2 x 330 Ω) limits the current to around 500 mA. As soon as C1 is charged, the current through the resistors falls to 30 mA, and the power to 300 mW, but taking account of the possible much stronger peak power, we have rated these resistors at 1 W.
The purpose of 4.7-MΩ resistors R7 and R8 of is to ensure that the 470-nF capacitor is discharged once AC line power is removed. Be cautious around these components, even when disconnected from the grid.

Rectification and regulation

Once the AC line voltage is reduced by C1, a bridge rectifier of four 1N4007 diodes (to use both half-cycles) which feeds an LP2950 to obtain the low voltage DC required for the BL600 module. The entry voltage for the regulator, limited to 6.8 V by D2, is smoothed by the 100-µF electrolytic capacitor C2 and filtered at input and output by 100-nF capacitors.

Optocoupler

Because of the use of the bridge rectifier, the voltage of the ground of the BL600 Bluetooth module floats with respect to the AC line wires. We therefore use an optocoupler (IC2) which maintains galvanic isolation between the output pin of the low voltage circuit (pin 12 of MOD1) and the triac circuit at 230 V.
But make no mistake, this ‘isolation’ thus obtained between the 230 V AC line voltage and the Bluetooth module does not change anything in respect of guarding against the risk of electrocution.

Output GPIO12

During tests at the Elektor Labs, it was found that the current requirements of the LED of the optocoupler and LED2 were too high for the output of the BL600 module to drive directly, so we have added a FET (T1).

LED

The blue LED shows the status of the Bluetooth connection on pin GPIO9: it flashes when the module is not connected and is on when it is. The red LED on pin GPIO12 shows the state of the optocoupler and thus the status of the load connected to the terminals (K2) controlled by the triac.

The triac

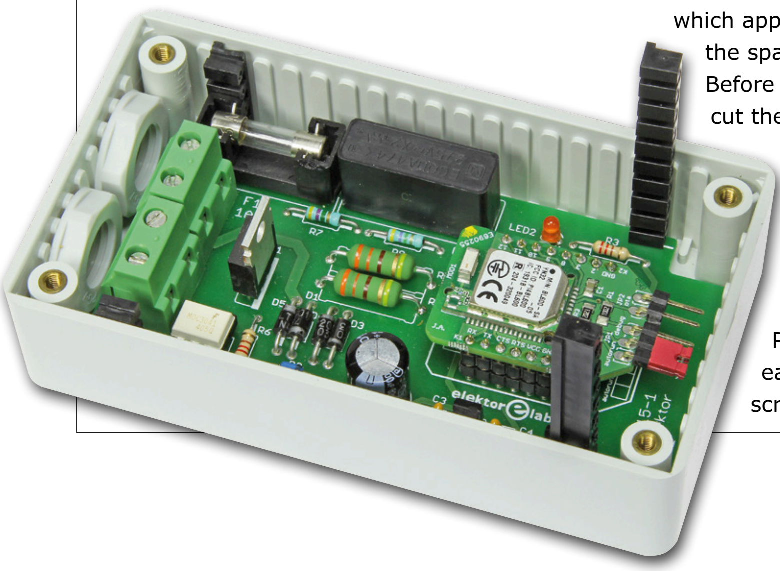
According to the NXP datasheet, the maximum average RMS current in the triac is 12 A. However the heating effects come into play here: a TO-220 package without a heatsink can dissipate around 1.5 watts at ambient temperature, and it is this which limits the current. The maximum allowable current is then estimated at 0.6 A, let’s say 0.5 A to give us a safe margin. The maximum power of the load is thus around 115 W — again let’s say 100 W to be safe.

Construction and test

Having programmed our eBOB-BL600 (see the section on the application for the module), construction and implementation should not pose particular problems, but we must not lose sight of the fact that this circuit, having no transformer, is connected direct to the AC line voltage. For our initial tests, instead of powering the switch from the grid, we’ll use a low voltage DC lab power supply connected to the solder pads of D2, before this diode is fitted to the board. Thus may we test the correct functioning of the BL600 without connecting the circuit to any dangerous voltages. In the absence of a lab power supply, you can use any mains adapter giving a DC output of 7 to 9 volts. We can then simply check for the establishment of a Bluetooth connection between the BL600 and the tablet or smartphone. Fortunately, with the circuit supplied from low voltage DC, it is also possible to program the BL600 in OTA mode!
LED2 shows if the switch is on or off. Establish communication with the Android application (see below) or use the appli-

Mounting in the case

Whatever type of case you use, don't forget to use cable glands to protect the wires from strains and pulling. There was not a lot of space in the model we used, but we got there. The nice feature of this particular case is that it can take four grooved and trimmable PCB guides which allow the PCB to be secured in place without any screws! On one side of these strips are grooves in which the PCB can sit, the other side has a U-shaped profile that allows them to slide up and down the grooves in the sides of the case. A very clever system which appreciably reduces the space requirements. Before closing the case, cut the strips to the right length — just up to the cover of the case. The other advantage is that you can fit and remove the PCB quickly and easily without using a screwdriver.



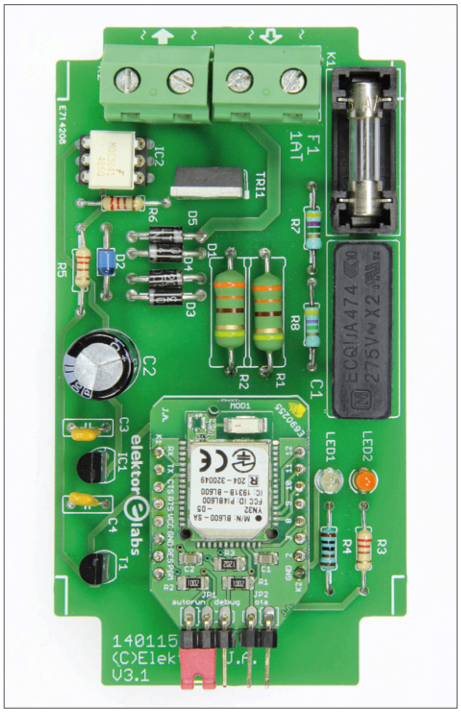


Figure 3. The prototype assembled and ready for installation in the case.

cation *Serial* from Laird: using the Scan function, start by establishing a connection with the BL600, and select "JA_SWITCH". Choose the option *Connect*. LED2 should react when you touch the button in the application, or when *Serial* sends a 1 (on) or a 0 (off).

Everything working OK? Install D2 on the PCB and then mount it, **well isolated**, in the case before you connect the mains voltage.

The application for the module

Both while testing the board and when programming the BL600, it should never

Component List

Resistors
R1,R2 = 330Ω 1W
R3,R5,R6 = 220Ω 0.25W
R4 = 1kΩ 0.25W 250V
R7,R8 = 4.7MΩ carbon film 0.25W 250V

Capacitors
C1 = 470nF 275 VAC, 22.5mm pitch (for 230 VAC)
C1 = 1µF 275 VAC 22.5 mm pitch (for 115 VAC)
C2 = 100µF 50 V radial
C3,C4 = 100nF 50V

Semiconductors:
LED1 = LED, red, 3mm
LED2 = LED, blue, 3mm
D1,D3,D4,D5 = 1N4007
D2 = BZX85C6V8 (6.8V 1W zener diode)
T1 = BS170
Tri1 = BT138-V
IC1 = LP2950-33
IC2 = MOC3041M

Miscellaneous
F1 = fuse, 1A slow-blow 20mm
Fuseholder, 5x20 mm (1162740)
K1,K2 = 2-way PCB screw terminal 0.3'' (7.68mm), 630V rated (1793006)
MOD1 = BL600 e-BoB, ready assembled, ElektorSTORE # 140270-91
Case = Hammond 1591BTCL (transparent) (1877127) or 1591BGY (grey), 112x62x31 mm (4437019)
PCB adapters for case (1876930)
PG7 cable gland (Lapp Cable) (1178859)

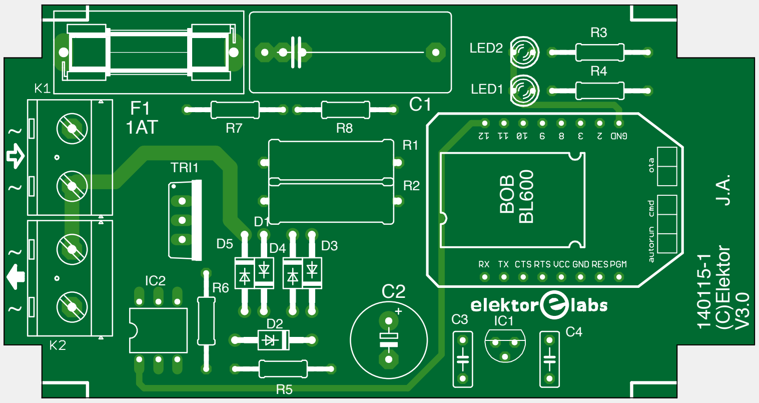


Figure 2. The printed circuit board of the BT-LE controlled AC Line switch.

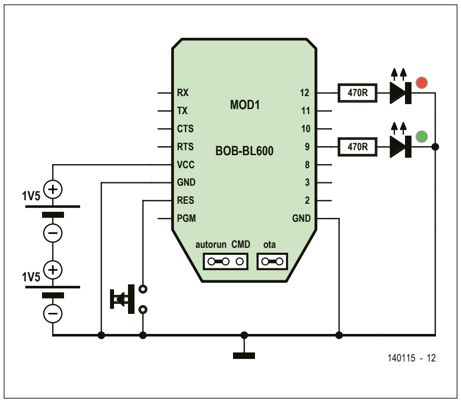


Figure 4. Test Circuit for programming of the BL600 — mount it on a breadboard away from AC line electricity.

be handled while it is connected to mains voltage. For programming, we recommend to mount the e-BoB BL600 on a breadboard (**Figure 4**) powered by two batteries, for loading the program.
You'll need two 470-Ω resistors, two LEDs (red and green), a pushbutton, and a battery holder and two AAA batteries (if you don't have those in your junk box, use the ones from your TV remote.)

Download from the Elektor website [5] to your Android phone the file $autorun$.pgmSWITCH.uwc, as well as the application Laird Toolkit [6] from their website. On the e-BoB BL600, put JP2 in the OTA position and JP1 in the autoRUN position, then power up your test setup. Start the Toolkit application and choose the OTA option (**Figure 5**), then tap on *Select File* et find the file $autorun$.pgmSWITCH.uwc on your telephone (**Figure 6**). Start *Scan*, then choose your module LAIRD BL600. If it does not work, reset the module (RESET). Tap on *Upload*... During the transfer from your telephone to the e-BoB, the scroll bar will progress… at the end, **don't forget to tap *DISCONNECT*.** Your BL600 is programmed. The green LED will flash. While you have your switch mounted on a breadboard, take the opportunity, before connecting the AC line voltage, to test the BL600 with the LEDs using the application described below. Start the application, switch the red LED on and off as if you were switching the triac. If it works, it's ready to go.

The Android program

For the Android application (version 4.3 or up), our model was the description of the SERIAL program in the July & August 2015 edition of Elektor Magazine [7]. The source code is available on the Elektor Magazine website [8]. The program is already available for download from *Google Play* [9]. The principle is simple. After connecting to the Bluetooth switch (JA_SWITCH), the application sends the character 1 or 0 to switch

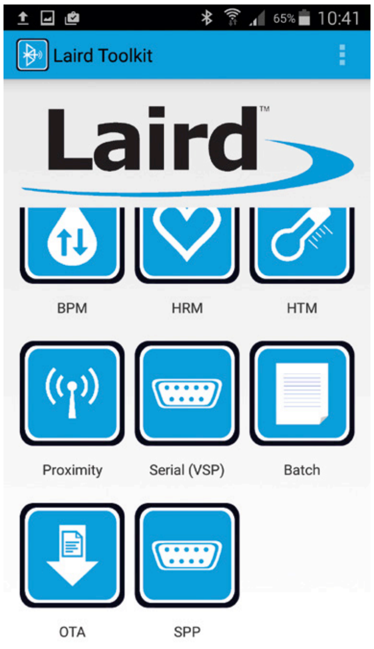


Figure 5. Copy of the Android application screen.

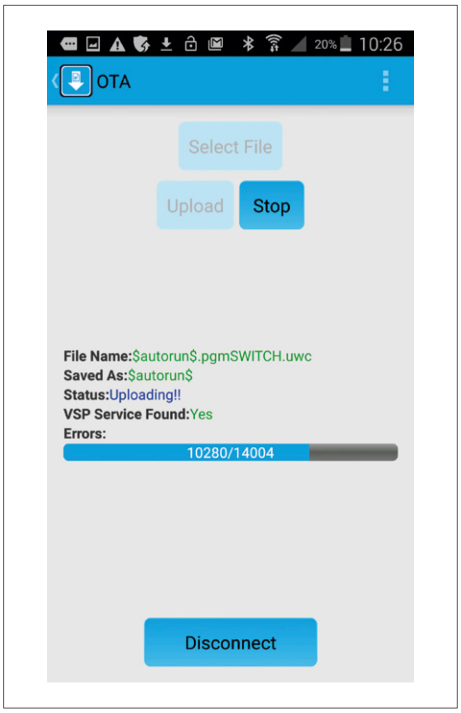
The roles of the Bluetooth Low Energy module

Our e-BoB BL600 carries out the following functions:

- Flashing LED1 (as an indicator of power as well as the absence of a connection to a smartphone)
- Connection via Bluetooth 4.0
- Reception via Bluetooth of commands sent from the telephone
  - 1: switch on the triac, to activate the connector K2
  - 0: Switch off the triac, to remove power from connector K2.
- Send to the telephone the status of connector K2:
  - *Your switch is OFF/ON* (useful to know the state of K2 when you establish a connection)
  - Light LED1 to show that the module is connected
  - Light LED2 to show the status of K2; it is the same pin which controls the triac

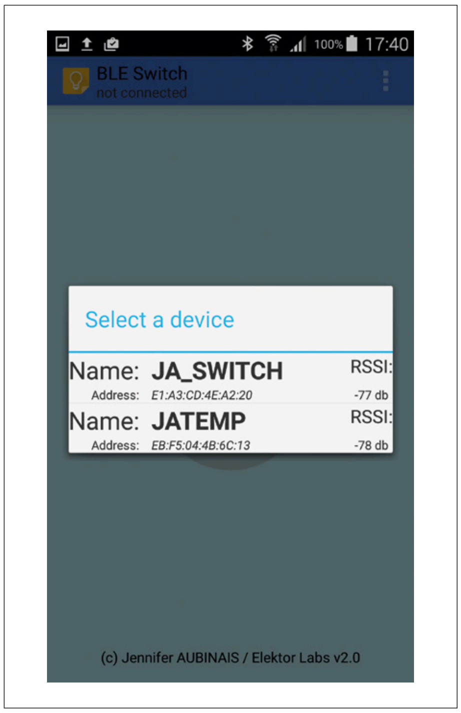
With the colors, it is easy to locate the code corresponding to these tasks in **listing 1**.

- Flashing LED1 (bleu): two timers start alternately: TIMER0 lights LED1, TIMER1 turns it off [3].
- Connection: we use the Handler HandlerBleMsg, renamed MyHandlerBleMsg, [4]. We will concern ourselves with two messages:
  - BLE_EVBLEMSG_CONNECT : (green)
    - After Bluetooth connection, stop the two timers (no matter which is active) by the function TIMERCANCEL
    - Switch on LED1 which was flashing
    - Read the state of the output which controls the triac, to send to the telephone text advising of its state
  - BLE_EVBLEMSG_DISCONNECT : (orange)
    - message sent on disconnection of the module
    - resume flashing LED1 by starting TIMER1
  - reception of a command from the téléphone: (red)
    - in the handler MyHandlerLoop
      - ◦ If the code 0x0D (carriage return) is seen in the characters received via Bluetooth (variable text$), we enter into the IF condition.
      - ◦ If the received character is "0"
        - GPIO12 output changes to 0
        - Send to the telephone the text "OK, Your switch is OFF"
      - ◦ If the received character is "1"
        - GPIO12 output changes to 1
        - Send to the telephone the text "OK, Your switch is ON"



File Name:$autorun$.pgmSWITCH.uwc
Saved As:$autorun$
Status:Uploading!!
VSP Service Found:Yes
Errors:
10280/14004

Figure 6. Copy of the "Upload" screen.



Select a device

Name: JA_SWITCH RSSI:
Address: E1:A3:CD:4E:A2:20 -77 db

Name: JATEMP RSSI:
Address: EB:F5:04:4B:6C:13 -78 db

(c) Jennifer AUBINAIS / Elektor Labs v2.0

Figure 7. Copy of the "JA_SWITCH" screen.

on or off. You can get the same result with the SERIAL application (TOOLKIT from *Laird Technologies* [6]): you connect and send the number 0 or 1 (don't forget to put in the CR for End of Line).

User Guide

This application, though simple to use, is nonetheless quite complex.

Before using it for the first time, the MAC address of your Bluetooth module is not in your phone's memory. We must first look for Bluetooth Low Energy devices with the UART service to find our device.

Start the application, tap on the menu, and tap on *Scan*. When the list of devices appears, select JA_SWITCH (Figure 7). The application then shows the status by the color of the button.

The module is connected, the switch is OFF. Tap above to switch ON. The switch is ON (Figure 11). Tap above to switch OFF.

To disconnect from your module, just slide your finger across the screen of your smartphone. You will then find one of the following states:

- The Bluetooth module is not connected and there is no MAC address for your switch in the phone's memory. You have to use the menu at the top right of the screen.
- The button at the center of the screen is flashing: the Bluetooth module can be connected directly by tapping on the button. This will work every time the application is started.
- Your Bluetooth switch is now ready to be used on your Android smartphone using the large colored button.

(140115)

Web Links

[1] https://laird-ews-support.desk.com/?b_id=1945

[2] BL600-eBoB Bluetooth Low Energy communication module (1), Elektor March & April 2015, www.elektormagazine.com/140270

[3] Elektor January & February 2016, www.elektormagazine.com/150329

[4] BL600 e-BoB (3): Elektor July & August 2015, www.elektormagazine.com/150129

[4] Elektor July & August 2015 www.elektormagazine.com/magazine/elektor-201507

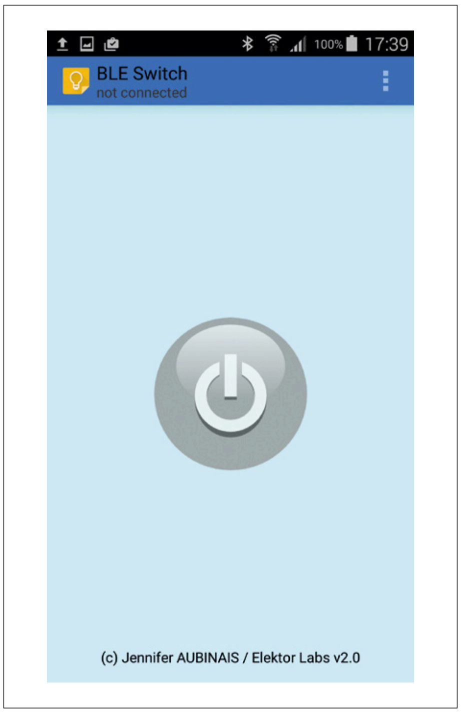
[5] $autorun$.pgmSWITCH.uwc, www.elektormagazine.comr/140115

[6] https://play.google.com/store/apps/details?id=com.lairdtech.lairdtoolkit&hl=en

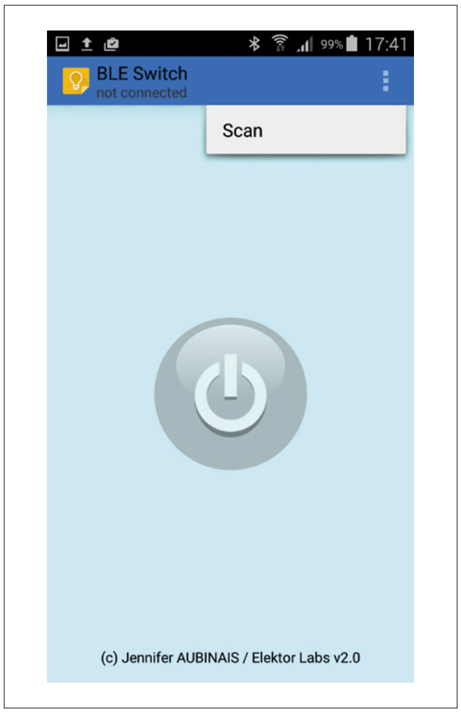
[7] BL600 e-BoB (5): Elektor November & December 2015, www.elektormagazine.com/150272

[8] Android Application — source code and APK installation file www.elektormagazine.com/140115

[9] Google PlayStore: https://play.google.com/store/apps/details?id=com.JA.bleswitch



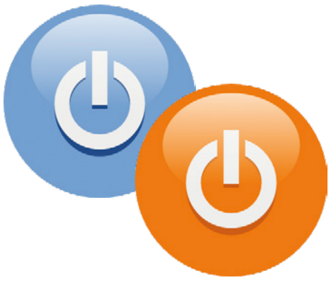
(c) Jennifer AUBINAIS / Elektor Labs v2.0



(c) Jennifer AUBINAIS / Elektor Labs v2.0

Listing 1

```
'//******************************************************************************
'// Laird Technologies (c) 2013
'// Jennifer AUBINAIS (c) 2015 version 1.1
'//******************************************************************************
'// Definitions
'//******************************************************************************
#define AUTO_STARTUP                          1
'//Set this to 0 to disable all debugging messages
#define ENABLE_DEBUG_PRINTS                   0
#define DEVICENAME                            "JA_SWITCH"
****** code here *****
//==============================================================================
// Led Connect flashes
//==============================================================================
FUNCTION FuncTimer0()
GpioWrite(9,1)
  TIMERSTART(1,100,0)
ENDFUNC 1
FUNCTION FuncTimer1()
  GpioWrite(9,0)
  TIMERSTART(0,1000,0)
ENDFUNC 1
//==============================================================================
// This handler is called when there is a BLE message
//==============================================================================
function MyHandlerLoop()
  DIM n, rc, tempo$, tx$
  DIM value, pos, return$
  tx$ = "0D"
  return$ = StrDehexize$(tx$)
  tempo$ = ""
  n = BleVSpRead(tempo$,20)
  IF (n > 0) THEN
    PRINT "$"
  ENDIF
  text$ = text$ + tempo$
  pos = STRPOS(text$,return$,0)
  IF ( pos >= 0 ) THEN
    DbgMsg("*")
    tx$ = "0"
    pos = STRPOS(text$,tx$,0)
    DbgMsg("OFF")
    Dim Txt$
    IF ( pos >= 0 ) THEN
      GpioWrite(12,0)
      MemSwitch = 0
      Txt$ = "OK, Your switch is OFF\n"
      rc = BleVspWrite(Txt$)
    ENDIF
    tx$ = "1"
    pos = STRPOS(text$,tx$,0)
    DbgMsg("ON")
    IF ( pos >= 0 ) THEN
      GpioWrite(12,1)
      MemSwitch = 1
      Txt$ = "OK, Your switch is ON\n"
      rc = BleVspWrite(Txt$)
    ENDIF
    text$ = ""
  ENDIF
endfunc 1
//==============================================================================
// This handler is called when there is a BLE message
//==============================================================================
function MyHandlerBleMsg(BYVAL nMsgId AS INTEGER, BYVAL nCtx AS INTEGER) as integer
  // Inform libraries
  //ConnMngrOnBleMsg(nMsgId,nCtx)
  AdvMngrOnBleMsg(nMsgId,nCtx)
  select nMsgId
    case BLE_EVBLEMSGID_CONNECT
      DbgMsgVal(" --- Connect : ",nCtx)
      TIMERCANCEL(0)
      TIMERCANCEL(1)
      GpioWrite(9,1)
      Dim Txt$
      if (MemSwitch == 0) then
        Txt$ = "Your switch is OFF\n"
      else
        Txt$ = "Your switch is ON\n"
      endif
      rc = BleVspWrite(Txt$)
      hConnLast = nCtx
      ShowConnParms(nCtx)
    case BLE_EVBLEMSGID_DISCONNECT
      DbgMsgVal(" --- Disconnect : ",nCtx)
      GpioWrite(9,0)
      TIMERSTART(1,10,0)
    ***** code here *****
'//******************************************************************************
'// Handler definitions
'//******************************************************************************
//all events have the same handler
OnEvent  EVVSPRX            call MyHandlerLoop //EVVSPRX is thrown when VSP is open and data has arrived
OnEvent  EVUARTRX           call MyHandlerLoop //EVUARTRX  = data has arrived at the UART interface
OnEvent  EVVSPTXEMPTY       call MyHandlerLoop
OnEvent  EVUARTTXEMPTY      call MyHandlerLoop
OnEvent  EVTMR0      call FuncTimer0
OnEvent  EVTMR1      call FuncTimer1
OnEvent  EVBLEMSG           call MyHandlerBleMsg // EVBLEMSG is called when there is a BLE message
OnEvent  EVBLE_ADV_TIMEOUT  call MyBlrAdvTimOut // TimeOut
'//******************************************************************************
'// Equivalent to main() in C
'//******************************************************************************
rc = GpioSetFunc(12,2,0)    // pin 12 : OFF / ON
rc = GpioSetFunc(9,2,0)    // pin 9 : Connected / NO connected
dim Adr$
Adr$=""
rc = bleadvertstart(0,Adr$,25,0,0)
TIMERSTART(0,10,0)
'//------------------------------------------------------------------------------
'// Wait for a synchronous event.
'//------------------------------------------------------------------------------
WaitEvent
```

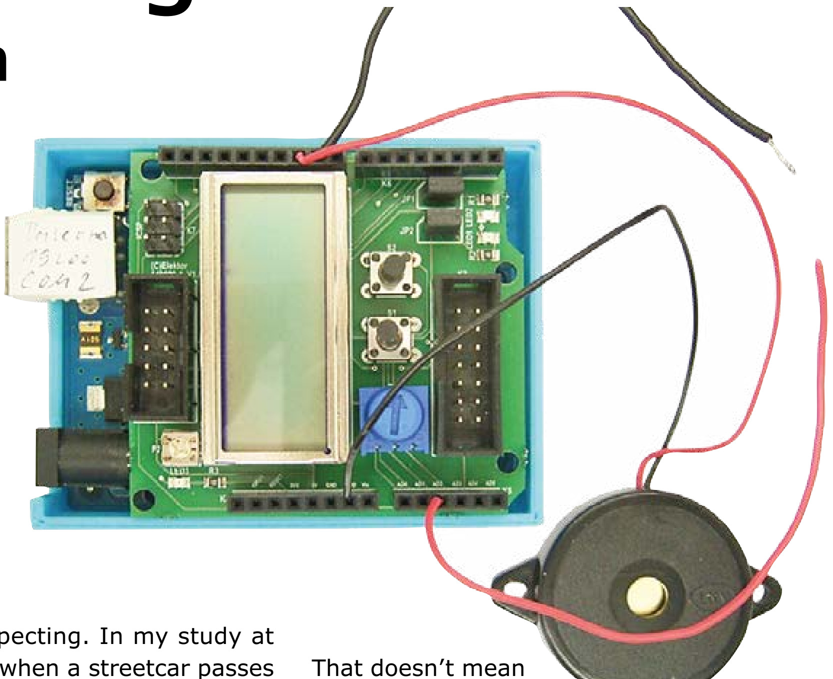


RF Detector using an Arduino

Programmed in Bascom

By Burkhard Kainka (Germany)

Add an RF receiver to a microcontroller and you open up many possibilities. It doesn't need to be complicated, for most applications you can get away with just using a simple diode detector; in fact we can even use a plain old LED for the job! Just take an Arduino Uno with an extension shield and you've already got all the hardware you'll need. Now add the code, programmed in Bascom.



If you delve back into the history of RF receiver designs you are sure to find reference to the simple detector receiver using a germanium (Ge) diode. Microcontrollers have inputs able to measure analog signals so there is no reason why we can't just hook up the output of the detector to one of the A/D inputs on the microcontroller and see if we can pick up some signals.

Once you've got it connected you have already built a signal strength meter. The readings you make can be useful, for example, in the world of amateur radio to tune an aerial. The resonant frequency of the detector circuit needs to be adjusted to be in the range of the measured frequency. You can also tune to a nearby medium wave station and check the received signal strength in your area. You might be surprised to detect things you weren't expecting. In my study at home I can tell when a streetcar passes by because it noticeably affects the received signal strength of an AM station around 720 kHz.

There are two main reasons for using a germanium diode in the detector circuit shown in (**Figure 1**). First off it has a low forward-conduction voltage. This means that signals as low as 100 mV will produce an output signal. Secondly its reverse-voltage resistance is not especially high, that's useful to dissipate any charge accumulating on the output capacitor.

If you replace it with a silicon diode such as a 1N4148, for example, you will need to receive a much stronger signal before you start to see an output signal from the circuit. You will also need to add some form of output load such as a 1 MΩ resistor.

That doesn't mean that silicon diodes are all bad for this sort of application. You can make use of a bias voltage to offset the diode's conduction threshold. **Figure 2** shows such a circuit without any form of tuned circuit for frequency selection so it's got a very wide bandwidth. With no received signal you can measure a voltage of around 0.6 V at the diode. When a signal is received this voltage level drops noticeably. This circuit works well with RF signals of around 100 mV. It can be used as an RF signal monitor and works across the entire short wave band without the need for any selector.

An LED as a detector diode?

Would it be possible to use the LED that's already fitted to the Elektor Extension-Shield? We already know that LEDs

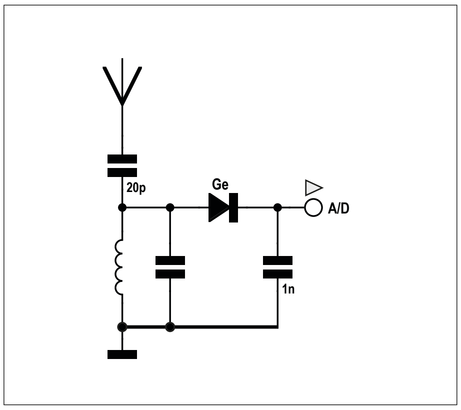


Figure 1. The classic detector receiver.

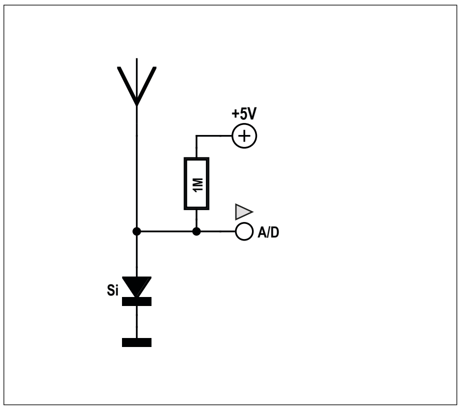


Figure 2. Si diode with bias voltage.

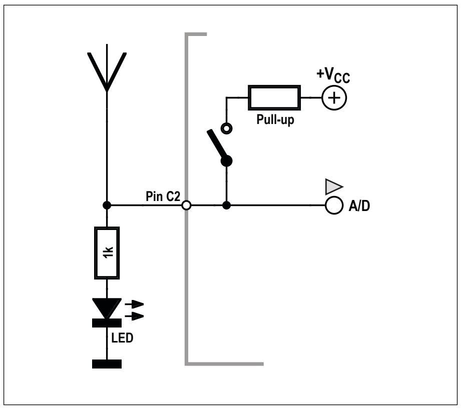


Figure 3. An LED RF detector.

Listing 1. Measuring the LED voltage [1].

```
‘---------------------------------------------------
‘UNO_RX1.BAS   B1 RF out, C2 RF in
‘---------------------------------------------------
$regfile = “m328pdef.dat”     ‘ ATmega328p
$crystal = 16000000           ‘ 16 MHz
$baud = 9600
$hwstack = 16
$swstack = 16
$framesize = 16

Dim D As Word

Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 ,
Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2
Cls
Cursor Off

Config Adc = Single , Prescaler = 64 , Reference = Avcc     ‘ 5V

Config Timer1 = Pwm , Prescale = 1 , Pwm = 10 , Compare A Pwm = Clear
Up
Tccr1a = &B10000010           ‘ Phase-correct PWM, Top=ICR1
Tccr1b = &B00010001           ‘ Prescaler=1

D = 8                         ‘ 1 MHz
Icr1 = D
Ocr1a = D / 2
Portc.2 = 1


Do
  D = Getadc(2)
  Print D
  Locate 1 , 1
  Lcd D
  Lcd “    “
  Waitms 500
Loop
```

also function as a photo diode, a voltage stabilizer, limiter and a varicap, surely we can get one to work as an RF detector as well. LED1 on the shield is already connected to the analog input ADC2. There is also a 1 kΩ resistor in series with the LED but that should not give a problem. The internal 30-kΩ pullup resistor can be configured to provide a bias voltage, perfect; we really don’t need anything else to build the circuit (see **Figure 3**).

And so to the software! Now we have already built our little RF test lab (**Listing 1**) and can begin programming it to generate an RF signal. A 1 MHz square wave signal is output from pin B1.

The program enables the pull up resistor on port pin C2 and continually measures the voltage on ADC2. Here you can read the value 410 which corresponds to a voltage at the LED of around 2 V. Connect a 10 cm length of insulated wire to C2 to act as an antenna. Attach a bare wire at B1 and hold the other end of it. Your body is now connected to the signal and becomes an antenna for the signal which can be picked up by a normal AM radio receiver. Now take the insulated wire on C2 and couple it to the RF signal. You will see the measured value drop to below 400. It’s interesting to note that the measured value remains at a constant level. The RF oscillations are not registered because the sample rate of the A/D converter is relatively low and this produces an averaging effect on the measured signal. The measurement shown is the average voltage across the LED which drops as the RF signal gets stronger.

An Integrating Detector

Maybe we could do with a bit more gain? This could be achieved in principle by using a higher value of pull up resistor. Even better would be to just switch on the pull up resistor briefly and then go into a high impedance state before the measurement is made. The charge across the LED will dissipate during the measurement period. The presence of an RF signal will increase the discharge rate.

You can think of the LED junction as having a small value of capacitance. Every peak of the received RF signal brings the LED briefly into conduction which has the effect of reducing the charge on its capacitor slightly. The value of this capacitor is only a few picofarads. That means you only need a very low level of RF current to produce a measurable effect. To give better sensitivity you can increase the delay between turning off the pull up and making a measurement. This will however make the circuit sensitive to low frequency signals which can cause interference. For this reason it’s better to make the measurement quickly after the pull up has been turned off using a relatively short sample time (`Prescaler = 8`).

The program in **Listing 2** performs averaging on the measurement samples to determine the zero level `D0`. By comparing the zero level with the input we can find out if an RF signal has been received. A drop of three A/D steps of the LED voltage is recognized as the threshold to indicate a signal has been received. Tests indicate that a received RF signal of around 50 mV is necessary. To flag this event, LED2 on the Elektor Shield is lit and a tone is produced at B2. You can hook up a simple piezo loudspeaker here to make the tone audible. The signal strength is also transferred serially but not available on the LCD due to timing constraints. When an RF signal is received at the input an (almost) constant tone will be audible at the output.

With this set up you can send and receive Morse characters. For this the output frequency was raised to 2 MHz to give increased range. For tests you can dab your finger on the output pin B1 to send Morse signals. The other hand should close enough to the receiving antenna to ensure reception of the signals. You can of course set up two Arduinos so that signals can be sent and received.

For test purposes a sine wave generator was used as a RF generator with a

finger touching the output signal so the body works as an antenna, it was possible to achieve a range of around 39 inches using an output signal of 16 V_{pp} and a frequency of 1 MHz.
So what can the circuit be used for? Without any additional circuitry you can use it to track down sources of RF interference. Many switch mode power supplies are guilty of high levels of RF noise. Energy saving lamps and conventional fluorescent lamps are also culprits when it comes to unwanted RF noise especially in the medium wave band. All of these sources can be easily detected with the integrating LED detector you have built from an Arduino.

By the way, don't overlook the other useful components on the extension shield (**Figure 4**). There are two push buttons and a pot to play with. With a little ingenuity you could make use of these to provide sensitivity adjustment, a Morse key, call button, mute and standby... ⏮

(150307)

Web Link

[1] www.elektor.com/150307

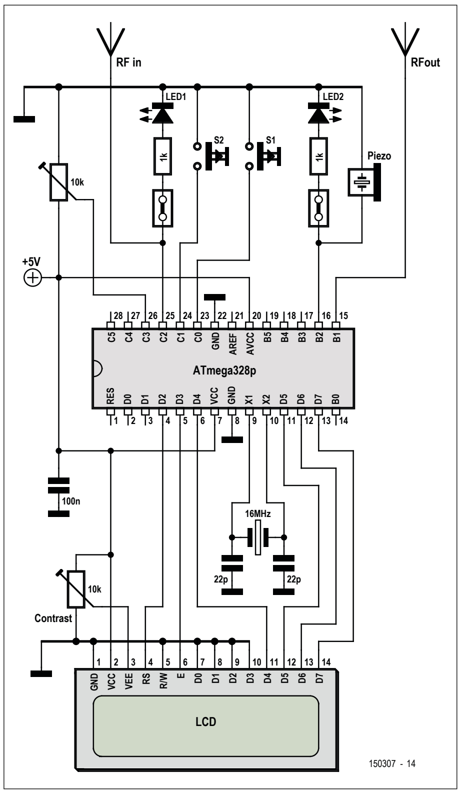


Figure 4. Except for the two antennae and the piezo beeper everything else is already on board the Extension shield.

Listing 2. RF receiver with sound output [1].

```
‘--------------------------------------------------------
‘UNO_RX2.BAS  B1 RF out, C2 RF in
‘--------------------------------------------------------
$regfile = “m328pdef.dat”       ‘ ATmega328p
$crystal = 16000000             ‘ 16 MHz
$baud = 9600
$hwstack = 16
$swstack = 16
$framesize = 16

Dim D As Word
Dim D0 As Word
Dim N As Byte

Ddrb.2 = 1

Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 ,
Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2
Config Lcd = 16 * 2
Cls
Cursor Off

Config Adc = Single , Prescaler = 8 , Reference = Avcc      ‘ 5V

Config Timer1 = Pwm , Prescale = 1 , Pwm = 10 , Compare A Pwm = Clear
Up
Tccr1a = &B10000010             ‘ Phase-correct PWM, Top=ICR1
Tccr1b = &B00010001             ‘ Prescaler=1

D = 4                           ‘ 2 MHz
Icr1 = D
Ocr1a = D / 2

D = 0
For N = 1 To 50
   Portc.2 = 1
   Waitus 100
   Portc.2 = 0
   D = D + Getadc(2)
Next N
D0 = D / 50

Do
  Portc.2 = 1
  Waitus 100
  Portc.2 = 0
  D = Getadc(2)
  If D < D0 Then
    D = D0 - D
    If D > 2 Then
      Print D
      ‘Locate 1 , 1
      ‘Lcd D
      ‘Lcd “    “
      Sound Portb.2 , 20 , 4000   ‘ LED2 and Piezo
    End If
  End If
Loop
```

The RV-3029-C2 "Swiss Clock"

Off-air awfully accurate drop-in RTC & calendar on I²C

By **Danny Winkler** (Design), **Elektor Labs India** (PCB design and software), **Jan Buiting** (Editing)

This RTC & Calendar board is built around Micro Crystal's type RV-3029-C2 integrated circuit. The '3029 is a real-time-clock/calendar (RTC) cheerfully communicating via the I²C protocol. It offers good precision out of the box, and fine precision when tweaked.

Switzerland is home to some of the best cheese, snow, ski tutors, vaults, tunnels and cuckoo clocks in the world. Here we use a product from Swiss company Micro Crystal; it's an electronic clock and very likely a spin-out development of the country's renowned watch industry (both mechanical and quartz).

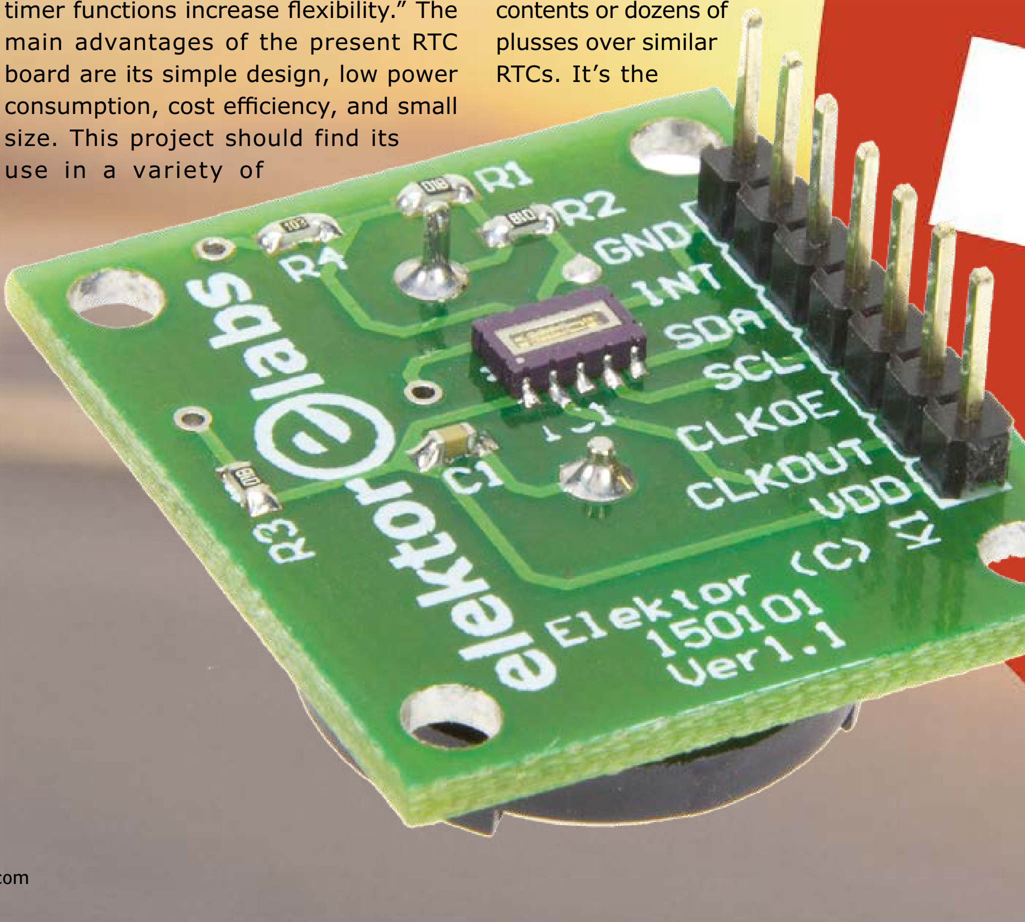
Putting our copy of *Asterix in Switzerland* aside for a moment on the Third Cuckoo, we read in the RV-3029-C2 datasheets [1]: "The RV-3029-C2 is a Real-Time-Clock Module with embedded crystal. This RTC has an I²C Bus (2-wire Serial Interface) and offers temperature compensated time. The STC (Smart Temperature Compensation) is calibrated in-factory and guarantees good time accuracy of ±6 ppm from −40°C to +85°C and ±8 ppm from -40°C to +125°C. Beside standard RTC functions, it includes a Backup-Battery Input with internal switchover function, a programmable Trickle-charge circuitry, an integrated Temperature Sensor with digital output and offers 8 Bytes of RAM and 2 Bytes of EEPROM for a customer application. The calendar function tracks year and leap year flags. The clock tracks second, minute and hour in 24-hour format. Programmable alarm setting and universal timer functions increase flexibility." The main advantages of the present RTC board are its simple design, low power consumption, cost efficiency, and small size. This project should find its use in a variety of applications like Smart Energy, utility metering, remote monitoring, embedded systems, and many others.

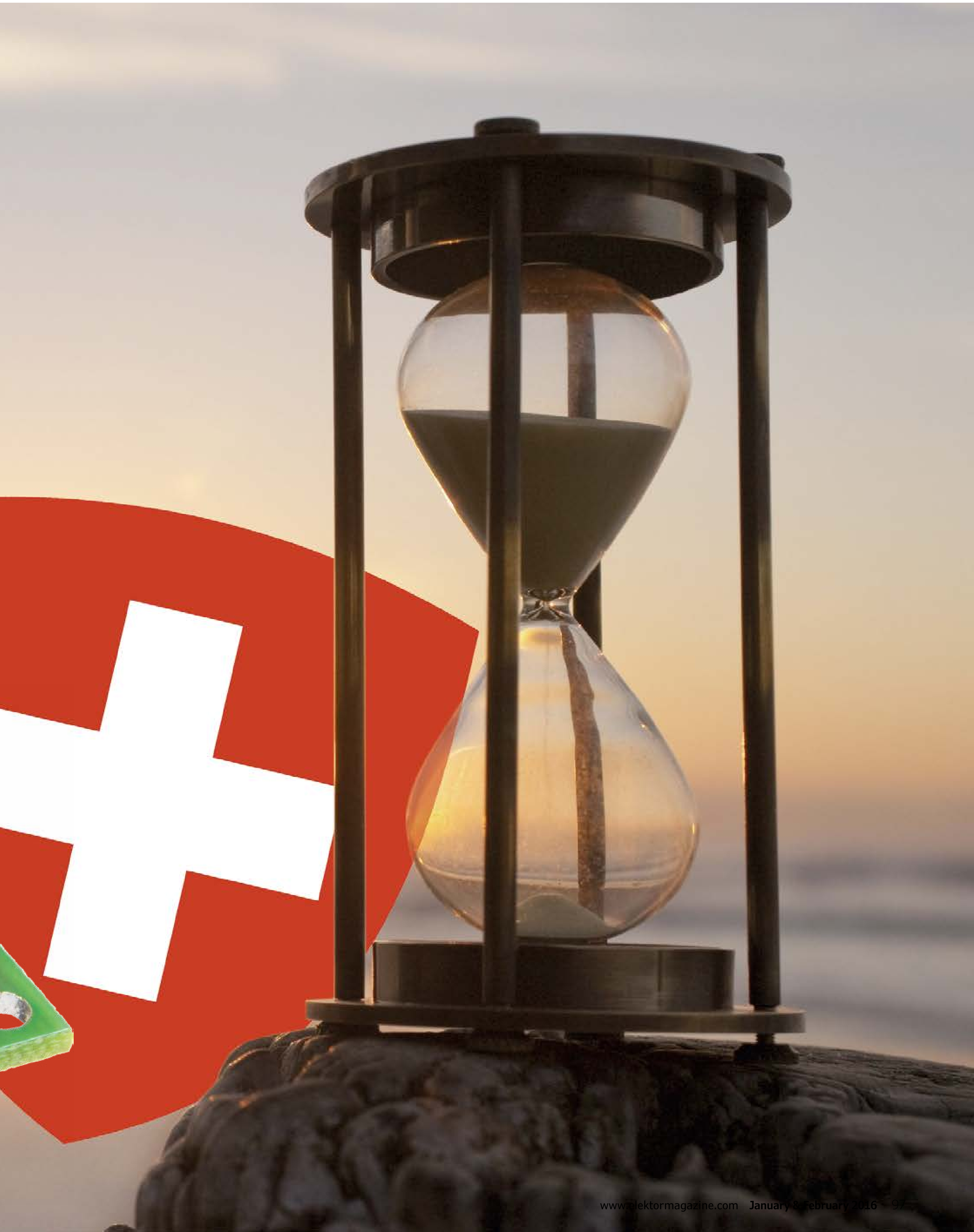
Features

- Micro Crystal RV-3029-C2 RTC/ Calendar chip
- Battery backup
- Selectable clock output
- 6 ppm max. error default
- 3 ppm error achievable with DTCXO tweaking
- I²C connectivity
- Free Arduino test & config utility
- Drop-in module

What's inside

Looking at the block diagram shown in **Figure 1**, admittedly the RV-3029-C2 does not have spectacular contents or dozens of plusses over similar RTCs. It's the





6-ppm (default) DTCXO-derived precision of the clock and the I²C connectivity though that stand out and these two features alone were enough to trigger Elektor Labs to leverage the author's contribution posted on www.elektor-labs.com to the In Progress stage, and have a PCB and demo/test software designed by Labs India.

Circuit description and options

The board aims to be a carrier that brings all of an IC's connectivity out on pins you can solder to, or plug into a breadboard. Hardware-wise, it's a drop-in unit and the fun should start (over) with the programming.
With that in mind, the schematic of the RV-3029-C2 board shown in **Figure 2** is unsurprisingly simple and small. Here we may have reached a BOM record low considering there's not even a quartz crystal to connect to the IC. Being on board the chip, the xtal can be tuned tightly to combat drift due to temperature, shock and ageing.
Version 1.1 of the project is designed to exploit the full functionality of the RV-3029-C2 as set out in its application manual. The module can be used in two configurations, which are important to keep separate.

- **Without backup supply voltage**
  When operating the RV-3029-C2 without a backup supply voltage, it is recommended to hold the VBK pin (#8) low with a 10-kΩ resistor to GND. The pull-up resistor on the INT signal pin (#7) can be tied directly to the supply voltage, VDD. For this configuration, **fit resistors R4 and R3, omit R1 and R2**.

- **With backup supply voltage**
  When operating the RV-3029-C2 with a Supercap or Lithium battery as a backup supply, the INT signal (#7) also works when the device operates from the VBK supply voltage, therefore it is recommended to tie the INT pullup resistor to VBK. When a Lithium battery is used, it is recommended to insert a 100–1000 Ω protection resistor to limit battery current and to prevent damage in case of soldering issues causing shorts between supply pins. For this configuration, **fit resistor R1 and R2; omit R3 and R4**.

Like a Swiss clock...

Time and date are quintessential to many projects, like a simple clock, a data logger or a task scheduler. For this purpose, you can opt to use a radio- or a quartz-controlled clock. A radio-controlled clock needs a radio signal, but what if the device is in the basement, behind tons of concrete or if it's otherwise well shielded? What if a long wave antenna is bothersome?
Then a clock ticking on a quartz crystal is an alternative.
But then a quartz crystal is not particularly stable in terms of temperature. Common crystals are specified to have a deviation up to ±20 ppm and would make a fine using an RTC IC like the DS1307. But what does ±20 ppm mean? It's a way of expressing the quartz accuracy measured in oscillations per million cycles. Mathematically, assuming a 20-ppm clock xtal oscillates 32,768 times at 25 degrees C, the accuracy in seconds per year can be calculated as:

$$\frac{s}{a} = \frac{\cancel{32678} \times secs \times mins \times hours \times days \times ppm}{10^6 \times \cancel{32678}}$$

$$\frac{s}{a} = 31536 \times ppm$$

meaning in this case a dismaying deviation of ±10.5 minutes per year — that's unacceptable and not only in in Switzerland ☺. To add to the woes, there's the added error due to temperature changes.
The RV-3029 RTC has a built in temperature sensor allowing it to achieve (with s/w tweaking) a minimum error of ±3 ppm, i.e. 1.5 minutes slow or fast per year. So anyone wanting to build an accurate clock should allow for the slightly higher cost and go for an RV-3029 (I²C bus) or an RV-3049 (SPI).
I went for RV-3029 mainly because the I²C bus is more common. The RV-3029 can be written to and read out via the bus like an ordinary EEPROM. In addition, it has an interrupt output, which can trigger an interrupt, for example at programmed times. Furthermore, it has a CLKOUT pin supplying a 32,768 Hz, 1024 Hz, 32 Hz or 1 Hz clock signal depending on your setting. I selected 1-second intervals. This is preferable for the RV-3029, as the I²C interface is known to 'hang' when polled in the milliseconds range. In this way the 1-second interval can trigger an interrupt on a regular basis enabling the micro to fetch the time.

It's essential to set bit 5 in register 03h to 0, and bit 0 in register 00h to 1, right when the supply voltage appears — otherwise the clock won't operate. Data for hours, minutes etc. are stored in BCD format for direct displaying on a 7-segment display, or for conversion as follows:

```
IntToBCD        x=INT/16*10
                y=INT/16*16
                y=INT−y
                x=x+y

BCDToInt        x=BCD/10*16
                y=BCD/10*10
                y=BCD−Y
                x=x+y
```

Example: using BCDToInt, BCD 27 → 10|0111 becomes Int 39 → 100111. So, to set minutes to 27, you transmit to word address 09h the '39d' minutes. The complete sequence is: `ACh 09h 27h`.

Even with these options catered for the whole affair boils down to seven external parts: one 16-mm 3-V button cell to keep the clock a-ticking with a backup voltage, the four supply config resistors, one 10-nF supply decoupling capacitor, and finally one pinheader, K1, that brings out the I²C pins SCL, SDA for you to connect embedded stuff to (like an Arduino). Oh and we also thought we'd bring CLKOUT (clock out) and CLKOE (clock output enable) out to pins.

Quick recap on RTC and I²C

The RV-3029-C2 is a CMOS real-time clock/calendar optimized for low, low power consumption. A programmable clock output, interrupt output and voltage low detector are also provided. All address and data are transferred serially via a two-line bidirectional I²C bus. Maximum bus speed is 400 Kbit/sec. The built-in word address register is incremented automatically after each data byte written or read.

The RV-3029-C2 supports the I²C bus protocol that's essentially bidirectional using two wires for communication between devices. The device that sends data onto the bus is defined as a Transmitter and the device receiving data as a Receiver. The device that controls the message is called a Master; the devices that are controlled by the master are referred to as Slaves. A master generates the Serial-Clock (SCL), controls the bus access and generates the START and STOP conditions. The RV-3029-C2 operates as a slave on the I²C bus and can work in standard mode up to 100 kHz clock rate and in fast mode with 400 kHz maximum clock-rate. SCL and SDA are open-drain I/O lines to allow connecting multiple devices. Both lines require connecting to a positive supply voltage via pull-up resistors. Data transfer may be initiated only when the bus is not busy.

In the rare case you're new to I²C, the best book ever published on the protocols and the plethora of devices out there, comes from Elektor: it's *LabWorX 1, Mastering the I²C Bus* by Vincent Himpe a.k.a. Free Electron.

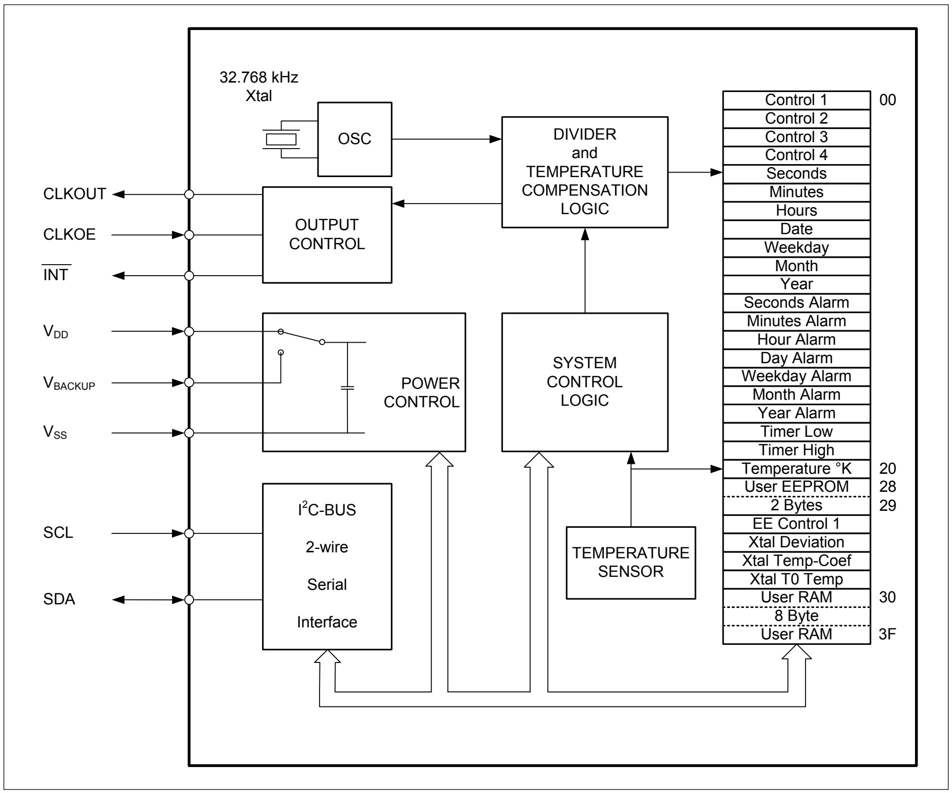


Figure 1. Block diagram of the RV-3029-C2 RTC chip. Source: Micro Crystal datasheet.

Construction

Sadly no sensible price could be negotiated for volume manufacturing of a ready-assembled unit for retailing through the ElektorStore. That leaves home construction from parts and the bare PCB as the only option. The PCB component overlay and parts list are printed in **Figure 3**. Fortunately only the RTC chip is a minor challenge to solder manually, the other parts should be doable observing the familiar methods for handling SMD devices.

The CR1620 button cell and its holder are mounted at the rear side of the board., Figuer 3 again shows the position indicated by the component overlay.

Testing procedure with Arduino

The RV-3029-C2 board can be conveniently tested using your Arduino, which as you know is versatile on I²C. We wrote a Sketch that's available for free downloading from the project web page [2], it's named _150101_i2c_rtc_test.ino. The sketch when compiled (using Wire.h and rv3029.h as helper programs) writes hh:mm:ss to the board and then reads back the time every second. To use it:

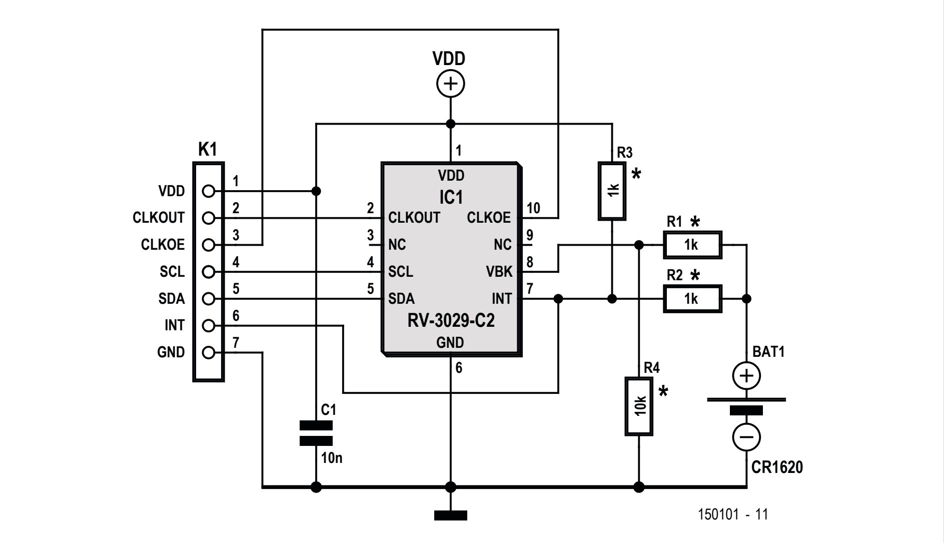


Figure 2. Schematic of the real-time clock module based on Micro Crystal's RV-3029-C2 chiplet.

Component List

Resistors
0.1W, 1%, 75V
R1,R2,R3 = 1kΩ*
R4 = 10kΩ*

Capacitors
C1 = 10nF, ±10%, X7R, 50V, 0603

Semiconductor
IC1 = RV-3029-C2 option B RTC, SMD, I2C, 32.768kHz (Micro Crystal), Farnell # 1641059

Miscellaneous
BAT1 = CR1620, 3V button cell, 16mm diam.
Holder for BAT1, single cell, 16mm
K1 = 7-pin pinheader, 0.1’’ pitch
PCB # 150101 from ElektorStore

* fit or omit depending on application, see text.

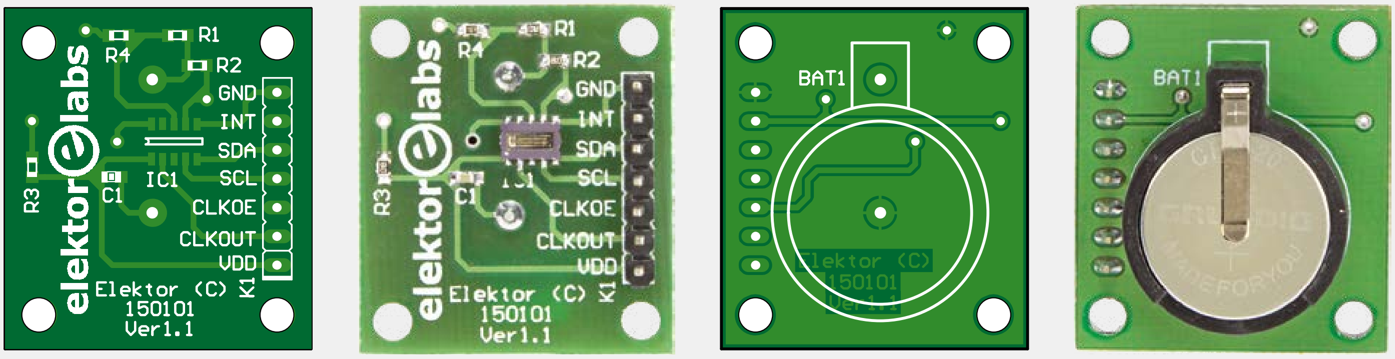


Figure 3. Printed circuit board layout and component list. Assembly is a challenge but doable if care and precision are exercised. Please note that the boards are shown here at 150% of their real size.

- connect the I²C and power connections via connector K1 on the e-BoB;
- upload the Sketch to the Arduino board;
- open the serial monitor at 9600 baud rate.

Now you should see HH:MM:SS on your serial monitor.

A section of the RTC set & read Sketch is shown in **Listing 1**. Here, some of the initialization and #including is done to prep the RTC chip for operation.

(150101)

Web Links

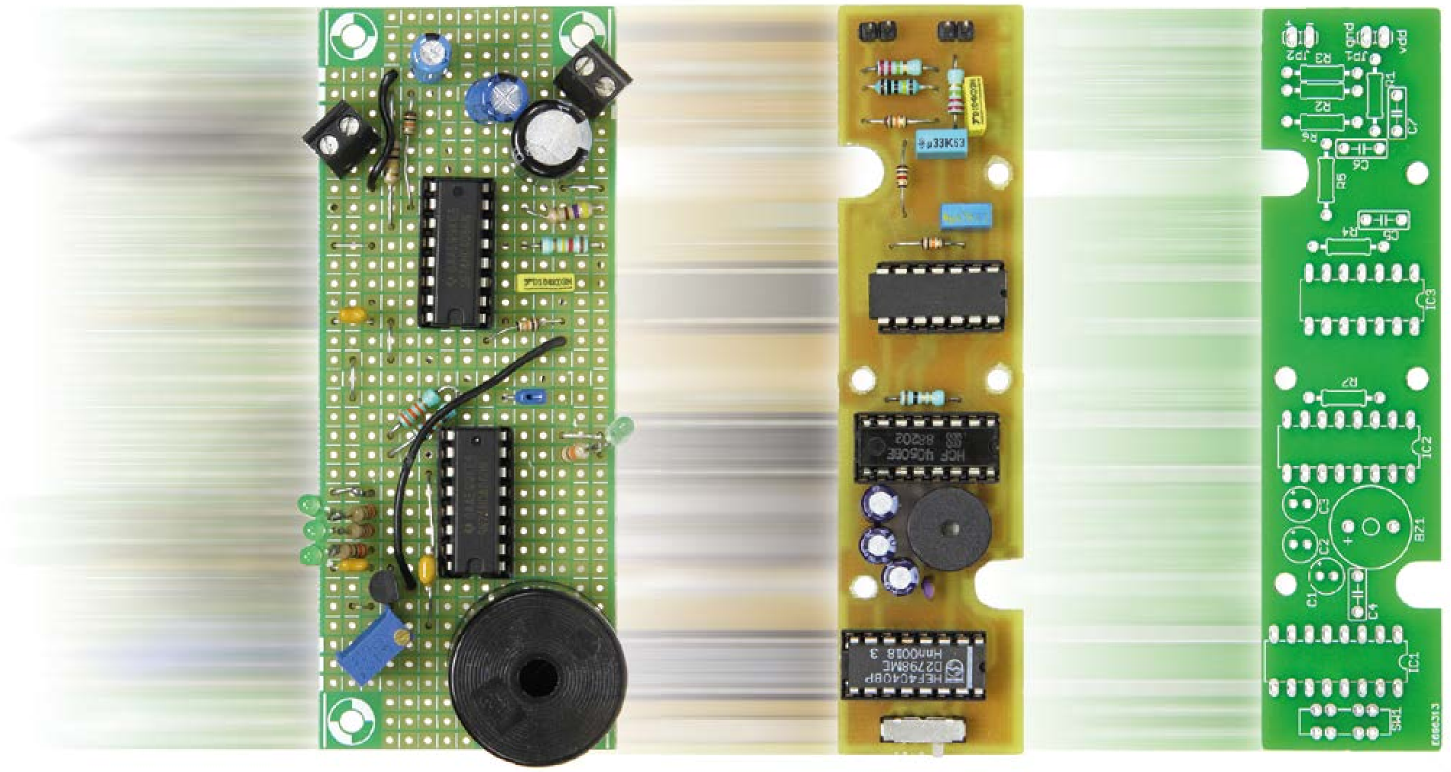
[1] Micro Crystal RV-3029-C2 datasheet: www.microcrystal.com/index.php/products/real-time-clocks

[2] Project’s web page: www.elektormagazine.com/150101

Listing 1. RV-3029-C2 test and control Sketch (extract)

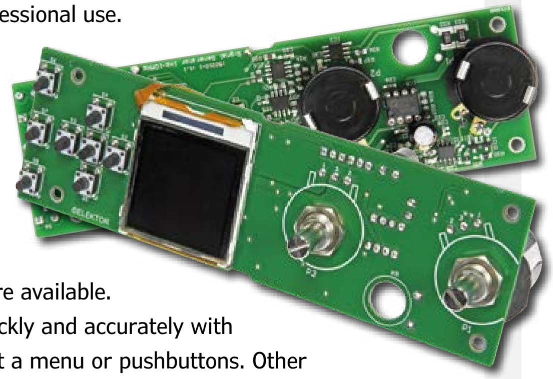
```
#include <Wire.h>
#include “rv3029.h”  // Needed to make the IDE accept our time
struct.

#define I2C_ADDRESS  (0x56)

uint8_t bcd2bin(uint8_t val)
{
  return ((val) & 0x0f) + ((val) >> 4) * 10;
}

uint8_t bin2bcd(uint8_t val)
{
  return (((val / 10) << 4) | (val % 10));
}

void rtc_set_time(uint8_t i2c_address, rtc_time_t& rtc_time)
{
  Wire.beginTransmission(i2c_address);
  Wire.write(0x08);  // Address of seconds register.
  Wire.write(bin2bcd(rtc_time.seconds));       // Set seconds
  Wire.write(bin2bcd(rtc_time.minutes));       // Set minutes
  Wire.write(bin2bcd(rtc_time.hours));         // Set hours
  Wire.write(bin2bcd(rtc_time.day_of_month));  // Set date
  Wire.write(bin2bcd(rtc_time.day_of_week));   // Set week day
  Wire.write(bin2bcd(rtc_time.month));         // Set month.
  Wire.write(bin2bcd(rtc_time.year));          // Set year.
  Wire.endTransmission();
}
```

```
void rtc_read_time(uint8_t i2c_address, rtc_time_t& rtc_time)
{
  Wire.beginTransmission(i2c_address);
  Wire.write(0x08); // Address of first time register (seconds).
  Wire.endTransmission();
  // Request the seven date/time bytes.
  Wire.requestFrom(i2c_address,(uint8_t)7);    // Typecast
                          // required to avoid C++ warning.
  rtc_time.seconds = bcd2bin(Wire.read()&0x7f);
  rtc_time.minutes = bcd2bin(Wire.read()&0x7f);
  uint8_t hours = Wire.read(); // Hours need more processing.
  rtc_time.day_of_month = bcd2bin(Wire.read()&0x3f);
  rtc_time.day_of_week = bcd2bin(Wire.read()&0x07);
  rtc_time.month = bcd2bin(Wire.read()&0x1f);
  rtc_time.year = bcd2bin(Wire.read());

  // Convert hours.
  if ((hours&(1<<6))!=0)
  {
    // 12h format.
    rtc_time.hours = bcd2bin(hours&0x1f);
    // Check PM flag.
    if ((hours&(1<<5))!=0) rtc_time.hours += 12;
  }
  else rtc_time.hours = bcd2bin(hours&0x3f);
}
```



welcome in your ONLINE STORE

EDITOR'S CHOICE



This function generator can be a great addition to a home lab or even for professional use. Sine, triangular and square wave signals with an amplitude from 0 to 16 Vpp (18 Vpp or square wave) are available. Amplitude and DC-offset can be set quickly and accurately with two 10-turn potentiometers i.e. without a menu or pushbuttons. Other settings respond to 8 pushbuttons. A 128x128 pixel display gives a clear view of measurements and menus. The heart of the generator is the AD9834 DDS chip which is controlled by an ARM7-core ADuC7024BSTZ62 micro. The project code was written in C, using uVision from Keil, and can be uploaded in two ways: via JTAG or a USB UART. The PCB fits perfectly behind the front of a stylish case from Hammond. A multi-voltage internal power supply and an AC line filter complete the generator.

Ton Giesberts, Elektor Labs

www.elektor.com/dds-function-generator

Elektor Bestsellers

1. Raspberry Pi 7" Touchscreen Display
   www.elektor.com/rpi-touchscreen

2. The EAGLE Companion
   www.elektor.com/eagle-companion
3. GREEN Membership
   www.elektor.com/green
4. Red Pitaya
   www.elektor.com/red-pitaya-instrument
5. Internet of Things
   www.elektor.com/iot-book
6. Raspberry Pi 37 Sensor Kit
   www.elektor.com/rpi-sensor-kit
7. Android I/O Board
   www.elektor.com/android-io-board

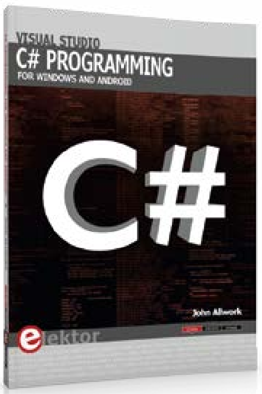
IoT-GET-U-GOING

In 35 fun projects, this book will show you how to build your own Internet of Things system. We'll cover the hardware 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: £25.95 • €35.96 • US $40.00

www.elektor.com//iot-get-u-going

C# Programming for Windows and Android

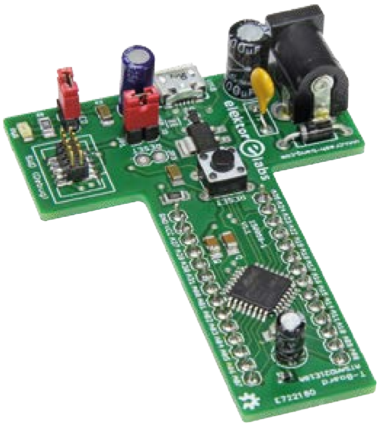


This book is aimed at people who want to learn about the C# language and development environment. It covers steps from installation, the .NET framework and object oriented programming, through to more advanced concepts including database applications, threading and multi-tasking and writing DLLs. The book is based on the Visual Studio 2015 development environment and latest C# additions including WPF applications, LINQ queries, Charts and new commands

member price: £28.95 • €40.46 • US $45.00

www.elektor.com/c-sharp-book

T-Board ARM



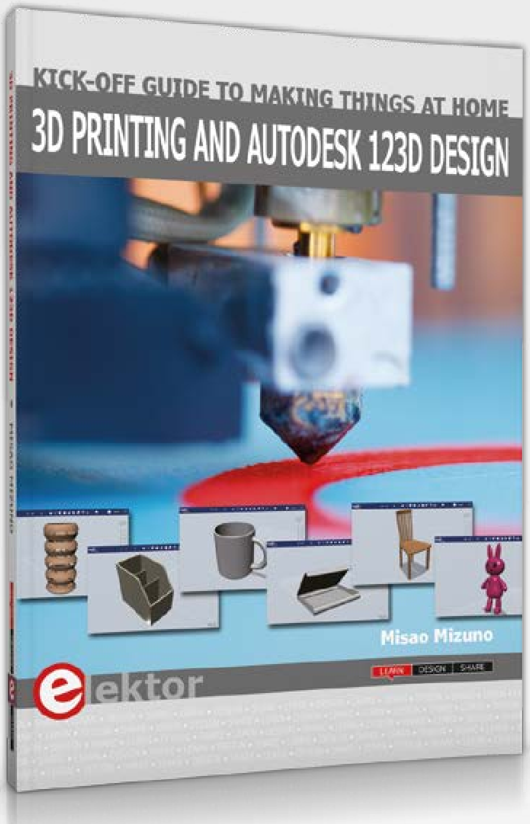
T-Boards have become something of an institution at Elektor, with the original 8-bit AVR versions now a little over a year old. Since then we've also seen wireless and audio T-Boards. Now we introduce the youngest (and most powerful) member of the family: the 32-bit ARM T-Board. This board comes with an Atmel SAM D 32-bit ARM Cortex-M0+ microcontroller (ATSAMD21E18) running up to 48 MHz with 256 KB Flash and 32 KB SRAM.

member price: £16.95 • €22.46 • US $25.00

www.elektor.com/t-board-arm

3D Printing and Autodesk 123D Design

This book provides you with basic knowledge and information about 3D printing technologies so that you can get started. You will learn about the world's latest trends and gain a background on product creation. Also, by reading this book, you will gain a thorough understanding of all crucial information regarding this topic. Such information includes some practical trainings of how to create 3D data that are essential when making things at home using "Autodesk 123D Design," a free 3D CAD. Furthermore, this book will teach you how to print objects using 3D printers and introduce to you features of the wide-ranging 3D printing services available.


KICK-OFF GUIDE TO MAKING THINGS AT HOME
3D PRINTING AND AUTODESK 123D DESIGN
Misao Mizuno
elektor
LEARN DESIGN SHARE

MEMBER PRICE: £27.95 • €38.20 • US $43.00

www.elektor.com/3d-printing

New book on 3D Printing

Limited time offer for GREEN and GOLD members:
15% Discount plus free shipping!

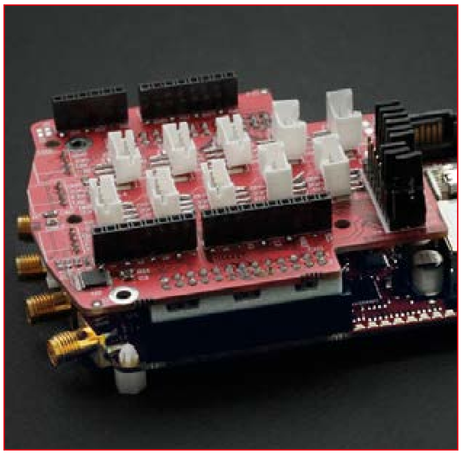
Smart Car Kit for Arduino

Great learning kit: Line-track, light-seek, obstacle-avoid and more!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

Sensor Extension Module for Red Pitaya

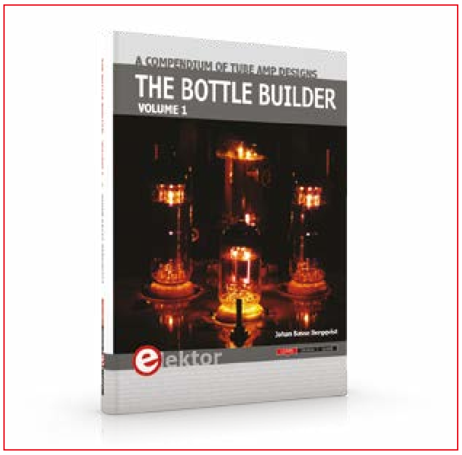


The Red Pitaya Sensor extension module is an add on board that can be plugged in on top of the Red Pitaya board. It enables users to interact with more than 30 sensors, indicators and actuators directly from the Red Pitaya Visual Programming environment. The sensor extension module is not just a perfect add on for newcomers having fun while learning to program and bring their ideas to fruition - but also for developers that need, or would like to prototype their ideas faster.

member price: £16.95 • €22.46 • US $25.00

www.elektor.com/red-pitaya-sensor-module

The Bottle Builder

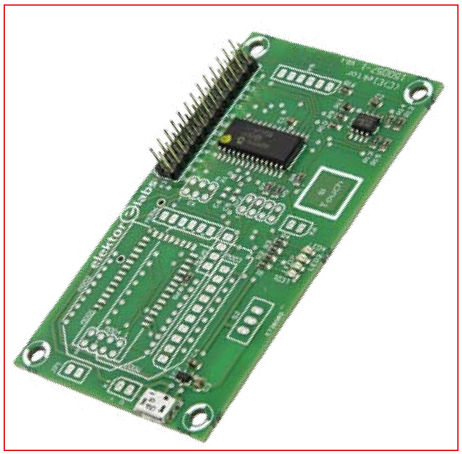


This massive compendium of tube amplifier designs presents an impressive gallery of tube amplifier projects described not just with a personal stance, but also humor, an open mind, good anecdotes, and an equally good emphasis on all the technical aspects of design and implementation. With every project, the focus is on what makes a particular amplifier "special" in terms of design or performance, which as we all know are not necessarily in agreement.

member price: £51.95 • €71.96 • US $80.00

www.elektor.com/bottle-builder

Android I/O Board



The Android I/O Board allows you to control a PIC microcontroller's port pins with your Android device via WiFi, USB or Bluetooth (version 2.0 and 4.0). The specially developed Java classes (libraries) for controlling the Android I/O Board allow you to focus your attention on the application rather than on the handling of complex I/O. Based on a PIC16F1938-I/SO the Android I/O Board offers accessibility to all 25 I/O pins.

member price: £16.95 • €22.46 • US $25.00

www.elektor.com/android-io-board

REVIEW OF THE MONTH

By **Kenn Mills**

The E-blocks Starter Kit Professional for PIC includes a full version of Flowcode 6 for PIC to allow quick and easy development of PIC based systems. It is usable for people who have little experience in programming but well suited for experts as well. Flowcode 6 allows you to build a program as a flow chart, using many ready-made components. This is time saving and the built-in software simulator helps you find bugs.

This low-cost kit contains a USB Multiprogrammer (incl. PIC16F877), a LED board, Switch board and LCD-board. The programming board has a system called DzGhostdz which allows monitoring of your hardware and provides the ability to quickly identify problems. Many additional boards can be purchased separately, allowing the construction of a wide range of systems. Data sheets are available for all boards, including circuits, which will enable you to duplicate the design on your own circuit board. This is a fantastic product for beginners or experts to rapidly develop systems. It will also develop knowledge of a wide range of interfaces and protocols.

Read this review and more about this product at
www.elektor.com/pic-starter-kit

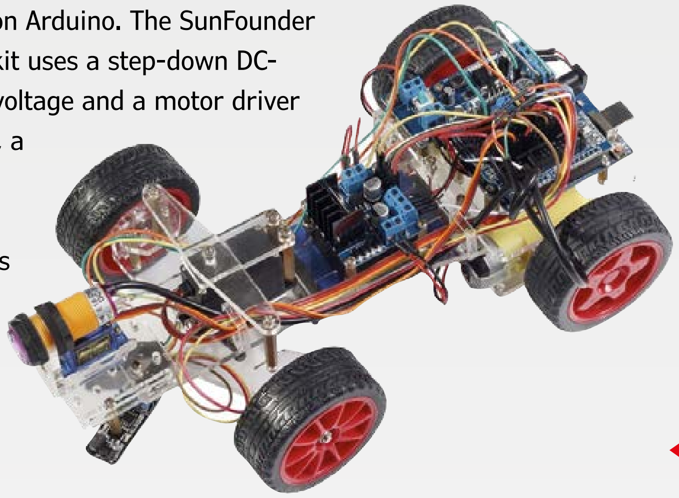
Voucher
elektor
VALUE
€100
The use of this voucher is limited to redeeming in the Elektor Store, www.elektor.com.
Voucher not valid for discount on your Elektor membership.
Note: voucher redeemable once only.
No outstanding balance awarded for total order values under €100.
Voucher not redeemable against cash. Voucher valid through December 31, 2015.

Submit a review of your favorite Elektor product and qualify to win a €100 voucher for redeeming in the Elektor Store.

For further information, please visit www.elektor.com/rotm

Voucher
elektor
VALUE
€5
CODE: EL-2016
The use of this voucher is limited to redeeming in the Elektor Store, www.elektor.com.
Voucher not valid for discount on your Elektor membership.
Note: voucher redeemable once only.
No outstanding balance awarded for total order values under €100.
Voucher not redeemable against cash. Voucher valid through February 29, 2016.

This is a complete learning kit based on Arduino. The SunFounder Uno R3 board is used as control. The kit uses a step-down DC-DC converter module to reduce input voltage and a motor driver module with L298N. More importantly, a sensor shield is used so that more I/Os can be led out from SunFounder Uno, which means a lot more functions can be expanded! With the sample sketches provided, you can make the car move forward/backward and turn left/right, follow a line, seek light and avoid obstacles.



Component List

- 1 pack x Acrylic Plates
- 1 pack x Threaded fasteners
- 1x Tower Pro Micro Servo SG90
- 1x Tower Pro Micro Servo MG995
- 2x Gear Reducer
- 4x Wheel
- 1x SunFounder UNO R3 board
- 1x DC Motor Driver Module
- 1x Step-down DC-DC Converter Module
- 1x Switch Module
- 1x Photoelectric Switch
- 4x Photosensitive Module
- 2x Obstacle Avoidance Module
- 5x Tracking Module
- 1x Dual 18650 battery holder
- 2x Ribbon (13cm)
- 1x USB Data Cable
- Several Dupont Wires
- 1x Screwdriver

MEMBER PRICE: £80.95 • €112.46 • US $124.00

www.elektor.com/smart-car-kit-for-arduino

New book on 3D Printing

Limited time offer for GREEN and GOLD members:
15% Discount plus free shipping!

Smart Car Kit for Arduino

Great learning kit: Line-track, light-seek, obstacle-avoid and more!

Elektor Store

An Aladdin's Cave of books, kits, gizmos and more. Fill your shopping cart today!

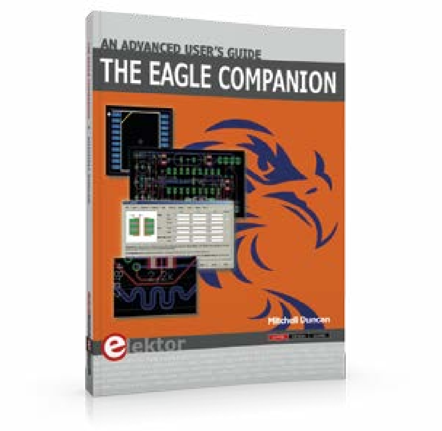
Internet of Things

The Internet of Things (IoT) is a new concept in intelligent automation and intelligent monitoring using the Internet as the communications medium. This book is written for students, for practising engineers and for hobbyists who want to learn more about the building blocks of an IoT system and also learn how to setup an IoT system using these blocks.

member price: £25.95 • €35.96 • US $40.00

www.elektor.com/iot-book

The EAGLE Companion

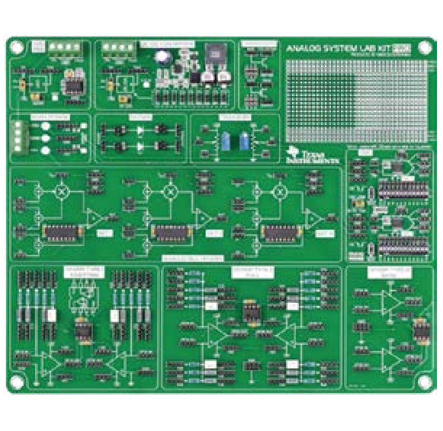


In over 600 pages, author Mitchell Duncan serves EAGLE users requiring deeper insights in the program's operation, functionality, and advanced features. EAGLE's hidden gem is its User Language Program (ULP) which is a gateway to customizing the program for individual needs and preferences. This full ULP manual is included in the book in unabridged form. If ever EAGLE had a faithful companion, it's truly this book!

member price: £28.95 • €40.46 • US $45.00

www.elektor.com/eagle-companion

Analog System Lab Kit



This kit is designed for undergraduate engineering students to perform analog lab experiments. The main idea behind Analog System Lab Kit PRO is to provide a cost efficient platform or test bed for students to realize almost any analog system using general purpose ICs such as OP-Amps and analog multipliers. ASLK PRO comes with three general-purpose operational amplifiers (TL082) and three wide-bandwidth precision analog multipliers (MPY634) from Texas Instruments.

member price: £83.95 • €116.96 • US $129.00

www.elektor.com/aslk-pro

Welcome to the SHARE section

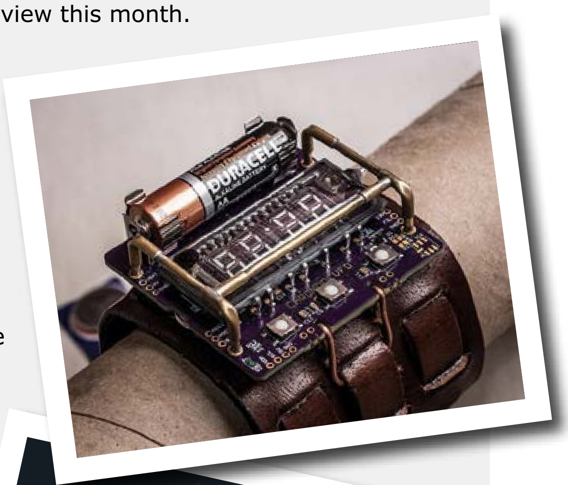
By **Thijs Beckers**
thijs.beckers@eimworld.com

Shared joy is double joy!

Problems at home? Elektor Labs can help. With electronic problems, that is. From water management to telephone nuisance? But of course not only that. There are many more interesting and useful projects to be found on our Labs website (www.elektor-labs.com). If there is a place that deserves the 'share' label, then our community website certainly is. The highlights page a little further on in the Share section shows merely the tip of the iceberg.

Of course, the 'new' Retronics is also one of the party. This time Editor-in-Chief and Scope Guru Jan Buiting talks about the super-sensitive Tektronix Type 503 oscilloscope, which he got as a gift (well almost) from a sympathetic reader of Elektor's famed 'vintage' pages. Why the 450 kHz bandwidth (yes, you read that correctly!) is a feature and not a bug you can read in this month's *Retronics* article.

You may already know that the Red Pitaya platform is a powerful open-source measuring instrument the size of a credit card. This *community-driven FPGA-powered* measuring instrument proves that the **sharing** of information can lead to great results! (Tell that to the *maker-community*...). You can read about the features of the recently introduced software extensions in the review this month.



Have you heard about *Steam Punk*? This culture does not only nourish artists and creative spirits, but also electronics enthusiasts, with ideas for projects and in some cases leads to wonderful contraptions. Harry Baggen checked out a few remarkable examples for you. Marvel at the steam-powered record player and the 3D-printer in an antique traveling trunk. With thanks to the most extensive network ever for sharing information: the Internet.
We share some more 'inside information' from the Labs with you in the small article about sparks from a lighter, which can, more or less, be used as a generator — an accidental discovery from Elektor Labs worker Luc Lemmens. Piqued your curiosity about all of this? Quickly continue reading... ◀

(150564)

Circuit Triggered by Sparks

By Harry Baggen (Elektor Labs)

A lighter as a generator? When you're in need of a light source that produces short pulses across a wide spectrum, then this could be just what you need. Even if you don't smoke (anymore)!

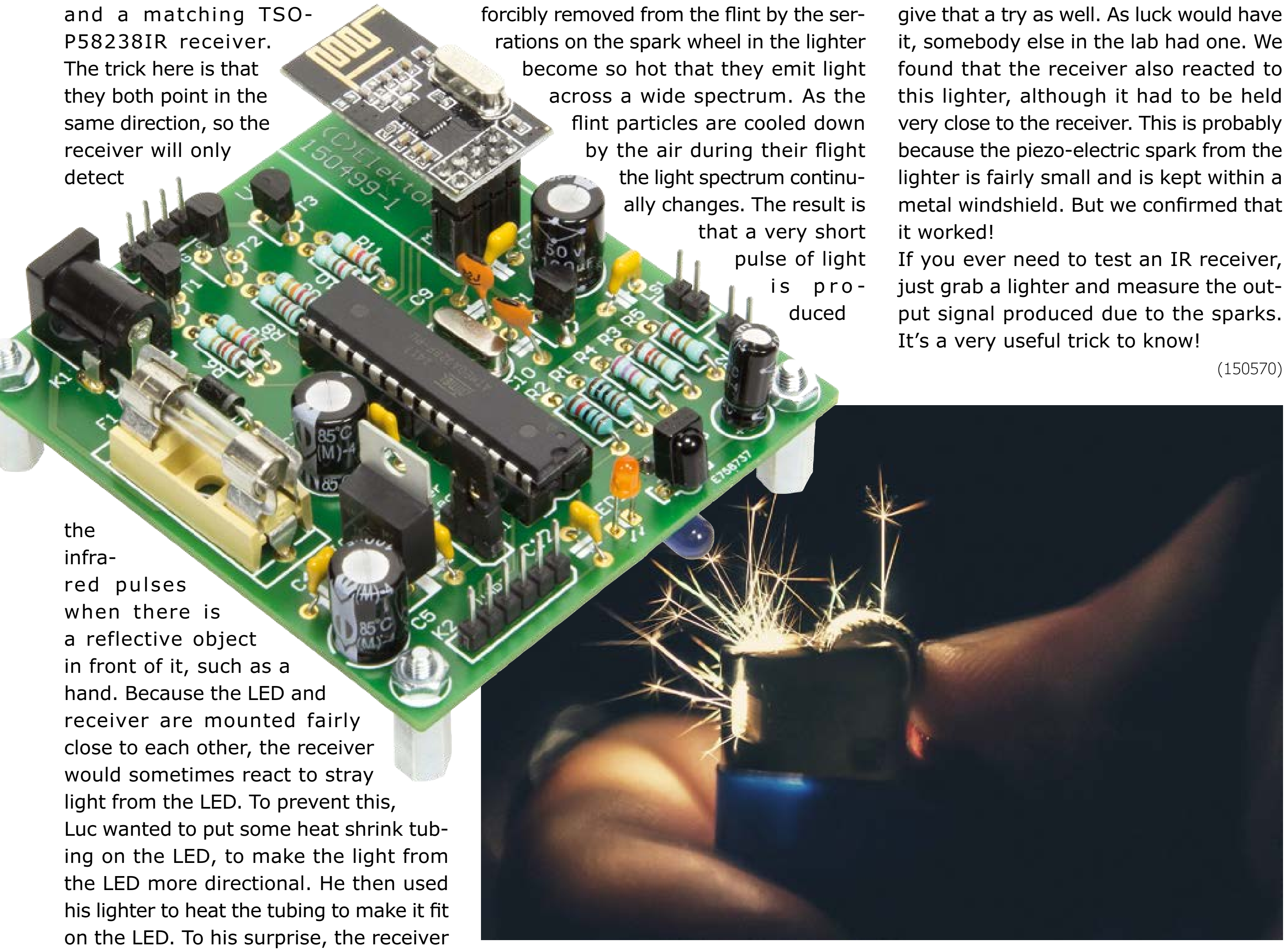
Sometimes (or often?) you come across an interesting incident purely by chance. No, we haven't seen a UFO fly over Elektor House! In this case, it really concerns a technical observation in our lab. Lab engineer, Luc Lemmens was hard at work on a circuit for this issue (Wireless Quiz Buttons RGB Style), and he was experimenting on the sensitivity of the infrared connection on the board designed for this circuit. It consists of an IR LED that transmits IR light pulses and a matching TSOP58238IR receiver. The trick here is that they both point in the same direction, so the receiver will only detect the infrared pulses when there is a reflective object in front of it, such as a hand. Because the LED and receiver are mounted fairly close to each other, the receiver would sometimes react to stray light from the LED. To prevent this, Luc wanted to put some heat shrink tubing on the LED, to make the light from the LED more directional. He then used his lighter to heat the tubing to make it fit on the LED. To his surprise, the receiver reacted to the light from the lighter (the circuit was still turned on). After he made the heat shrink tubing fit snugly on the LED, he experimented some more with his lighter. He discovered that the IR receiver reacted to the sparks from the flint when the lighter was a much as 20 or 30 centimeters away. To be clear, it wasn't the flame, which could stay off!

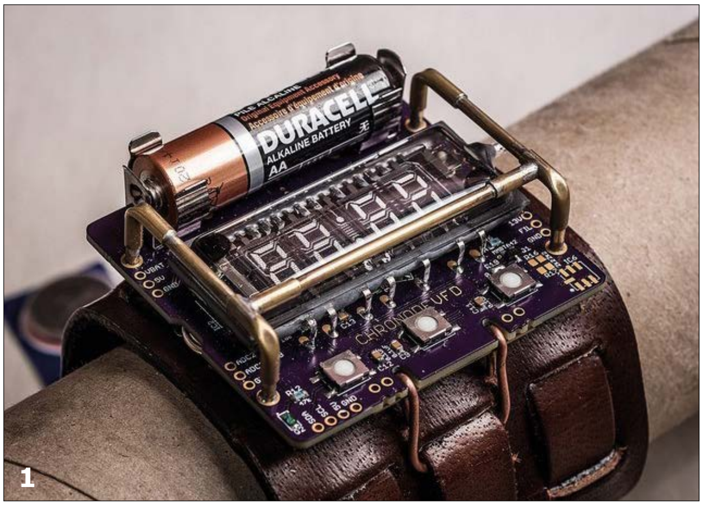
The explanation is understandable, but it's not something that would come to mind straight away: The particles that are forcibly removed from the flint by the serrations on the spark wheel in the lighter become so hot that they emit light across a wide spectrum. As the flint particles are cooled down by the air during their flight the light spectrum continually changes. The result is that a very short pulse of light is produced across a very wide spectrum. This also includes a good amount of infra-red (the optimum sensitivity of the IR receiver is at 950 nm), which the receiver detects. This IC does include several other filter circuits, but the light pulses from the flint particles have no problem passing through, since no further decoding is carried out by the circuit in this case (they wouldn't work on a TV infra-red circuit).

In theory, this should also work with a piezo-electric lighter, so we wanted to give that a try as well. As luck would have it, somebody else in the lab had one. We found that the receiver also reacted to this lighter, although it had to be held very close to the receiver. This is probably because the piezo-electric spark from the lighter is fairly small and is kept within a metal windshield. But we confirmed that it worked!

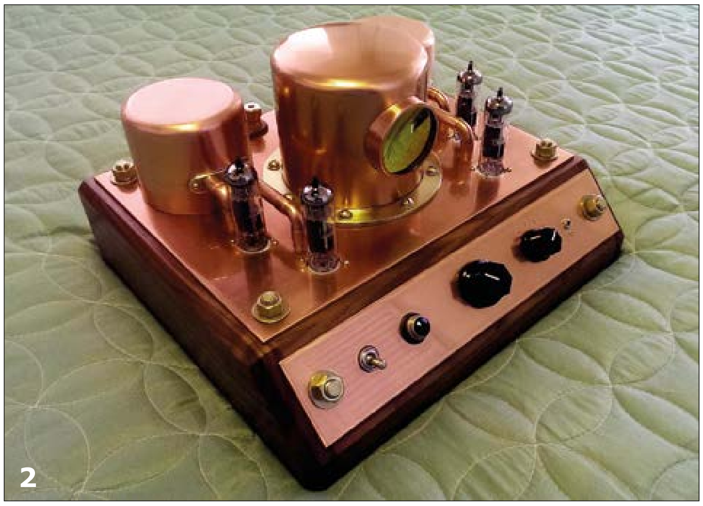
If you ever need to test an IR receiver, just grab a lighter and measure the output signal produced due to the sparks. It's a very useful trick to know!

(150570)





1



2

Steampunk Electronics

It's all in the presentation

By Harry Baggen (Elektor Labs)

How do you fit a modern device in a nineteenth-century jacket? When we search on the Internet for so-called Steampunk designs it's obvious that many people are engaged with this topic, including electronics enthusiasts. Here is a selection of remarkable electronics designs.

When a colleague dropped the term Steampunk a while ago I had no idea what he was talking about. After some explanation and a little searching on the Internet I was not only enlightened but also surprised to see how many people used to, and still do, keep themselves busy with this subject. First a short explanation for those who are not familiar with what this is (free after from Wikipedia): Steampunk is a subgenre of science fiction and sometimes fantasy that incorporates technology and aesthetic designs inspired by 19th-century industrial steam-powered machinery. These are supplemented by futuristic inventions and machines, or actual existing machines (such as computers), which in these stories are invented much earlier than in reality.

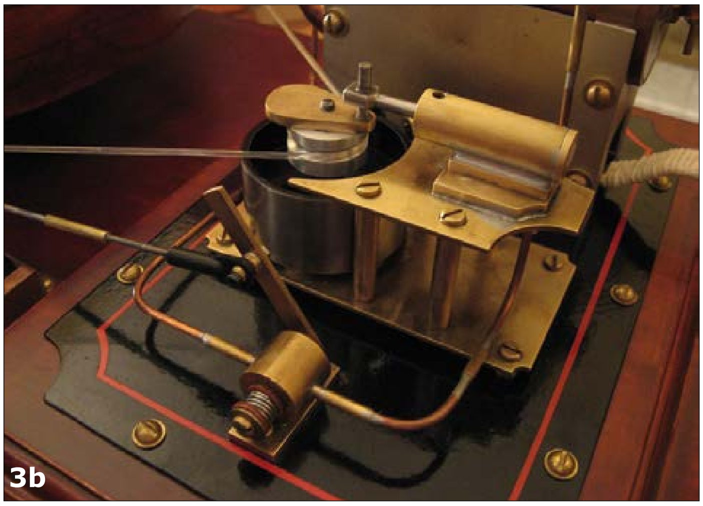
These days Steampunk is a culture of its own, with appropriate clothing and devices that reflect the style. There are even special Steampunk stores that sell clothes, masks, jewelry, etc.

There are also many technologically savvy individuals who build devices in the Steampunk style. These can be existing devices in Victorian guise or entirely newly developed devices with a Steampunk appearance. Once you start to browse around on the Internet you come across the most unusual and wonderful objects. I will show a few of them here.

A handsome example of 'minimalist' Steampunk is the ChronodeVFD-watch made by J. M. De Cristofaro, better known as 'johngineer' [1]. For his own use he developed a digital watch based on a small vacuum-fluorescent display, hence the name. He designed the circuit board for the watch in such a way that there was room to mount a small guard. This not only serves to protect the display, but also to give the watch — as he calls it himself — a cyberpunk appearance. The entire device is very handsome, with a leather wrist strap and a small brass buckle. In one way or another it is complemented by the Duracell battery, because of its black/bronze appearance.

Mike Cochran, owner of the American company CopperSteam, has completely specialized in the development and production of audio equipment and sculptures in Steampunk style. A particularly beautiful valve amplifier is the SteamAmp II [2]. The enclosure finished in wood and copper is impressive to see. The transformers are covered by boiler-like enclosures which also have pipework connected to them. The middle 'boiler' even has an inspection window on the front. It is as if you're looking at some kind of triple-stage steam engine. Do you still have time to enjoy listing to music after admiring such a device?

And while we're on the subject of audio: I find the record player made by Simon Jansen to be one of the most remarkable Steampunk devices, the Steampunk-o-phone [3]. Simon didn't actually intend to build a Steampunk-like device, but a record player that was driven by a small steam engine. Everything was designed and made by himself, including the entire steam engine. Because an old-fashioned record player with horn loudspeaker didn't turn out to work properly, in the end the decision was made to use a modern version with a lightweight tone arm and stylus. The turntable is really driven by a small steam engine, there is no small drive motor hiding anywhere! To control the speed there is an Arduino, which measures the RPM and based on that controls the pressure in



the steam boiler. The control is not optimal and the designer agrees that it doesn't sound the best, but what does it matter? It is a fantastic design!

For the computer enthusiasts a Steampunk laptop is certainly a nice gadget, if you are prepared to spend the money on it. Richard Nagy has specialized entirely in this and in the past few years has built numerous variations of laptops, monitors and keyboards in Steampunk style. A number of these are even for sale from the Datamancer website [4]. The laptops in particular are small treasures and everything still works as it should. The lid is even decorated with a plate that has an engraved image (go to the website for more details). At the moment there are no laptops available, but there are still several different keyboards and monitors.
And finally, what do you think of a 3D-printer in Steampunk style? After purchasing a 3D-printer, its owner John Davis, at some time, thought it would be a nice idea to give the printer an extravagant enclosure in the shape of an antique wooden steamer trunk from the nineteen twenties [5]. He had to make a lot of changes to the trunk (and the printer) so that it would fit nicely and continues to operate correctly. The trunk can actually be closed, where the two leather belts hold everything nicely together. A few old analog meters give everything a more technical appearance. The only modern thing that is visible from the outside is the four-line display.

A record player that's actually powered by a small steam engine

Perhaps all these Steampunk devices are not really all that practical (or not usable at all), but they are often terrific eye candy. And that is certainly already worth the effort!

(150575)

Weblinks

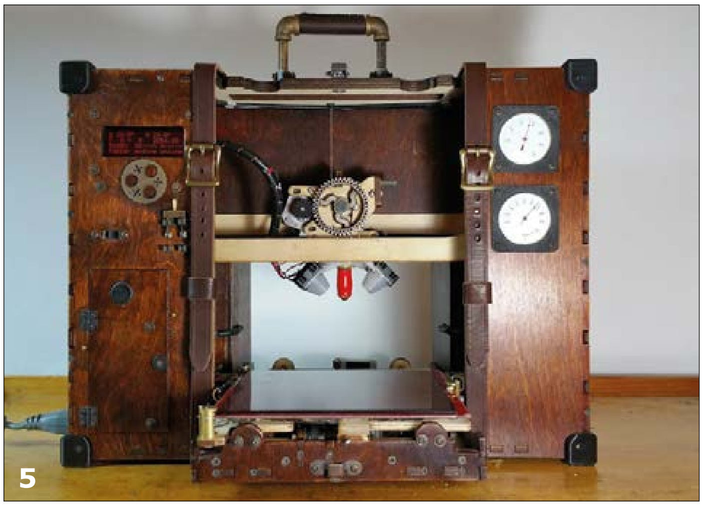
[1] www.johngineer.com/blog/?p=1595

[2] http://coppersteam.com/

[3] www.asciimation.co.nz/bb/2010/11/30/a-steam-punk-record-player

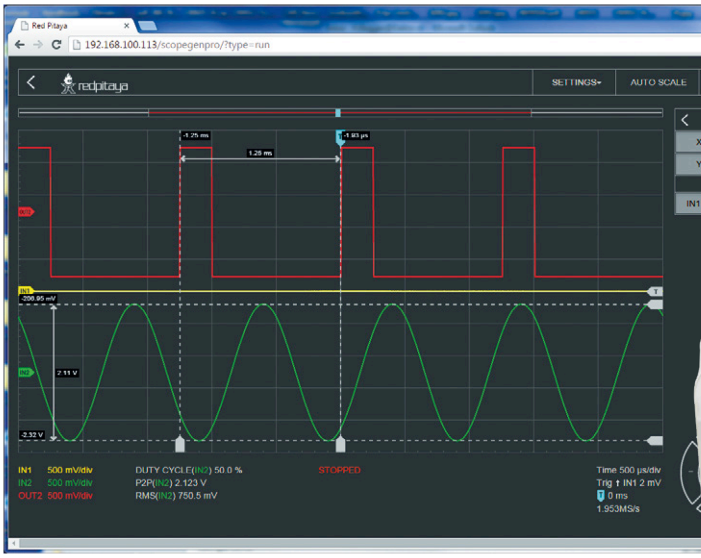
[4] www.datamancer.net/steampunklaptop2/steampunklaptop2.htm

[5] http://blog.atmel.com/2015/09/17/maker-creates-a-steampunk-inspired-3d-printer/



New Apps for the Red Pitaya

Oscilloscope/generator and spectrum analyzer



IN1 500 mV/div
IN2 500 mV/div
OUT2 500 mV/div
DUTY CYCLE(IN2) 50.0 %
P2P(IN2) 2.123 V
RMS(IN2) 750.5 mV
STOPPED
Time 500 µs/div
Trig ↑ IN1 2 mV
T 0 ms
1.953MS/s

By Harry Baggen (Elektor Labs)

The Red Pitaya has become a popular board among Elektor readers, which can be used for measuring purposes as well as for various applications and experiments of their own. In this article we will look at a few new apps that the team from Red Pitaya have recently introduced.

Red Pitaya is the name of a powerful, open-source experimenting board that is built around a Zynq-FPGA made by Xilinx, which, in addition to the FPGA part, also contains a dual-core ARM Cortex A9 processor. The board is ideal for use in various measuring applications thanks to the presence of dual A/D and D/A converters with a resolution of 14 bits and a maximum sampling rate of 125 Msamples/s. The processor part runs Linux, and while that is a familiar environment for many, writing your own applications nevertheless requires that you have a considerable knowledge of programming (refer to the article 'Red Pitaya' in the December 2014 edition [1]). The developers from Red Pitaya have realized this too and therefore have recently unveiled a very simple programming environment for this board that works using building blocks. Using this, beginners and even children can put a program together quickly. This language 'Visual Programming' looks, at first glance, like the programming language 'Scratch', but offers a much greater number of features. One of the remarkable features is that the building blocks can be easily combined with other programming languages such as Python and C/C++.

However, we will save the in-depth look at this programming language for another time. What we would like to look at in this article is how to use the Red Pitaya as a 'normal' measuring instrument. Many electronics enthusiasts will probably purchase a Red Pitaya board because of its excellent measurement specifications and initially start with a few of the standard applications, such as the scope and the generator. So, not actually programming anything yourself. For this type of user the developers have created a few Pro-apps, which are not free, unlike the other apps in the Bazaar (the 'store' that collects all the official and user-contributed apps). At this moment there are two: an oscilloscope/function generator and a spectrum analyzer. They cost €49 and €39 (ex. VAT) respectively and can be purchased from, among other places, the Elektor Store. Exactly these functions are those most used by electronics designers. Manuals are not available (yet), but this article will describe nearly all their settings and features.

> There is also a PWM signal where you can set the duty cycle

Getting started

Okay, let's get started. With most USB scopes you install the PC software and you can then connect the instrument using a USB cable. This is different with the Red Pitaya because it doesn't work over USB but uses a network connection instead.

Before you start with the board you will first have to decide which way you are going to use it: wired or wireless, connected to an existing network or as a stand-alone access point (handy when using while on the road). After making your choice you will need to download an image of the operating system and copy it to a micro-SD card and subsequently, depending on how the Red Pitaya is to be connected to your network, you may have to modify a few files. How this is done is explained in detail on the Red Pitaya website for the various network configurations. We first tried it with a wired connection to our network at Elektor House and then via a wireless connection using a WiFi dongle. After a few initial problems we nevertheless got it up and running quite quickly, but it is quite a different procedure than what you are accustomed to. You can now connect to the Red Pitaya using a web browser. The advantage of this is that you can use this board from any operating system and any device: Windows, iOS, Android, PC or smartphone, they all work.

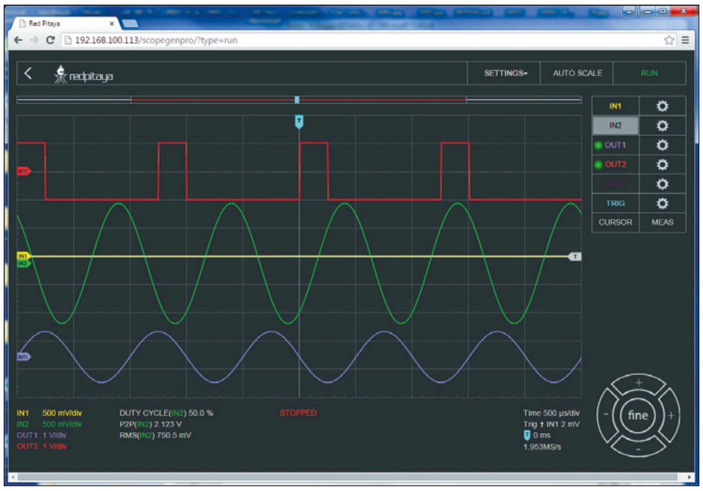
A remark regarding the connections to the Red Pitaya: The board is provided with the less common SMA connectors (mainly chosen to keep the dimensions of the board small). Therefore make sure you have four SMA to BNC adapters on hand so that you can easily connect standard BNC probes and cables to the inputs and outputs.
The initial window that appears in the browser shows which apps are installed on your Red Pitaya. A small lock symbol is displayed next to the paid-for apps. You can unlock such an app by buying a code and subsequently enter it under 'My RP' after logging in. You can also see which IP address on the network was assigned to your RP.

Oscilloscope

After starting the oscilloscope app (**Figure 1**), a grid appears with a few user interface elements around it on a gray background (typical for a paid-for app from Red Pitaya; the free apps generally have a white background). Note that you can try the app out in demo mode first before you buy. To the left of the grid there are a few buttons for operating on the various input and output signals, as well as the triggering and a few measurement options. Below the grid are the time and sensitivity settings plus some other information. At bottom right there is a kind of four-way operating button (cursor) where you can change the scales and the values of a number of other settings. In the top bar there is also a settings button (which at the moment only contains the calibration menu), an auto scale button and a run/stop button.

Most of the options in the relatively plain (but therefore very restful) appearance of the screen are self-explanatory. After clicking the auto scale button the input signals will quickly appear on the screen and it is noticeable that everything on the screen appears to be very responsive, despite the fact that we are working via a network connection (the amount of traffic on the network will certainly also have an effect here).

It takes a little bit to get familiar with the settings on the right-hand side. If, for example, you want to change a setting for input signal IN1, then you have to click on the little gear symbol to the right of it. There now appears a sub-menu where the signal can be turned on or off or inverted, you can change the offset, the attenuation of the attached probe and select the input sensitivity of the board. The latter is not something that you expect with a (USB) scope. Normally you can switch the input sensitivity from within the software, but that is not possible with the Red Pitaya because it doesn't have the necessary relays or electronic switches on the board. This is not a problem in a fixed setup because you can take that into account. But when used as a general-purpose scope this is an issue you will always have to keep in mind! As an alternative you can always begin by setting the probe in the 10x attenuation position before making a measurement. In combination with the 1-V sensitivity setting on the board you will then have an input range of 100 V peak-to-peak.



192.168.100.113/scopegenpro/?type=run

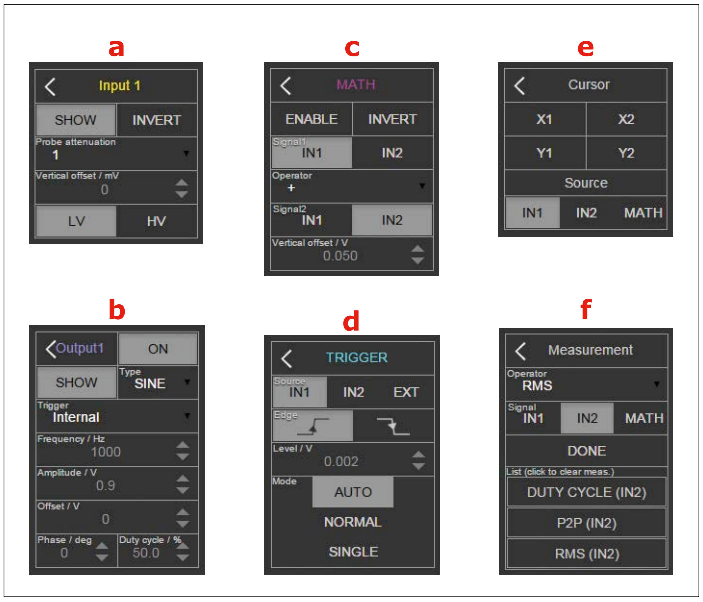
IN1 500 mV/div
IN2 500 mV/div
OUT1 1 V/div
OUT2 1 V/div

DUTY CYCLE(IN2) 50.0 %
P2P(IN2) 2.123 V
RMS(IN2) 750.5 mV

Time 500 µs/div
Trig ↑ IN1 2 mV
0 ms
1.953MS/s

1

For the outputs we can select from various predefined wave shapes. All the common signals can be found here. In addition to frequency we can naturally also set the amplitude, but also the offset and phase displacement. There is also a PWM signal where you can set the duty cycle. It is nice that the wave shapes can be displayed immediately, without first having to make a connection between the output and input.



2

By clicking or touching the input or output name (**Figures 2a and 2b**) you can changes the settings of these with the cursor keys at bottom right. It is unusual that the sensitivity can be set individually for each signal. It is therefore possible to set four different sensitivities; the time scale, however, does apply to all signals simultaneously. Remarkable is also the fine-button between the cursor buttons. When you click on this the settings become 100x smaller. This is perhaps useful for some applications?

With the 'Math' button (**2c**) mathematical calculations can be performed on or between the signals, such as add and subtract, absolute value and *dy*/*d*t. The calculated signal then appears on the screen. In all likelihood the number of available functions will be expanded in the future.
Under the 'Trig' button (**2d**) we find the standard options, such as rising and falling edge, trigger level and auto/normal/single. The button 'Cursor' (**2e**) leads to the options to display one or two vertical and horizontal cursor lines, between which the time and voltage respectively are measured. You can position these cursors in any desired location.
Finally there is the button 'Meas' (**2f**), which is used to display various measurements on the inputs on the screen. There are currently 8 options, such as peak-to-peak value, RMS value and frequency. Multiple measurement options can be selected simultaneously.

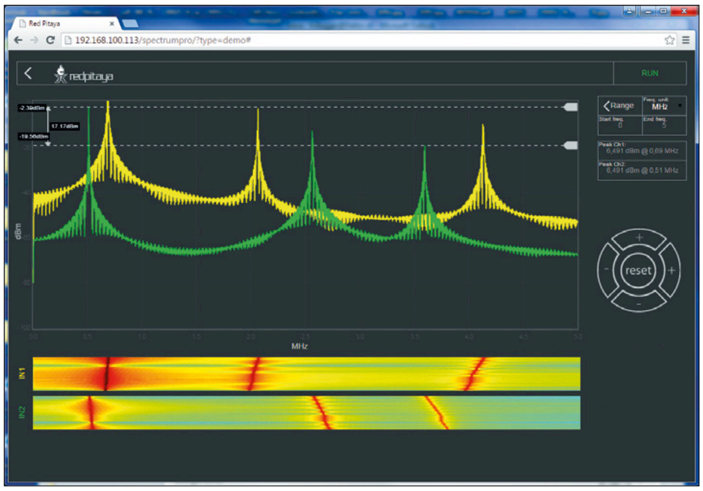
These are the features that the scope instrument currently has to offer. These are mainly the things that you will also find on most other (USB) scopes, but the designers are still working on expanding the number of features and those will become available in due course. In any case, this is sufficient for all basic oscilloscope work.

Spectrum analyzer

About the spectrum analyzer we can be quite brief, since the design and operation corresponds to that of the oscilloscope (**Figure 3**). The arrangement of the grid is different of course: The vertical scale gives signal strength in dBm, the horizontal scale the frequency in Hz, kHz or MHz. In the top bar there is now only the Start/Run button. Below that are the settings for the two input channels. Here you can choose whether a channel is to be displayed or not or whether it has to be 'frozen'. Under the button 'Cursor' we find the same options as with the scope: two horizontal and two vertical measuring lines in between which the values are displayed. Under 'Range' the frequency unit can be selected and you can manually enter the start and end frequencies. Below these fields the measured peak values and corresponding frequencies for the two input channels are continuously displayed.

Using the cursor buttons at bottom right the *X* and *Y* scales can be moved to a smaller range if we want to zoom in for a closer look. With the reset button in the middle we return to the default settings.
Below the measurement field there are two bars, which show the change of the spectrum over time in color. There is no scale displayed, but it nevertheless gives a fair impression of what is happening to the signal.

That is the spectrum analyzer so far. The feature set is quite limited and we missed, among other things, the choice between different sampling windows, but here too the designers have assured us that they are still working on expanding the feature set.



3

The new apps are a great development and make it possible to use the Red Pitaya as a universal measuring instrument, without having to delve into the details of the hardware or put a program together yourself. Hopefully the developers will surprise us in the near future with all kinds of additional functionality for these apps. ◀

(150595)

Web Link

[1] www.elektormagazine.com/140277

Warm and Humid

Recalibration of ChipCap2 sensors

How much is the accuracy of a sensor affected by the soldering of it? Here we share a few of the things we experienced with the temperature/humidity sensor that is on the CC2-BoB.

By **Luc Lemmens** (Elektor Labs)

The ChipCap2 temperature/humidity sensors from Amphenol Advanced Sensors are affordable, completely calibrated and temperature-compensated components with a high degree of accuracy. Furthermore, they offer the choice of an analog (PWM) or digital (I²C) output, which makes them ideal for use in all sorts of electronics projects. The only disadvantage of this particular sensor is its package, a kind of lead-less 8-pin SMD, which is difficult to solder by hand. This, therefore, was the reason for Elektor to design a BoB (Break-out-Board) which has this IC already mounted on the board (refer to the January & February 2015 edition). In this way it becomes very easy to use this IC in a prototype or plug it into a breadboard. When mounting the ChipCap2 sensor on the circuit board, the soldering instructions of the manufacturer have to be strictly adhered to, otherwise the specified accuracy will no longer be obtained. During the first production run of the CC2-BoBs, the sensors apparently got a little too hot when they were soldered, which caused their calibration to drift. According to the application note from Amphenol (see www.ge-mcs.com/download/moisture-humidity/916-127B.pdf) these components may exhibit an offset in the measured values after the soldering process anyway. Some time later — after rehydration — this offset will disappear by itself.

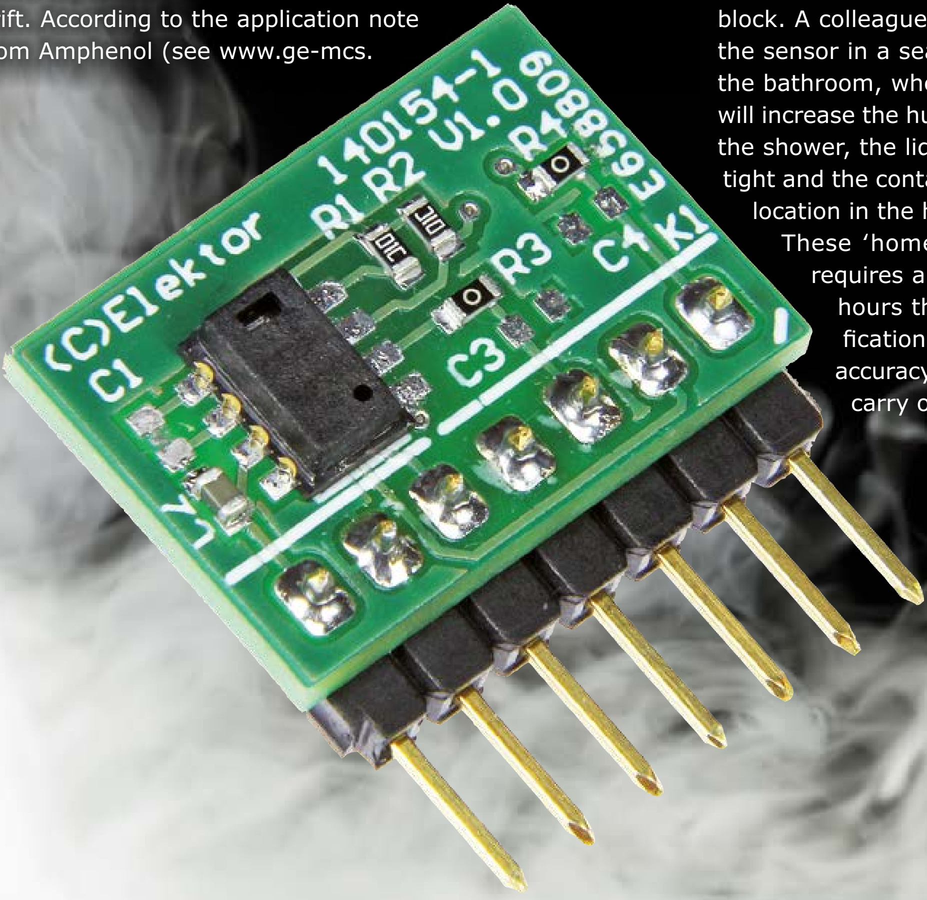
With the sensors that were manufactured in 2015 this automatic restore process — also after having been over-heated — will normally return the sensors to their specified accuracy. But this is not the case with the older sensors that we used for the first batch of circuit boards. The application note describes a separate procedure for this, which has to be followed in order to obtain correct measurements again after the sensors have been too hot. This procedure is as follows:

- Three hours 'baking' at 120 °C
- Store for a period of 60 hours at a temperature of 30 ±5 °C and at a relative humidity of between 75% and 85%

I wanted to test this procedure to verify whether it worked properly. For the first step an ordinary oven can be used, for the second step the high relative humidity is the main stumbling block. A colleague came up with the brilliant idea of placing the sensor in a seal-able (freezer type) storage container in the bathroom, where the lid is initially left off. A hot shower will increase the humidity in the bathroom considerably. After the shower, the lid is placed on the container to make it air-tight and the container is then placed for 60 hours in a warm location in the house.

These 'home remedies' proved to work perfectly. It requires a bit of patience, but after waiting some 63 hours the sensor nicely operates within its specifications again. If you have any doubts about the accuracy of your Elektor CC2-BoB, then you too can carry out this procedure with your little BoB. ◂

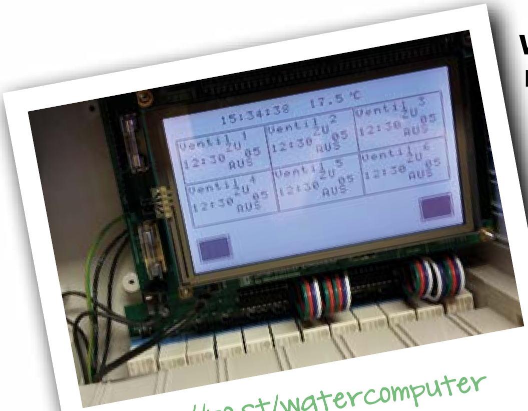
(150546)



Dot Labs...

even solves domestic problems

Often projects are born out of frustration. A good place to get frustrated is in and around the home and not only because of the beloved family members. Here's a selection of projects that attempt to solve some common household problems.



http://po.st/watercomputer

Water Computer

Last summer the author spent so much time watering his garden that there was almost no time left for electronic projects. Perched holding a hosepipe in hand instead of a soldering iron or a probe, the author came up with a fine working solution, which he nicknamed Water Computer. It consists of a 12-stage timer with 12 relay outputs to control the valves.

http://po.st/stopcalling

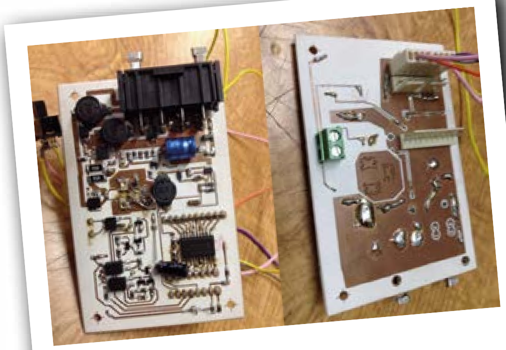
Block Anonymous Phone Calls

Despite a clear decline compared to mobile telephony, in many parts of the world POTS (Plain Old Telephone Service) aka landlines remains the basic form of residential and small business service connection to the telephone network. This anonymous call rejection circuit for POTS lines screens you from callers having blocked their Caller ID information, as telemarketers often do. It analyses the Caller Identifier data (CLIP) sent between the first two rings of incoming phone calls, rejecting any P (Private) or O (Out of area) call types.

Garden Hose Bumper with "point of bump detection"

You normally use bumpers around a robot to protect furniture or people that get in the way. This idea uses a length of 1/2" garden hose (fitted with a single SRF02 ultrasonic sensor) as a bumper around the robot. When the hose is pushed in by as little as 2 to 4 mm it produces reflections of the ultrasonic waves inside the pipe, which allows the point of contact to be determined. The project uses an ATmega8 board with LCD display to show the hose length.

http://po.st/hosebumper



http://po.st/Dali

DALI Power LED Driver

In all sincerity we try to be impartial and not favor one solution to a problem over another. On our LABS website we perceive a standard for digital lighting control. Does the name DALI ring a bell? Digital Addressable Lighting Interface (DALI) defines a communication protocol and physical interface standard for use in lighting control applications, jointly developed and specified by several manufacturers of lighting equipment. Many controllers, from NXP for example, come with applications supporting DALI, but do you have any experience of applications using DALI in the real world? Join us and share your knowledge, your expertise, or interest!

Pye P87BQ Radio Find & Restore

Elektor 3/2015 (May & June), page 122 (150149)

FEEDBACK. Dear Elektor friends, here's a follow-on to this brilliant article. In the 1950s and 60s there was a series of tubes (valves) made for extremely low plate (anode)/screen grid voltages. Filament (heater) voltage: 6.3 V; plate (anode) voltage: 6.3 V or 12.6 V, intended specially for use in automobile radios. The types were ECC86, ECH83 EF97, EF98, EBF83 and ECF83. Employing these made the use of noisy, inefficient chopper vibrators unnecessary. In those days the final amplifier was presumably equipped with transistors.

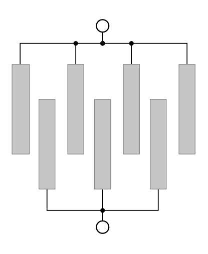
Manfred Ruprechtsberger

Err-lectronics

Corrections, Updates and Feedback to published articles

Platino Solder Station

Elektor 4/2015 (July & August), p. 98 (140107)



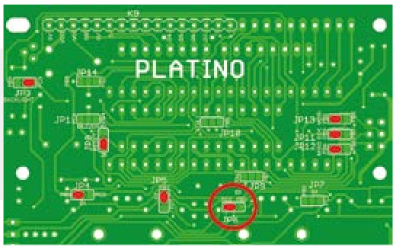
FEEDBACK. The problem — that the program cannot tell, after 15 minutes have passed, whether you're still soldering — is relatively easy to solve. You use conductive silver paint to make two interleaved comb-like shapes on the handle of the soldering iron. These overlays will form an adequately large sensor that your skin resistance will activate while soldering (see drawing). This touch sensor needs to be polled by the controller to enable the Platino to know how long it is since the user last handled the soldering iron. The 15-minute timer is reset every time the sensor is touched. This means you no longer need to press the rotary encoder button every 15 minutes. If you really haven't touched the sensor for 15 minutes, then a quick tap will switch the heat output back on again. The idea is not new and was published in a special edition of the German magazine *ELO* around the end of the 1970s or beginning of the 80s. In those days the multivibrators required were made using discrete transistors. That soldering station has been in use here ever since those days and has always served me faithfully.

K.-L. Buttke

Platino Solder Station

Elektor 4/2015, (July & August), page 98 (140107)

CORRECTION. Jumper JP6 was not mentioned in the article. It needs to be placed in position PC2 (see illustration).



Platino

Elektor October 2011, page 14 (100892)

UPDATE. A new version of the "Platino" multi-function AVR board has been released. The new version v1.4 now offers an additional 3.3-V voltage regulator (besides the 5-V regulator still provided) and simplifies access to the second UART in the 40-pin AVR controller (UART1, at pins PD2 and PD3). Additionally there is now the option to install an extra screw terminal block for power supply purposes. The screen printing for component placement has been improved on both sides of the PCB, in order to avoid accidentally misplaced components. This new board has a different number (150555-1) and the revision count now begins with v1.4, to avoid any confusion with previous versions of the board.

Incidentally, did you know that there is now a Platino expansion board? This goes by the name of Platino Add-On Board. The Platino is also programmable using the Arduino IDE. All you need do is to download and install the Platino Library (120094-11.zip) from our website.

[www.elektor.com/platino-add-on-board-140433-1] *[www.elektormagazine.com/120094]*

Bike Inclinometer

Elektor 5/2015 (September & October), page 80 (140458)

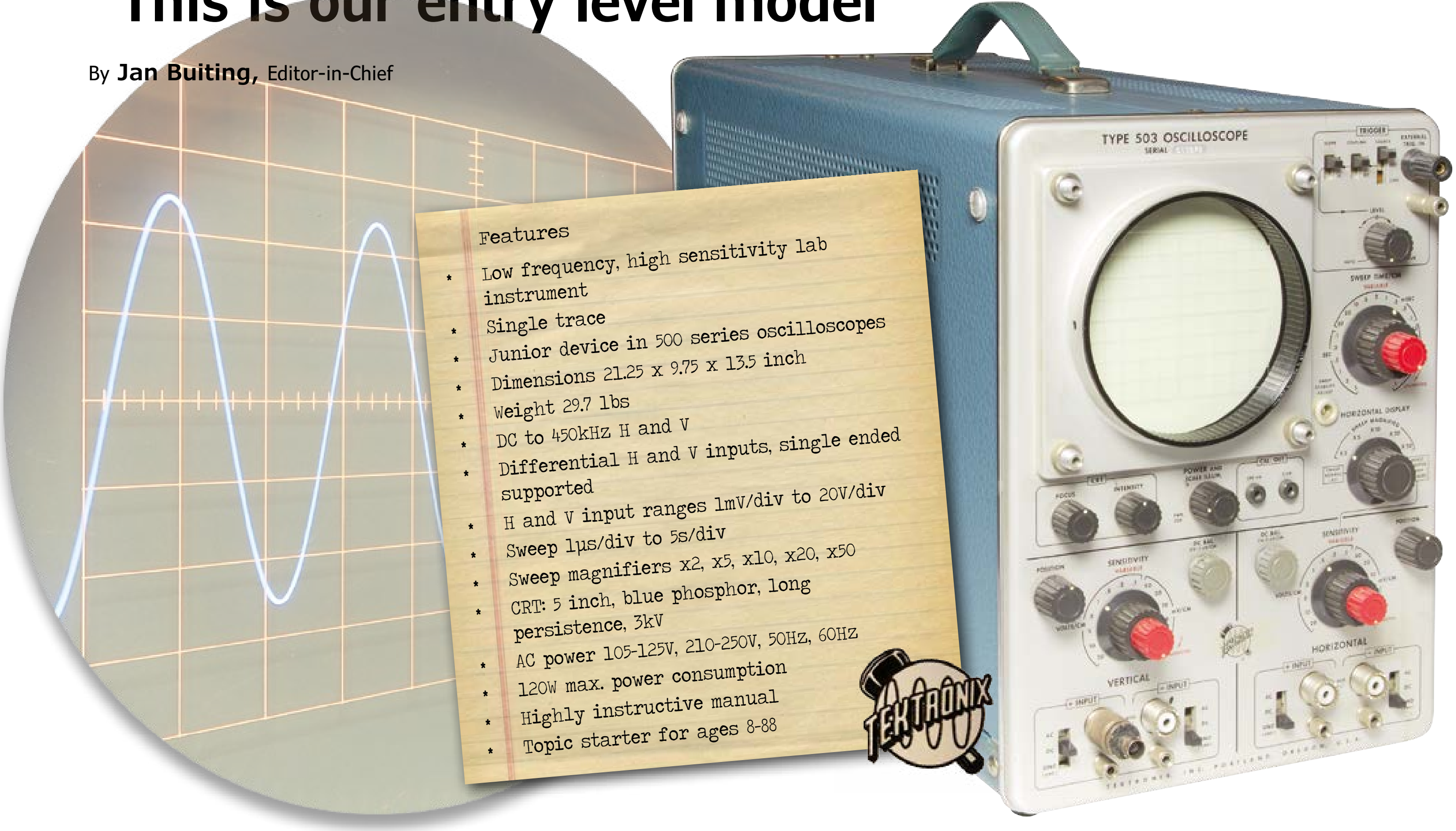
UPDATE. The author has kindly provided the Assembler source code for the project. As always you can download it gratis, it's archive file 140458-11.zip. *[www.elektormagazine.com/140458]* ◀

(150488)

Tektronix Type 503 Oscilloscope

“This is our entry level model”

By **Jan Buiting,** Editor-in-Chief



Features

- Low frequency, high sensitivity lab instrument
- Single trace
- Junior device in 500 series oscilloscopes
- Dimensions 21.25 x 9.75 x 13.5 inch
- Weight 29.7 lbs
- DC to 450kHz H and V
- Differential H and V inputs, single ended supported
- H and V input ranges 1mV/div to 20V/div
- Sweep 1µs/div to 5s/div
- Sweep magnifiers x2, x5, x10, x20, x50
- CRT: 5 inch, blue phosphor, long persistence, 3kV
- AC power 105-125V, 210-250V, 50Hz, 60Hz
- 120W max. power consumption
- Highly instructive manual
- Topic starter for ages 8-88

Delighted with the response I got to an earlier installment on Lazarusing a Tek 564 ‘scope I now tackle an older but lighter one, the 503. Junior and modest as it may look compared to its bigger brothers in the 500 series, the 503 has enough tech content to amuse and educate.

One way to retort non-technical people crying out “what on earth is that” at a piece of vintage electronics in sight is LBSIA: “label basically says it all”, and continue your soldering work. Next you will hear something like “That’s an ermm, it says… T-tektronix Type 503, O-what? Oscilloscope. Oregon. Right! What a monster, ton of controls, they don’t make ‘m like that anymore. Tubes eh?”

Actually the 503 is a *small* oscilloscope, and it’s described as *lightweight* and *an entry-level instrument* in the Tektronix 1960 sales brochure. Aimed at colleges, students and everyone else out of pocket for a ‘big’ 500-series oscilloscope the 503 sold well and was produced till about 1975. Today, it is a gem of early 1960s test instrument technology, with peculiarities and problems worth highlighting and chuckling over, 55 years on.

OMG only 450 kHz

Buying a Tek 503 back in 1964 would set you back a good $650, the equivalent of a car for college or high school commuting. Competition was abundant in this price segment, even from home-made instruments described in the hobby magazines. Sales were not helped by one specification of the 503 that looked poor on the face of it: *Bandwidth DC-450 KHz*. At that time even a homebrew oscilloscope would run rings around the Tek 503 in terms of frequency range. However the 503’s limited bandwidth and a few other unique features put it in application areas that explain everything: biomedical, chemical and elementary physics.

Longwave transmitters like VOA were powerful and plentiful in the USA and the sheer thought of interference on the ‘scope trace due to RF breakthrough was (and is) intolerable with those tiny biomedical signals. Hence inside the 503 a concerted effort is made to reduce the bandwidth to about 450 kHz instead of going for the max and boast megacycles in the sales brochure. Here’s the crux: with a reduction of the bandwidth comes an increase in sensitivity and these biomedical researchers are known to talk microvolts. By the way, Tektronix’ upper frequency limits for scopes as found in the product specs are always conservative. All ten 500-series Tek scopes I have collected so far, easily achieve 20% up on the MHz bandwidth specification — that's 25 years after their last calibration and despite thick dust layers, grime, bashing and utter neglect.

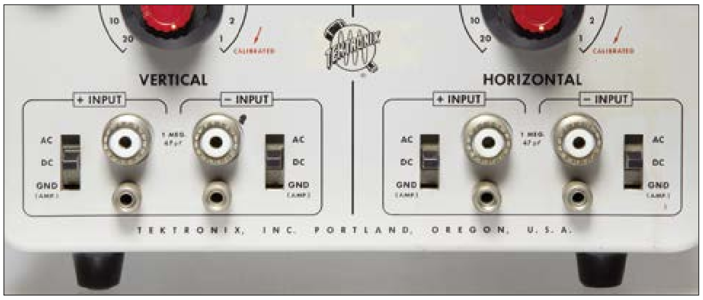


Figure 1. Not too common: an oscilloscope with differential inputs for both Vertical and Horizontal. That's great for X-Y displaying and for measurements not referenced to ground.

Ah, differential

Most of you will think of an oscilloscope as *the* essential instrument for classic electronics design and repair. However in the past the 'scope also saw wide use in physics and research labs where quantities had to be measured by (budding) scientists without a clue about vacuum tubes or capacitors. It was the task of the lab's electronic engineer "to make it visible", so how about: muscle response in horse legs, micro vibrations, brain activity, tiny temperature changes, impact forces, strain, acidity, wind force, opacity, blood pressure, or subsonic noise? Strange at it may sound, in these cases it's actually beneficial to be ignorant about grounding, or potentials above ground. Like when you want to examine the waveform *across* a complex reactive component and not caring if it's at –500 volts w.r.t. the chassis. Instead of having one probe to stick into the circuit judiciously, and the ground lead clipped to the chassis, with a *differential* measurement you work with two 'floating' probes. In short, the 503 is great for all measurements not referenced to ground.

As **Figure 1** shows, differential inputting is not limited to Vertical only; to make the 503 even more versatile its Horizontal inputs are differential also, making the instrument an ideal X-Y monitor for some really groovy Lissajous experiments or curve tracing. Besides X and Y, there's a further 'dimension', Z, which implies intensity modulation via the CRT cathode. Awesome. As a matter of course the H and V inputs can also be configured for single-ended measuring (i.e. the traditional way) helped by a regular swept timebase on H.

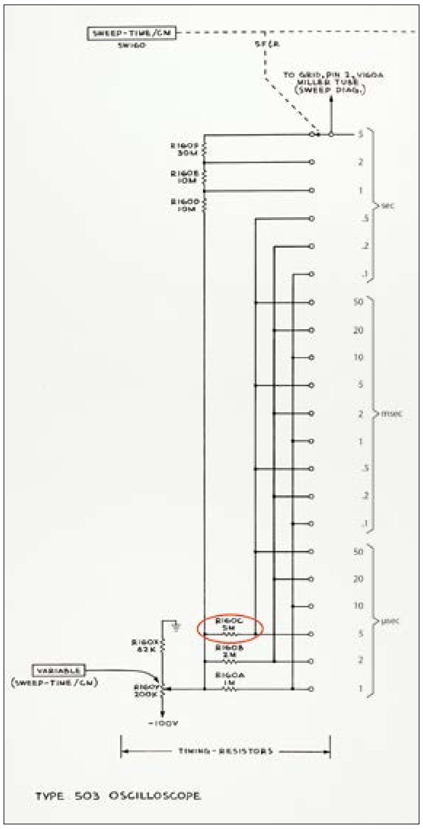


Figure 2. The 503's timing resistors on multi-deck wafer switch SW160.

Yes Sir, 1 mV per division

Most 'scopes feature 5 mV per division as the highest sensitivity setting on the Vertical input. The 503 due to its purposely limited bandwidth and matched pairs of Type 8393 Nuvistors achieves 1 mV/div sensitivity on the V and H inputs and it's surprisingly stable in that range. Closely observing the unbelievably crisp and bright CRT trace, one could resolve 100 µV on the H or V inputs — the order of magnitude of a strong LW transmission on your flying leads or EMI from worn brushes in the neighbor's vacuum cleaner. The common mode rejection ratio on the differential inputs is 100:1 min. on ranges 1 mV to 0.2 V, and 30:1 min. on the other ranges.

Origin, repair work and oddities

This particular 503 came with a 1970s, oversized 202-1 Scope Cart and the original manual, which is rare. The seller kindly arranged for transport of the set to his brother who lives within reasonable distance from my home. This happened at the occasion of a family gathering, some two months after agreeing on the sale for a ridiculously low price. The seller knew *Retronics* and was happy to contribute. I was told the 503 was fine except for a few issues that "surface easily when using the instrument."

From my *503 Lazarus Notebook*, starting October 23, 2015:

Clean outside, some paint chips, Tek logo obscured with tape. Power plug L-N-E resistances ok. Powers up ok on variac, blue phosphor T-5030-P2! No fan! Pleasant smell. Issues: V-VAR noisy, all "5" timebase settings faulty. Poor 5mV, 500mV CAL waveshapes. Inside: untouched, fine dust, Philips tubes mostly, CRT clamp broken, fine wht mold on CRT base cushions. Solder spool found! One mod on 12.6V. Gem!

Suspiciously, the Tek logo on the otherwise pristine front was obscured with very poor quality tape.

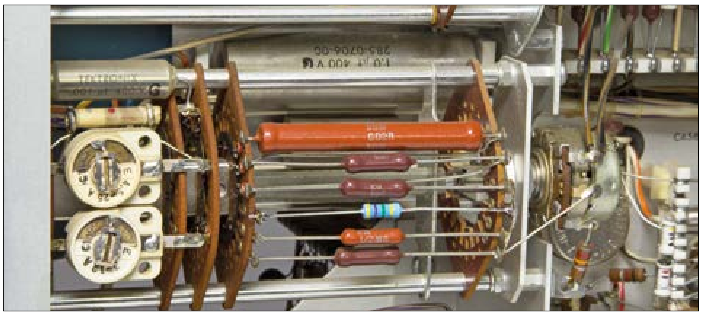


Figure 3. The 5-MΩ 1% tolerance resistor responsible for the "5" positions of the timebase having gone infinite it had to be replaced. A 4.7-MΩ resistor was quickly installed whilst researching the cost of the correct component from Farnell/Newark. I may also decide to fit a 300-kΩ R in series.

The first issue to tackle was the timebase which was erratic or inactive on switch positions with a 5 in it, except the very slowest, 5 seconds/div (!). An easy one. Looking at the schematic of the timebase switch (**Figure 2**), it's definitely R160C at fault. And indeed this resistor measured infinite ohms on my Fluke 79 DMM. I replaced it with a 4.7 MΩ for the while (**Figure 3**), which restored the functionality of the "5" positions immediately. Next up was the noisy V VAR

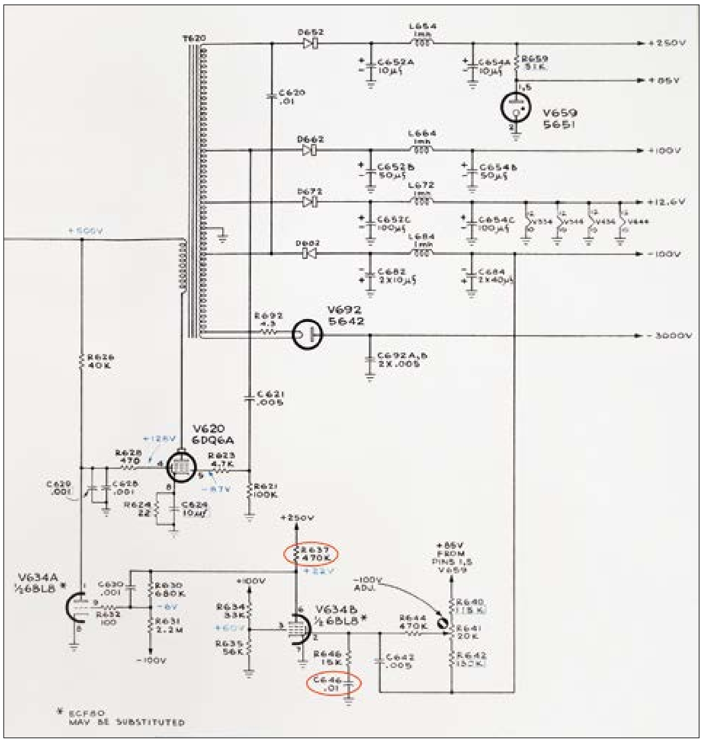


Figure 4. Partial schematic of the 25-KHz switch mode power supply with regulated outputs on all 6 voltages although the control loop appears active on the −100 V rail only. The secret is… T620! The reference voltage is +85 V from gas-filled stabilizer V659. One faulty part was discovered here, it may well have caused the instrument to work for years at supply voltages 20% up from spec.

pot. I use *Tuner 600* spray, a deoxidizer that does not leave deposits hence does not require any further washing agent. Problem solved, time to calibrate the DC Balance on Vertical. All okay.

Because the scope worked fine with no ripple in the trace and just a tad too much sensitivity on Vertical I was unaware of a serious problem hiding in the power supply. In Tektronix scopes of the 1950s and 1960s the PSU invariably supplies multiple voltages which are regulated from a common reference level. In the 503 that's an 85-V gas stabilizer type 5651. From the +85 V, −100 V is derived by a two-tube regulator, see **Figure 4** for the circuit extract. Instead of −100 V I measured a worrying −120 V and a consistent 20% 'up' error on all other output voltages (+299 V, −120 V; +14.5 V… help!). That's thanks to transformer T620 whose closely maintained winding ratios lock the secondary voltages together with amazing precision. Fearing for the life of my four Nuvistor heaters for which I saw "+12.6 V" printed, I quickly inserted a 12-Ω 1-W series resistor for peace of mind. Later when I looked up the 8393 Nuvistor [1] I found a heater voltage spec of "13.5 V ±1.5 V", so sighs of relief.

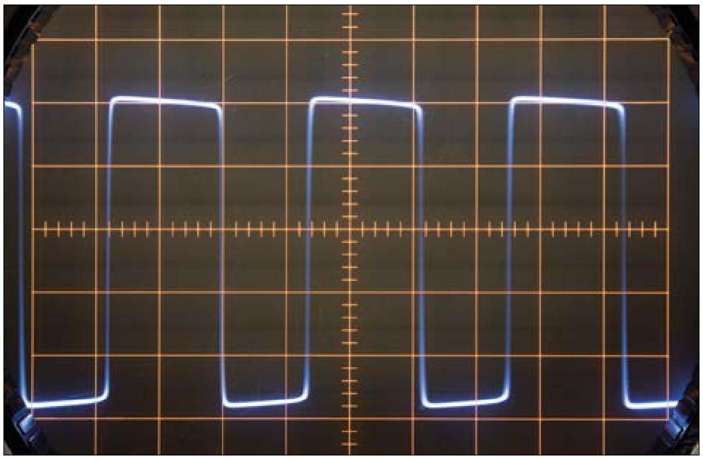


Figure 5. The waveform of the 500 mV test signal as produced by the 503's internal calibrator is wanting in slope steepness, stability, and duty cycle accuracy — Tektronix should have done a better job here.

The central oscillator V620, a big 6DQ6A, operates at 25-30 kHz and basically supplies all HT rails and the Nuvistor heaters. A variation in any of the SMPS output voltages results in V620 counteracting the effect as its screen grid (g2) potential gets adjusted proportionally by V634A as part of a control loop. In my case, no effect was observed from turning the −100 V adjuster, which made me think no control voltage was reaching the input grid (g1) of V634B (the pentode half of an ECF80). This grid stubbornly measured −1.1 volts or so, i.e. way too little to do any regulation. Consequently the anode was at a ridiculous +1.5 V or so, meaning that pentode was conducting hard! Also, V_{g2} was at an unaccounted for +44 V, meaning the tube was pulling substantial screen grid current from p.d. R634/R635.

The moment of glory happened when I unsoldered one leg of C646 and the reference voltage went to −98.5 V instantly. Bingo. C646 measured okay for value but showed intermittent shorting when rapped, a typical effect of older ceramic. With just 15 kΩ in series, a short in C646 pulls the V634B input grid effectively to ground, which is too positive. Unable to find a 10-nF ceramic replacement right away I went for a Philips 'mustard' C widely overrated at 400 VDC (still C646 was specified at 500 V). Can anyone elucidate the function of network R646-C646?

So the secret to the low weight of the 503 is its 25-kHz step-up SMPS wasting far less power than a big transformer humming away at 50 (60) Hz. The down side: failure of one 5-cents component can endanger all others. Not sure if the C646 short represents a worst case scenario but the defective supply sort of leveled out at 'just' +20% up instead of auto-destructing. No harm was done and the C646 repair turned out easy and cheap.

While examining the other parts in the V634 section it turned out that R637 was well on its way to reaching 550 kΩ (!). Apparently it had withstood about 300 V across its legs for an undetermined time, instead of about 225 V. I replaced it with a 470-kΩ 0.5-W resistor from my 1970s parts stock. I also inspected other resistors for drift but none drew suspicion, and no leaky caps were found either. R644 now adjusts to −100.0 V perfectly and I was impressed to see +100.8 V, +250.1 V, +12.8 V, and −2975 V on the other supply outputs, with 25 kHz ripple negligible.

Further minor repairs that were a pleasure to do were undusting the internal .2 V GAIN and COARSE DC BAL VERT. pots using my Tuner 600 spray can. This brought all range adjustments back to life and responding as they should.

Neon show – poor show

When using the 503 to display its own 500-mV and 5-mV calibrator voltages I could not help thinking the rounded and otherwise sloppy waveform was owing to the scope's limited bandwidth (**Figure 5**). But then the waveforms looked even worse on my 50-MHz 547 scope. Looking at the schematic of the calibrator section in the 503 (**Figure 6**) reveals a low cost solution based around two neon tubes. It's a far cry from the standard 10 mV-100 V stepped calibrator circuit found in the bigger 500's and I cannot understand why Tek did not put one extra 6DJ8 tube on the 503 chassis to do the job properly.

The square wave is produced by the turning on and off of B886. This is accomplished by the combined action of B886, B883, and C883. B886 and B883 are neon tubes which normally drop about 60 volts when they are not conducting. However, if they are not conducting, they require about 80 volts across them to start conducting. This principle allows a crude 350-Hz square wave generator to be made that sadly is subject to wide tolerances on the neon tubes with resulting imprecision in terms of output duty factor. Tek are honest about it though, printing: 350 Hz ±50%; Duty Cycle 30-70%. The rise time of the square wave is, ermm, really sad.

I did adjust the calibrator's two output levels to specification and they work but I find the calibrator a poor performer in such a fine instrument.

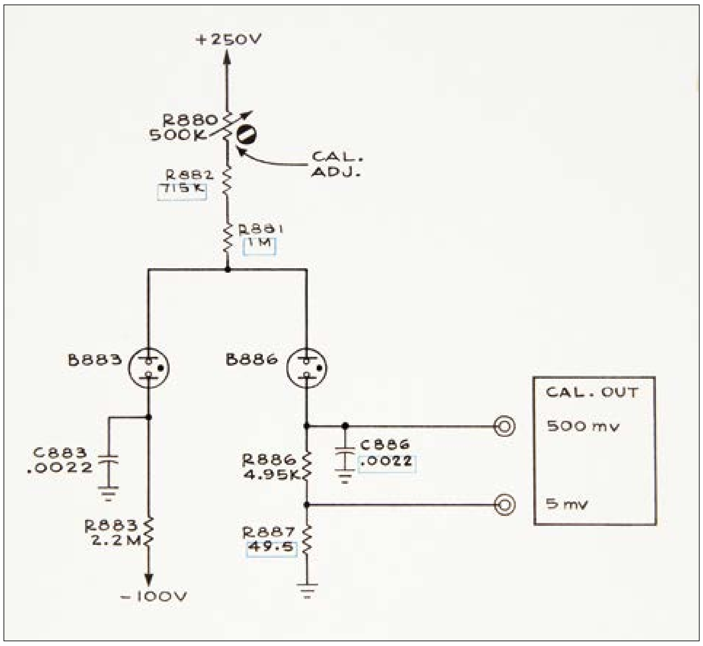


Figure 6. Schematic of the Poor Man's 350-Hz 5-mV and 500-mV out calibrator in the 503 oscilloscope. A 60 cents design, it could have come straight out of *Popular Electronics* magazine!

Underrated, scorned

On a number of forums and web pages dedicated to vintage Tektronix oscilloscopes I saw mentions of the 503 being a poor performer and finicky in terms of its power supply. I've already covered the 450 kHz bandwidth issue that's actually a feature. And the power supply is basically a clever design given the 'lightweight' spec set out for the instrument.

Most light scorning of the 503 is (a) unwittingly in relation to the big ones in the 500 series, (b) not appreciative of the time when the product was launched or (c) in reference to bodged or ill repaired instruments.

The 503 when given some care, and with forgiveness for its ageing components, is a fine instrument. It has exactly the same top class components soldered with silver-bearing solder in ceramic carriers as the big $2K+ 'scopes in the 500 series. Only fewer components, and no bulky power transformer thanks to its switching supply. It's worthy of any electronics student, be it in 1960 or 2015 (with different purposes though), and handles basic audio measurements very well an authentic environment. It's quiet too, not having a fan! The 503 manual is magnificent and no economizing can be seen here whatsoever except there's no little cartoon figure hidden in a schematic, what a pity!

Although they're faster, lighter, etc. I can see none of today's plastic-fantastic 'scopes working 55 years from now, let alone amuse anyone through a *Retronics* story. ⏮

(150573)

Web Link

[1] 8393 Nuvistor: www.tubezone.net/pdf/8393.pdf

ESTD 2004

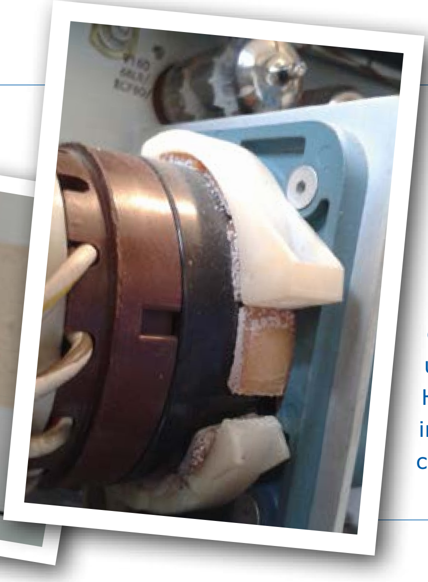
www.elektor.tv

Retronics is a monthly section covering vintage electronics including legendary Elektor designs. Contributions, suggestions and requests are welcome; please telegraph editor@elektor.com

Document it on Arrival

I always use my camera to document odd features of equipment donated to Retronics. I have never seen a piece of vintage equipment without a quirk or a story behind it.

In the case of the 503: white mold on rubber-ish cushion pieces around the CRT base, and cheap tape to obscure the Tek logo. A case of industrial espionage?



Tip: if placed on a newspaper even heavy equipment can be turned and shifted with ease across a veneered flat surface, without causing damage or upsetting the missus. Handy for an initial inspection or light cleaning.

The Electronics Year 2015

What the experts say

By **Thijs Beckers** (Editor-in-Chief, Elektor Netherlands)

An innumerable amount of new developments got created in the past year. Of course at Elektor we try to monitor our huge field of interest and keep an eye on the most promising inventions. And so does everyone else; from farmer to vet, from *lab rat* to sewer inspector, from pilot to kite-flyer. Here we would like to present to you a largely uncensored collection of personal comments from people in the field on *The Electronics Year 2015*.

Fons Janssen

Field Applications Engineer, Maxim Integrated

Nothing really comes to mind. All that happened last year is more of an evolution of existing stuff (so more of the same). RPi 2 isn't a new concept, it's 'just' an upgrade. The same goes for smartphones and tablets, they only get bigger and better screens, more MIPS and more memory. But maybe **IoT** might be something to keep an eye on. We run an IoT competition internally, but I haven't been able to come up with ideas. It isn't very obvious to come up with something truly new. Maybe Apple will present something genius? There are probably some good things happening in the health care arena though, such as **printing skin** using 3D printers and other nanotechnologies. Also **fitness tracking** and **remote patient monitoring** are truly a new thing and can contribute the society.

The VW emissions scandal ("Dieselgate") comes to mind

Electric cars.

Moore's Law will last for a while. Other technologies will enable the growth when physical limitations are reached for the current technology.
AI is an interesting technology, but a far cry from what Hollywood makes us believe.
2050? We'll be struggling with over-population in the world.

Alfred Rosenkränzer

Development Engineer and Group Leader, Advantest

Nothing really comes to mind.

IoT. I don't understand this hype. Most of my colleagues don't understand it either.
Autonomous driving vehicles. They don't solve city-parking problems (I commute by S-Bahn). Fiasco: **VW**.

I have no idea.

Moore's Law will last until we reach the physical boundaries of what can be made.
George Orwell was too early with his predictions (1984), it would take huge efforts to monitor everyone and would be largely useless.
On AI: Nothing against AI and its progress, but I would love to see humanity use its natural intelligence better!
2050? My remains will probably be resting in peace, so it doesn't matter to me anymore.

Robert Lacoste

Manager, Alciom

As I'm mainly working in the RF business I target for my first choice on that topic: 2015 saw a huge devel-

To many people the beginning of the new year is often a time to reflect and look back. It is also a time to start looking ahead possibly guided by a few new year's resolutions. To encourage our connections and start a conversation, I came up with a short list of questions to which the answer could be as short as one word or as long as a 1000 words.
Then I asked a selection of Elektor's relations and regular contributors to give their honest and personal reaction. What follows is an uncensored, integral rendition of the candid replies I received. I'm sure there are some gems in there you certainly didn't see coming! At least I didn't...
A big "Thank you" to everyone who took the time and made an effort to write down his thoughts.

What are the best five technological inventions/ developments/products of 2015?

What is the most worthless technological invention/development/product or biggest fiasco of 2015?

What do you think is the most promising development for 2016?

Open discussion.
Pick one or say what you have to say:

- How long will Moore's Law last?
- How accurate were George Orwell's predictions?
- Artificial intelligence: your opinion?
- 2050: where do we stand?
- Anything else (yes, anything! Even non-technical topics)

opment of dedicated **network infrastructure** for what everybody call the Internet of Things: Sigfox was one of the first, but the LoRa Alliance is running very fast also. These solutions would allow to basically connect anything to the cloud without relying on a home gateway. A huge list of new opportunities will now be possible.
Another area which showed very impressive new products is the high-end **test instruments** arena. A few years ago nobody would have dreamed of a 100 GHz oscilloscope with no less than a 240 Gsps real-time ADC. Now it exists (LabMaster 10-100Zi from Teledyne Lecroy), even if you may need deep pockets to buy one of them...

Well, let's say 90% of the "**connected things**". I read somewhere that the average usage duration of connected watches or other activity trackers is about 3 months from purchase, meaning that after 3 months the product stays in a drawer for years... We definitively have to find actual useful usages of this technology!

Very likely the one that nobody knows about. More seriously I hope that 2016 will see another push forwards in the **3D-printing** technology. May be effective conductive-material printing, and not the awful resistive solutions that exist today?

Moore's Law will last as long as someone would not claim that a 3 GHz dual-core 64-bit processor is overkill to do word processing and basic internet browsing. I hope that Moore's Law will be replaced by another law stating that the energy efficiency for a given task is doubling every year or so.
I guess that Mr. Orwell would have found Facebook, Twitter and the likes very interesting applications. Being able to know in near real time where any of your friends are and what they are doing is quite frightening. And the Internet of Things will make that even easier. And I'm even not talking about Bluetooth or 2G/3G tracking of your smartphone, which is becoming very easy thanks to low cost software-defined radio kits.
Take any AI researcher from the 70s or even 80s and put him in front of an application like Siri. He will without any doubt conclude that AI is here. For me it is no longer a dream, just an available technology.

Clemens Valens

Technical Manager, Elektor Labs

Autonomous vehicles? I have absolutely no idea.

The Arduino split.

Wireless & security.

We are always too optimistic about the speed of innovation/evolution on the short term, but in the long run we are usually too pessimistic. Orwell was too "optimistic" for 1984, but never expected what we have in 2015. Moore's Law:

idem, artificial intelligence: idem. 2050? Things will have changed but it's impossible to predict in what way. Something that we never expected will be big, something that we did expect will be old hat.

Albert Miao

CTO, Seeed

Magic leap; **Windows Surface Book**, the surprise from Microsoft; **Voltera**: Circuit Board Prototyping Machine; **Tesla Model X**;

Qualcomm Snapdragon 810, it has failed big time! **iPad Pro**.

The **communication technology** (like the LTE cat M/0, BLE 4.2, Wi-Fi low energy, etc.) for IoT;

Moore's Law will last less than 10 years.
I can't agree with George Orwell's predictions anymore. Cloud computing and big data will help humanity create a real AI machine, but it will be very dangerous for the human world if we cannot use an AI machine right!
In 2050 every personal device will be integrated into one VR device; No computer, phone, wrist strap...

Jean-Paul Kok

Director, AAN B.V. / StreamLine

LoRa technology, revolution for the years to come. Data communication at long distances without the use of a SIM card.
Technological achievement: Bell test at TU Delft on entanglement of electrons.
Progress in development of autonomous cars.
Life style products — innovative bracelets, providing online feedback on patient status

Microsoft's $7.6 Billion write-off on the Nokia deal.

Development of quantum computers, e.g. the deal between Intel, TNO and TU Delft, Netherlands.

Moore's predictions will come to an end in 2020, where I agree with Bob Colwell.

Harry Baggen

Content Producer, Elektor Labs

Loophole-free **Bell test at TU Delft**: Einstein was wrong.
Small and cheap modular computers, like the **$9 CHIP**.
Better and more affordable **3D printers** (TIKO).
The first commercial **foldable displays** (Samsung).

Not worthless, but overhyped: **smartwatches**.

Virtual 3D-displays, like the Hololens from Microsoft.

Moore's Law will not last longer than a few years; by then we have reached semiconductors on an atomic level.

Alexander Pokorny (Dipl.-Ing.)

Deputy Section Manager Application, GLYN GmbH & Co. KG

Renesas – Synergy platform. Developers today don't like to read thousands of manual pages to know how to use a microcontroller. Developers today no longer have the time to develop a product with USB/Ethernet over several years. The answer is the Synergy family. A combination of MCU and qualified & certified Software.
FTDI – EVE2 – FT8xx. The second generation of the "Embedded Video Engine (EVE)" allows for driving a TFT with 18 to 24-bit color depth and 800 × 600 pixels over a simple SPI, Dual SPI or Quad SPI interface. So an "old" application with a "small" MCU could be very easily expanded with an attractive display. New outfit for old applications.

VW – Motor management scandal. A chance for **electromobility** – but the car manufacturers in Germany have missed this train.
Apple – from a leading brand to a "me too" company.

This is a question of perspective. I think for Europe the subject "Energy" is *the* big thing. So the **Powerwall** from the Tesla could be the big thing for the next year. Homeowners will be able to store their self-generated power from solar cells instead of selling surplus energy at a low price and having to buy it back at a higher price. I expect many different companies will jump on the environmental bandwagon and design products for use with the Powerwall. I hope that we as a distributor benefit from this process, with many new customers and new projects.

At some point everything ends. For many years Moore's Law was correct, but now we are approaching the edge of economic viability. With smaller die structures the costs for design and production processes increase exponentially. So these developments are only cost effective in special market segments, not for the mass market.

Rok Mesar and **Črt Valentinčič**

CEO and CTO, Red Pitaya

All **drone rescue projects** that were made to save people's lives; **Cicret bracelet**; **Smartphone with 3D** that doesn't require 3D glasses; **NDA digital data storage**/ hard disk; **XMOS parallel processing** microcontrollers.

Nothing. We believe Hoffmann car still has the lead ;).

Follow us and stay tuned...

As the transistors are getting smaller and smaller the quantum mechanical behavior increases probability for errors and it seems like we are very close to the limit and we are not able to make chips smaller. In the next years we will probably be tweaking Moore's Law with improved error correction techniques and chips that will grow in 3D, but in parallel we will have to develop new technologies such as molecular transistors.

George Orwell's prediction of big brother sure came true, not in the way of big dictator that watches us through the telescreens, but more as the big data that is collected by companies from all sorts of private and public devices like smartphones, computers, web cameras and others. Collecting all this data can be used in a very good way to help the humanity or it can be exploited, so the question remains the same: "Who will watch the watchmen?"

Real intelligence consists of the abilities to learn from experience, adapt to new situations, understand and handle abstract concepts and use knowledge to manipulate one's environment. Nowadays devices are far from being intelligent even if the marketing guys are using the word *smart* together with every new product. Our smartphones or any other "smart" device in reality are not smart at all. In fact, they are more or less only able to execute some mathematically complex tasks, completely relying on very simple integrated algorithms and our own decisions. Most of their output is totally predictable and they cannot make any complex decisions at all. In the future this is sure about to change, since storage and processing capabilities will become more and more available and devices will be able to collect more and more input data and analyze it to gain some experience and knowledge from it.

In the future we will be surrounded and equipped with super smart products that will hopefully make our lives safer and even more comfortable. If YouTube still exists, we will surely be laughing at artificial intelligent machines that will be doing stupid things and cats that will be playing with them, but in the end we will still find our friends as the most valuable intelligence that ever existed, and as expected we will most probably be enjoying the same life qualities that already existed in 2015.

Neil Gruending

Software Developer, WS Technologies

DesignSpark Electrical: I've always had to use the wrong CAD tools for wiring diagrams in the past but they never worked very well. DesignSpark Electrical changes that and the price point can't be beat.

I think that **security and trust** in products has taken a big hit this year, especially in the automotive industry. There were several significant security vulnerabilities discovered this year that could affect a car's safety. And of then there was the **Volkswagen** diesel debacle. I think high profile issues like this reiterate that some engineering decisions can have a lot of consequences.

We already have lots of processor cores but I think that **power management ICs** (PMICs) like the Cypress S6AE10xA will be the next step in maximizing power efficiency. I still think an analog chip is one of the best ways to decide when to power up a processor, especially when their quiescent current is a couple hundred nanoamps.

I don't think that Moore's law is dead, I just don't think it matters as much anymore. I think that technology is moving towards overall efficiency over raw computational power. Designers are now having to maximize the amount of work that can be done per watt and it's leading to some really interesting multicore processors. Makes me wonder what Intel has in mind with its Altera acquisition.

Burkhard Kainka

Author, Elektor, Franzis Verlag GmbH

PLL-Chip SI5351, Raspberry Pi 2, latest AVR 32 Bit ARM-derivatives; For me privately: newly ‚discovered software to use: **Python, Processing**.

Deutsche Telekom ceases ISDN; Voice-over-IP is significantly less reliable.

Internet of Things with small autonomous microcontrollers.

Artificial intelligence is widely overrated. Machines can only handle large amount of data. Real intelligence is only found in humans.

Michele Scarlatella

Marketing Communication, STMicroelectronics

Large deployment of **wearables for health and fitness**; **Drones** moving into mass market; **Self-driving cars** and car-to-car and car-to-infrastructure communication; **Affordable development tools** that make programming and development easier for makers and hobbyists; Security and infrastructure for **mobile payments**.

Difficult to say.

Starting deployment of **5G**

Robotics and AI may take over in many daily tasks and free us to work on “higher level” jobs. To skeptics we may argument that people were reluctant also when automobiles and washing machines were introduced...
The challenges we have to solve: energy and natural resources for materials and food. If we don’t come up with a solution, tension will grow, creating social and political turmoil, of which consequences can be very dire and destroy many benefits of our technology advancements

Michel Künemann

Freelance Electronic Designer

Elon Musk’s **PowerWall**: 10 KWh for $3500 with a 10 year guarantee.

Politicians when they try to deal with sustainable development.

Increasing role of electricity in transportations

Teach your children to dominate technology. Otherwise they will become technology’s slaves.

Thomas Scherer

Freelance Editor Elektor, Psychologist

7 nm Chips (Moore’s Law is not dead), gradual improvements in **batteries and solar cells**, **LED lamps** are as cheap as CFL lamps, **4K TV**/**5K monitors** (iMac)/ Retina displays everywhere, **self-driving cars**.

ITER progress is very slow; Chip speeds reach a limit; Google Glass; iWatch.

Self-driving cars, 3D-stacking of chips.

Moore’s Law will be at an end with 5-nm structures in about 2018. This will be followed by massive parallel computing (chips with thousands of cores).
George Orwell’s predictions were unbelievable precise for an exponential growth function.
AI will come and will change the whole world and society, starting about 2025. Biggest change in history. But many silly theories are doing their rounds.
In 2050 I will most probably be dead (I’m really sorry ;)) Maybe it will not be much of a joy to live in 35 years, but predictions on these heavy changes are nearly impossible. At least keyboards and mice are dead too.

Henrik Flodell

Sr. Product Marketing Mgr / Development Tools, Atmel Corporation

Amazon Echo. Being able to tell your connected devices what you want them to do is so much easier than poking around on your phone trying to find the right app for that particular lightbulb.

The **Overdrive app**. Technically not from 2015, but I just discovered it. It lets you check out eBooks and audiobooks from your local library on your phone. It has made my commute so much more enjoyable.

If I can shamelessly plug an Atmel product, the **SAMW25** WiFi module. It combines our SAMD21 MCU with our WINC1500 network controller in a module form-factor. Perfect for small devices and very easy to get started with.

This gets nominated even though the product is not yet available, and even though it got booted off Kickstarter: **Skarp**. Why? Because lasers...

The **Hololens** from Microsoft.

Also, honorable mention to this one: **Delorean Hovercraft**. It was tearing up the San Francisco Bay on Back to the Future Day, making everyone that saw it cheer.

The one that comes to mind is **Coin**. It sounded cool, but got severely delayed, and when it appeared there were a lot of issues with the card not working at certain merchants. And it almost immediately got obsoleted by the US switching POS terminals to EMV enabled ones.

Smart connected devices. They are here already, but there will be even more of them in 2016, and they are actually starting to justify the "smart" label. We will see a significant growth in the **IoT** market in 2016 for certain.

Make sure you visit at least one Maker Faire during 2016, especially if you have never been to one before. The creativity on display is mind blowing, and there is sure to be something of interest for just about anyone. Also, while you are there, stop by the Atmel booth and say hi!

Herman Tuininga

Owner, Salland Electronics bv;
Chairman, Industrial Electronics at FHI association.

A lot of technology is evolving but not really new. For example: For a long time the subject of Gartner's hype cycle has been the **Internet of Things**, which is getting more and more attention in 2015. What the driving force behind IoT really is, is a worldwide infrastructure called The Internet and the electronic building blocks. Due to Moore's law electronic SoCs are becoming more and more powerful, cheaper and power efficient, see Bluetooth LE. As predicted by Gartner an Internet connection in 2020 will cost about $1. Another example is the fast increasing **3D printing** market, which will have a great impact in our society including robotic development.

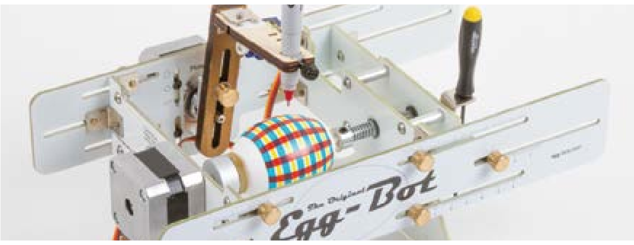
Smart (IoT) objects are initiated and exploding on Kickstarter, Indiegogo etc. **Wearables**, Internet connectivity integrated in all kind of stuff, are we really waiting for a fork with connectivity (HAPIfork)? Or a multi-functional beach cooling box (raised more than 13 million), or the Egg-Minder? Gadgets overwhelm the market. How does this reflect on sustainability and society needs and relevance?

Besides the fact that everyone is looking for the next big thing, a buzz in SC is: What will the potential impact of a disruptive AI in the near future be? Will technology (robotics, artificial intelligence etc.) develop in such a way that we cannot control it anymore and will it become a malevolent singularity? Would it end the world as we know it? I would like respond to this like Wim Hendriksen did at the DevLab conference January 2015, celebrating our 10-year anniversary: "Deal with it!" ⏮

(150522)

Compiled by **Aniek Reuling**

Not just for Egg-sperts



The EggBot is a blockbuster! Search for it online and you'll find some amazing works of art produced with it. And not just eggs; the EggBot is able to draw on most spherical or egg-shaped objects with a diameter ranging from 3 to 10 centimeters. It's fun for the whole family. Not only can you personalize Easter eggs or golf balls, this robot is also a great introduction to Computer Numerical Control and do-it-yourself robotics for adults and youngsters ages 10 and up.

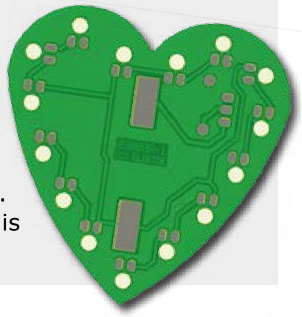
Maker Faire Berlin

After Taipei and New York, the Maker Faire landed in Berlin and of course Elektor couldn't miss out. Our staff was lucky, because at the same time the city was celebrating 25 years of German Unity during the "Tag der Deutschen Einheit". This of course caused a very late Saturday night. The Elektor booth was visited by our friends from Red Pitaya and the soldering course for youngsters drew quite a crowd. It was good to see that increasing numbers of MAKE projects contain electronics. So MAKE(r) community: here we come!

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 Holidays are over, a new year has kicked in and inevitably Valentine's Day is just around the corner. Bored with the usual roses and chocolates? Elektor has the answer! In 2004, **'Electronic Valentine Heart'** was a Mini Project designed by our youngest author ever aged 14 at that time. This first (and only) Elektor Valentine circuit consisted of 15 LEDs inserted from the solder side of a single-sided circuit board leaving the front free for painting. Whose heart wouldn't melt for this electronic one?



NEW IN EVERY EDITION
Elektor-Ethics

PEOPLE NEWS • Various Elektor staffers including Client Manager Margriet Debeij and Elektor Germany German Headquarters in Aachen • Jason Long is among the new freelance designers for Elektor Labs + Business Magazine and all related media • Elektor Labs' Jan Visser visited the famous Dorkbot hacker

Ethical Robots and Robot Ethics

elektorethics by Tessel Renzenbrink

As the field of robotics progresses, robots increasingly share the public and private space with humans. Think of self-driving cars on the roads, carebots in elderly homes and Rumba and its next iterations taking care of household chores. When robots co-exist with humans they need to conform to social norms, first and foremost those norms relating to safety. When you visit someone in their home, you don't take the shortest route to the living room if this means stepping on their pet. And neither should a robot.

Professor Alan Winfield builds ethical robots to enhance robot safety. He is a Professor of electronic engineering at the University of the West of England. Together with his colleagues Christian Blum and Wenguo Liu he built a robot that will prevent a human from coming to harm even if that compromises its own safety. They tested it in a scenario where a human risks falling into a hole deep enough to cause serious harm.

The engineers set up an experimental environment that looks like a miniature football field. Placed at the center of the field is an e-puck robot[1] (an open hardware robot developed for educational purposes) which is given the mission to reach a destination near the end of the field. In between it and its goal is a virtual hole it must avoid so as not to come to virtual harm. In all instances of the experiment the robot navigates successfully around the hole and reaches its destination.
Then a second e-puck robot is introduced into the field. It is marked with an H to highlight it plays the role of the human. The proxy-human is unaware of the hole and moves straight toward it. The first robot (marked with an A for Asimov) abandons its mission and alters its course to collide with the proxy-human to divert it from the hole, despite the fact that this trajectory increases its own risk of coming to harm.

Continued overleaf...

Editor-in-Chief Jens Nickel will be moving to our new

Jan Buiting has been appointed Editor-in-Chief of Elektor

evening at RWTH Aachen

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: **Menno van der Veen, MSc.**

Age: **66**

Education: **Engineering Physics**

Publications: **Trans Tube Amplifiers, Modern Valve Electronics (sequel), and more**

Training on: **Tube amplifier designs, toroidal transformers**

Who is Menno van der Veen?
After graduation, I worked as a teacher. When I was 40, I launched my own engineering bureau, focusing on tube amplifiers and toroidals. I have agreements with several factories producing and selling my transformers all over the world.

Who is your biggest role model in electronics?
Too many to count. I get my inspiration from all around me, but there are two names in particular I'd like to mention. Two fellow Dutchmen, Professor Hendrik de Waard and S.J. Hellings, MSc. still inspire me today.

What will be the most key electronics development?
I'm sure this will be something to do with computers. Hopefully batteries will improve a lot. Also, I hope we won't forget to really listen. I already see the development of cordless smart speakers, which is amazing! But audiophile listening is being cast aside and that is exactly where my interests lie.

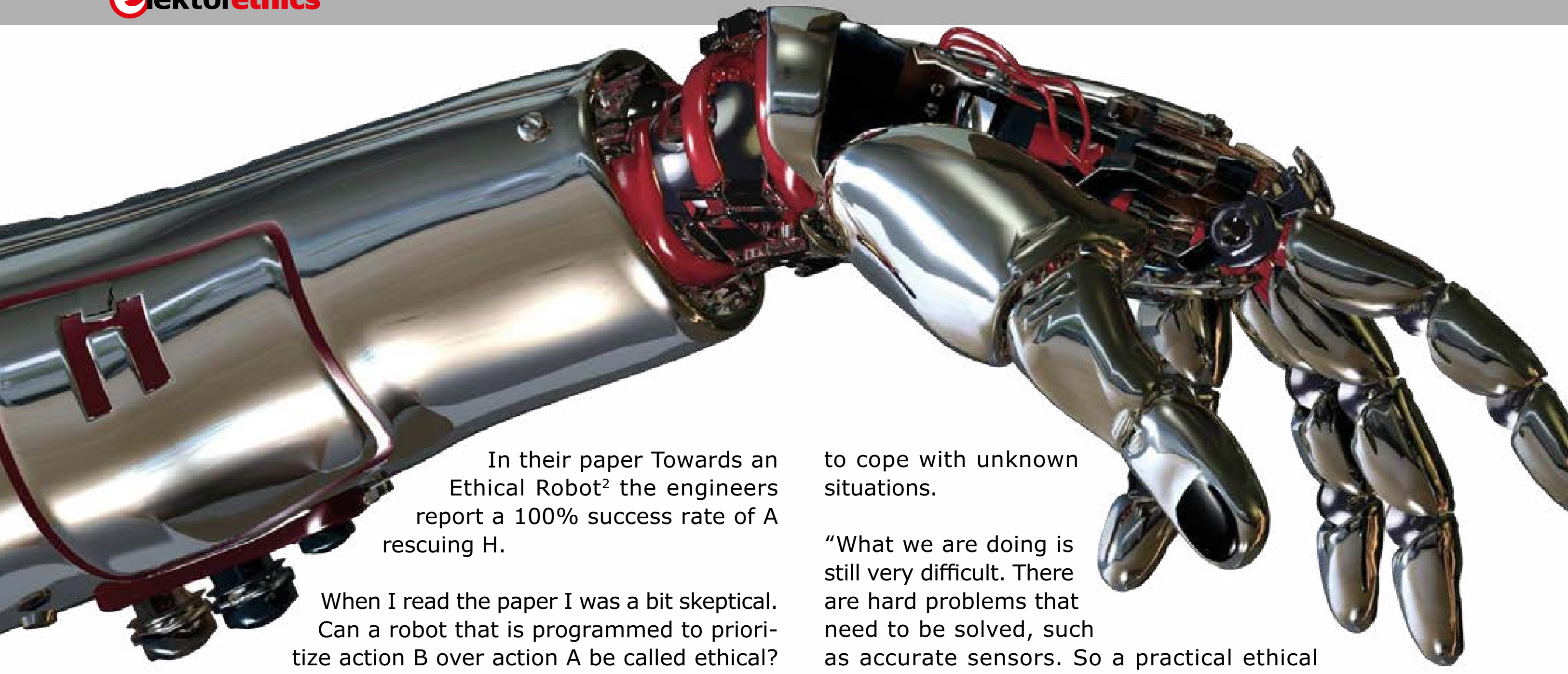
What topics will you be writing about in the future?
I've written four books, starting with theoretical concepts and scientific studies. My focus is now on moving towards smart techniques, as you can read in my fourth book. This area is very promising, I expect that in my next book I will go deeper into this. It's like going on an adventure. Every day brings new insights and I like to share these with the world.

What makes the Netherlands unique in terms of electronic innovations?
That's difficult to answer. Everywhere I look I see new ideas and possibilities. To me, the Netherlands aren't much different from, say, the US in this respect.

Suppose you get €500 to buy stuff in the Elektor Store: what's it going to be? Why?
I'd buy a new PicoScope and books, especially books. There are a lot of brainiacs out there who can teach us a lot.

What's the project you are most proud of? Why?
There are three things I'm really happy with: my toroidals, my Auto-Bias circuit for tubes, and my newest Trans Amplifiers. These three give me the feeling that I too managed to make a contribution to the electronics community. ⏮

(150574)



In their paper Towards an Ethical Robot[2] the engineers report a 100% success rate of A rescuing H.

When I read the paper I was a bit skeptical. Can a robot that is programmed to prioritize action B over action A be called ethical? Moreover, is the experiment scalable? In the setup the robot finds itself in one situation to which it can respond with a limited set of possible actions. But what if the robot is placed in the unstructured highly diverse environment that is the human world? It can encounter an endless number of different situations and it's impossible to hard code a response to them all.

Ethical rule

When I met with Prof. Winfield in a cafe in London to speak about his work and I expressed my skepticism to him, he politely disagreed with me: "What we are not hard coding are all the thousands of situations a robot can find itself in, the only thing we're hard coding is its choice of how to behave given several alternatives. The big advance of the experiment and the particular architecture is that we put a simulation of the robot and the world inside the robot."

The simulation enables the robot to predict the consequences of its own actions without committing to them. In this case Robot A senses Robot H is headed toward the hole. When it runs a sequence of what-if scenario's it predicts that if it moves ahead right it will block H's path, if it moves ahead left it will reach it destination goal.

The mechanism to predict consequences is called a Consequence Engine. The next step is to attribute weight to the possible actions. Here an 'ethical rule' is introduced: saving the proxy-human always takes precedence over all other possible actions, even if it compromises the robot's own well-being.

"What it means", continued Prof. Winfield, "is that, providing the robot can sense the world sufficiently well, and can then initialize – in other words reflect - what it sees in its internal model accurately, than it has the ability to simulate a number of next possible actions. The ethical rule simply chooses one of those actions on the basis of the simulated future consequences. Because the simulation has been initialized with the world as it is at this particular moment, the robot should be able to cope with unknown situations.

"What we are doing is still very difficult. There are hard problems that need to be solved, such as accurate sensors. So a practical ethical robot, even the way we are making an ethical robot that can deal with the real world, is still a long way ahead. It will most likely require all sorts of advances in sensing as well extensions to the basic architecture we've developed."

To my question whether the robot can be called ethical even though it makes no moral judgments of its own Prof. Winfield responded: "Being programmed to behave ethically does not mean you are not ethical. The difference between you and I and our simple robot is that you and I can choose to behave ethically or not. And that's a responsibility of being adult humans. But our simple robots can not choose - they are hard-wired to behave in the way they do."

Ethical choice

The issue of moral judgment also came up in the experiment when Prof. Winfield and colleagues introduced a second proxy-human (H2) onto the football field. H and H2 are both headed towards the hole presenting robot A with the dilemma of which one to save. The experiment ran multiple times and in some cases A saved H, in others H2 and sometimes even both. However, in several instances A was unable to decide and kept going back and forth between the two proxy-humans, eventually saving neither. In their paper Prof. Winfield and his colleagues write: "We could introduce a rule, or heuristic, that allows A to choose H or H2, but deliberately chose not to on the grounds that such a rule should be determined on ethical rather than engineering grounds. If ethical robots prove to be a practical proposition their design and validation will need to be a collaborative effort of roboticist and ethicist."

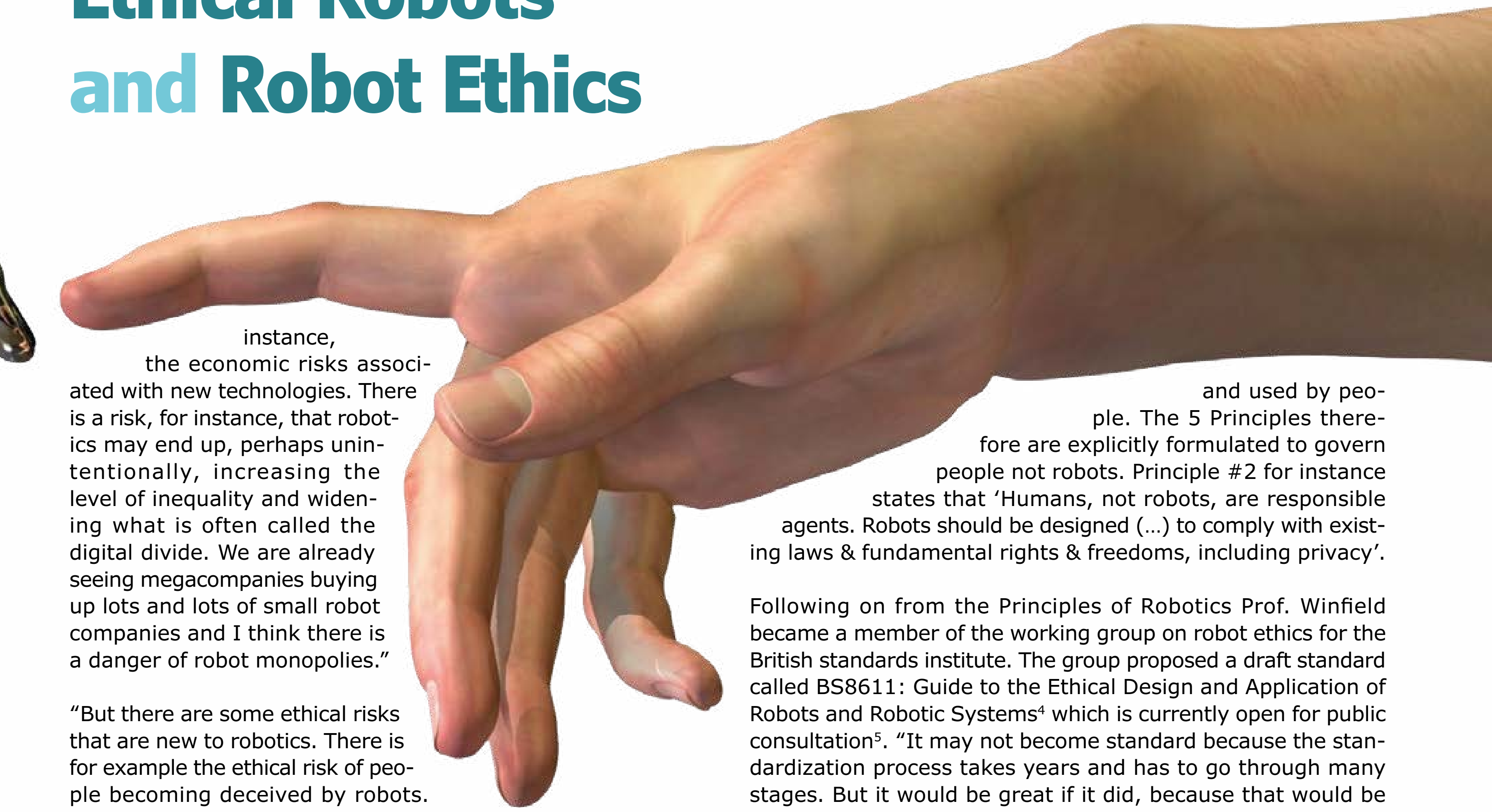
Robot ethics

Before practically implementing ethics in robots, Prof. Winfield was already involved in robot ethics: reflecting on how to develop the field of robotics in such a way it is beneficial to society. When asked why we need robot ethics he said: "Robots are no different to any other technology in that they present ethical hazards. Some of the ethical hazards associated with robots are the same as the ethical hazards associated with any technology. For

[1] The engineers have since replaced the e-puck robot with the more versatile Nao robot.

[2] http://link.springer.com/chapter/10.1007/978-3-319-10401-0_8.
You can download a final draft version of the paper here: https://drive.google.com/file/d/0BwjY2P_eeOeiZFMtc0dja3pJd28/edit?pli=1

Ethical Robots and Robot Ethics



instance, the economic risks associated with new technologies. There is a risk, for instance, that robotics may end up, perhaps unintentionally, increasing the level of inequality and widening what is often called the digital divide. We are already seeing megacompanies buying up lots and lots of small robot companies and I think there is a danger of robot monopolies."

"But there are some ethical risks that are new to robotics. There is for example the ethical risk of people becoming deceived by robots. Imagine a robot pet that behaves in a way that suggests the robot cares for you. In the extreme this could lead to the Tamagotchi effect: robots that care for you and you end up having to care for them. You can imagine an unscrupulous manufacturer effectively asking you to pay extra money otherwise your robot might 'die'. You and I would probably say 'Oh your kidding, I am not paying you extra money, it's a robot'. But unfortunately there are vulnerable people who might be taken in by that kind of scam. That's an ethical risk, associated with robots. Maybe a bigger risk for robots than washing machines or motor cars."

"So yes, there need to be robot ethics because there are potential ethical risks, a whole range of them. From the personal risks – the risks to the user of the robot – all the way through to societal risks, economic risks and even wider environmental risks. A full treatment of robot ethics needs to cover that whole spectrum."

Standardization

In 2010 Prof. Winfield co-organized a workshop to discuss robot ethics under the banner of the Engineering and Physical Science Research Council, the agency that provides most of the funding for robotics in the UK. The workshop resulted in the publication of 5 Principles of Robotics[3]. The workshop delegates noted that science fiction and the media have created an image of robots as some type of independent species that can either help or harm people. However, robots are a technology build and used by people. The 5 Principles therefore are explicitly formulated to govern people not robots. Principle #2 for instance states that 'Humans, not robots, are responsible agents. Robots should be designed (…) to comply with existing laws & fundamental rights & freedoms, including privacy'.

Following on from the Principles of Robotics Prof. Winfield became a member of the working group on robot ethics for the British standards institute. The group proposed a draft standard called BS8611: Guide to the Ethical Design and Application of Robots and Robotic Systems[4] which is currently open for public consultation[5]. "It may not become standard because the standardization process takes years and has to go through many stages. But it would be great if it did, because that would be the world's first standard on robot ethics", said Prof. Winfield. "I can see ethics as the beginning of the process which lead into standards. Standards are really important because they make products safer. Standards allow people to have confidence, and hence trust in products."

Joint responsibility

"Robot ethics is not owned and should not be owned by exclusively roboticists", said Prof. Winfield. "There needs to be philosophers, lawyers, a whole range of people involved. For a set of ethical principles to be accepted, for them to gain traction, they have to be robust, credible, meaningful, to people who are not engineers. In fact, especially to people who are not engineers."

I asked Prof. Winfield what electronic engineers like himself can contribute to the ethics of robotics. "When I was young I did not care. I was just utterly focused and fascinated by the technology and nothing else. As I got older I gradually became more aware of the societal impact of electronics. Electronics is just like science, it is not ethically neutral. When you design a piece of electronics you're not working in an ethical vacuum. I would urge electronic engineers to take an interest in the wider ethical and societal implications of the very circuits that they are designing."

[3] https://www.epsrc.ac.uk/research/ourportfolio/themes/engineering/activities/principlesofrobotics/

[4] https://standardsdevelopment.bsigroup.com/Home/Project/201500218

[5] http://drafts.bsigroup.com/Home/Details/55513

Hexadoku The Original Elektorized Sudoku

It's a new year, a fresh start and so on and so forth... so no more feeble excuses to delay your active participation in our "hexadecimalized-Sudoku" puzzle challenge. 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 February 1, 2015, 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 6/2015 (November & December) is: **A2FBE**.
The €50 / £40 / $70 book vouchers have been awarded to: Dirk Dreessen (Belgium), Michalis Fostiropoulos (Greece) and Alex Lo Furno (Italy.

Congratulations everyone!

		7	C									5	2		
2				0							F				1
1	5					F	9	2	C					8	0
F	3	6				C			0				9	7	A
9	A	8	3		0		6	F		C		2	1	4	5
4	2	D	B	3	C					A	5	7	F	0	8
5	6	E	1	8	A					4	7	C	B	D	3
7	C	F	0	D	4	5			2	8	3	9	6	A	E
B	D	A	9	5	8	E			F	2	1	0	7	6	C
0	E	1	6	7	9					5	4	A	3	F	D
8	F	3	2	6	D					E	9	4	5	1	B
C	4	5	7		F		3	B		D		8	E	2	9
D	1	C				3			4				A	9	2
E	7					D	0	A	1					3	6
3				C							B				4
		2	A									D	0		

C	F	4	6	D	E	0	A	3	9	1	7	2	B	5	8
9	1	0	7	F	2	B	C	D	5	4	8	A	E	6	3
2	8	3	D	9	1	5	7	6	A	B	E	4	C	0	F
5	A	B	E	3	8	4	6	C	F	0	2	1	7	D	9
F	5	C	2	1	0	6	3	8	B	9	4	7	D	A	E
0	3	D	8	A	C	7	9	5	1	E	6	F	2	4	B
6	4	7	A	2	B	8	E	F	0	3	D	C	1	9	5
B	9	E	1	4	F	D	5	A	7	2	C	6	3	8	0
A	2	F	B	E	4	9	8	0	3	C	1	D	5	7	6
1	C	9	3	5	6	F	0	E	D	7	B	8	A	2	4
7	D	5	0	B	A	1	2	4	6	8	F	3	9	E	C
E	6	8	4	C	7	3	D	9	2	5	A	B	0	F	1
3	B	2	5	6	9	A	F	1	4	D	0	E	8	C	7
8	E	6	F	0	3	2	B	7	C	A	9	5	4	1	D
D	0	1	C	7	5	E	4	2	8	6	3	9	F	B	A
4	7	A	9	8	D	C	1	B	E	F	5	0	6	3	2

The competition is not open to employees of Elektor International Media, its subsidiaries, licensees and/or associated publishing houses.

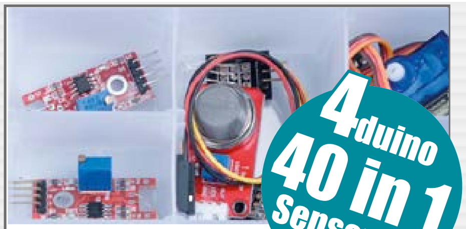
Professional Quality @ Discount Prices!

reichelt.co.uk elektronik

- ✓ More than 45 years of experience
- ✓ 24-hour shipping
- ✓ More than 50,000 products

onlineshop languages:

Payment Methods: SOFORT BANKING, VISA, amazon payments, PayPal



4duino 40 in 1 Sensor kit

This sensor kit gives you practically everything you need for your next Arduino project.

- Joystick
- Sound
- Relays
- Touch
- RGB
- Shock sensor
- Switch
- IR transmitter / receiver
- Temperature sensor
- Humidity sensor
- and much more

38,51 £

ARD SENSOR KIT-2

Ultrasonic distance sensor

- Range: 3 – 400 cm
- Tolerance: 0.3 mm
- 5 V DC

5,01 £

DEBO SEN ULTRA

SUBSCRIBE NOW!

Newsletter

Receive weekly fresh information about

- ✓ product innovations
- ✓ specials
- ✓ Price reductions

Arduino Uno Rev. 3

ARDUINO

- ATmega 328
- 14 digital I/O interfaces (6 as PWM output)
- 6 analog inputs
- USB connection
- ICSP header

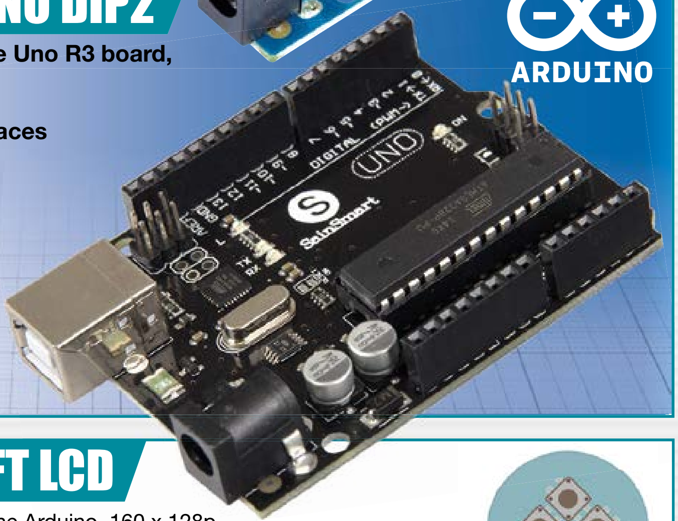
ARDUINO UNO

14,92 £

ARDUINO UNO DIP2

ARDUINO

- Arduino compatible Uno R3 board, DIP version
- ATmega 328
- 14 digital I/O interfaces (6 as PWM output)
- 6 analog inputs
- USB connection
- ICSP header



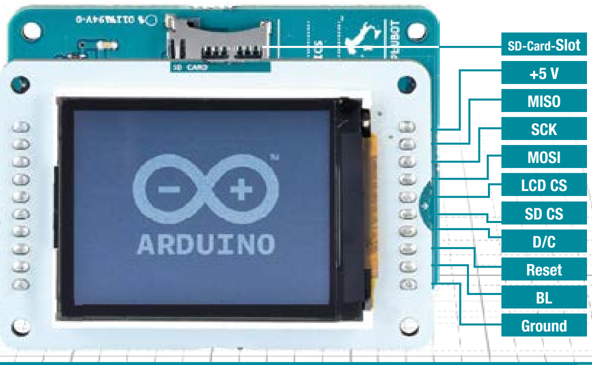
ARDUINO UNO DIP2

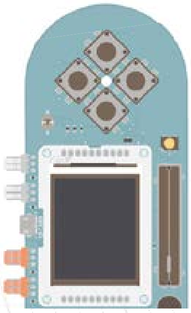
12,68 £

ARDUINO TFT LCD

4.5 cm (1.77") LCD for the Arduino, 160 x 128p

- 18-bit backlit TFT LCD display
- MicroSD card slot on back





However, it can also be used with other AVR-based Arduinos™ (UNO, Leonardo, etc.).

ARDUINO TFT LCD

18,62 £

Order now! **www.reichelt.co.uk**

Order Hotline: **+49 (0)4422 955-360**

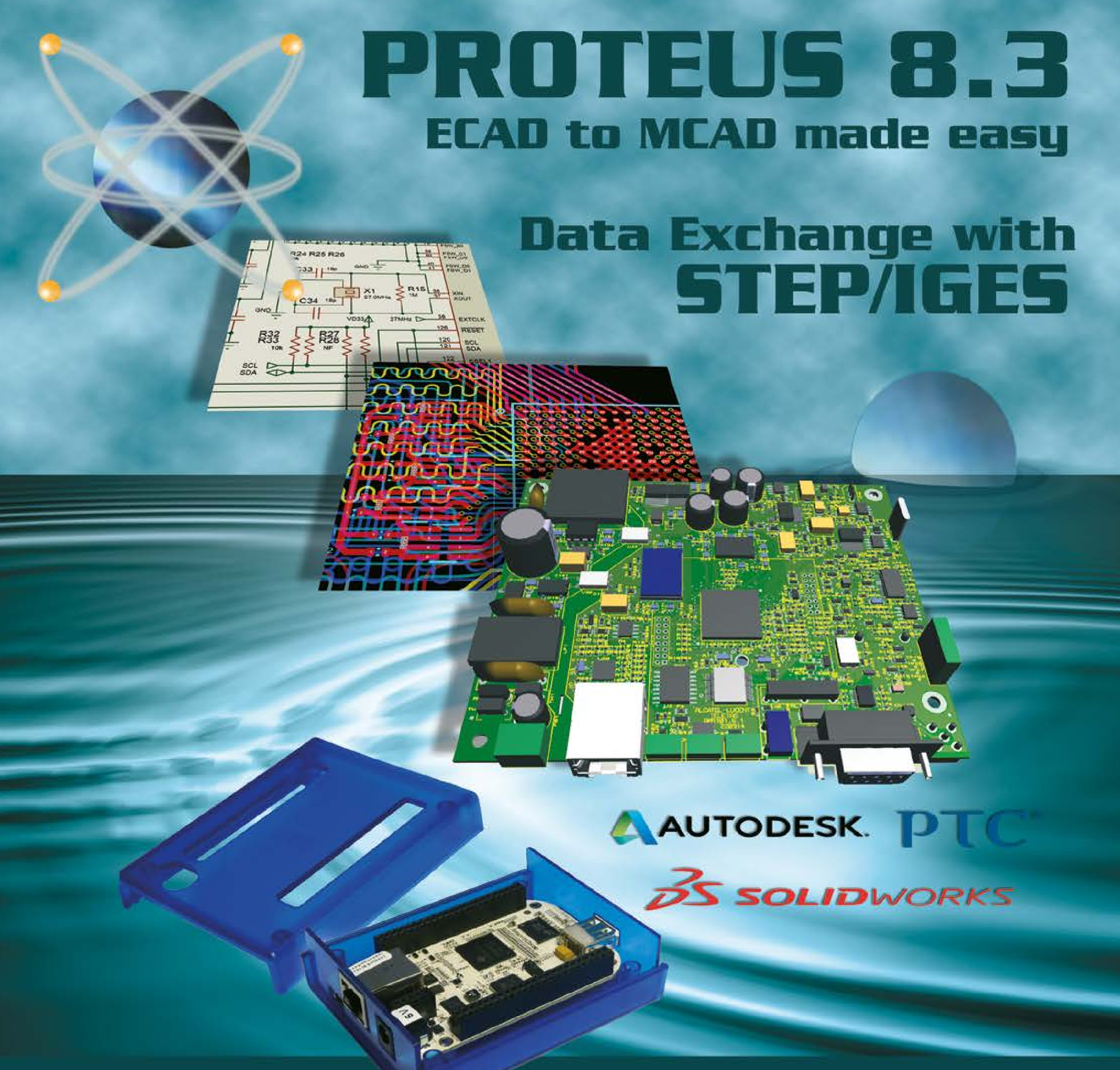
Daily prices ! Price as of : 4.12.2015

Prices in £ incl. statutory VAT, plus shipping costs
reichelt elektronik, Elektronikring 1, 26452 Sande (Germany)

PROTEUS 8.3

ECAD to MCAD made easy

Data Exchange with STEP/IGES



The Proteus Design Suite now includes full support for data exchange with Mechanical CAD packages via the STEP/IGES file formats. This allows you to better visualise your design and helps quickly solve fixtures, fittings and casement problems.

Import 3D STEP/IGES models for your parts and visualise inside the Proteus Design Suite. Export your completed board to Solidworks or other MCAD software.

Visit www.labcenter.com
Tel:+44 01756753440 E-Mail info@labcenter.com

Labcenter Electronics Ltd, 21 Hardy Grange, BD23 5AJ

labcenter Electronics www.labcenter.com